

令和3年度「専修学校における先端技術利活用実証研究」
専修学校遠隔教育導入モデル構築プロジェクト

遠隔教育によるチームプログラミング導入モデルの構築 ＜事業成果報告書＞

本報告書は、文部科学省の教育推進事業委託費による委託事業として、学校法人浦山学園 富山情報ビジネス専門学校が実施した令和3年度「専修学校における先端技術利活用実証研究」の成果をとりまとめたものです。

学校法人浦山学園 富山情報ビジネス専門学校

目次

事業成果報告書	5
専門学校におけるプログラム開発教育に関する状況調査	47
プログラマ育成に関する実態調査（学校）	83
プログラマ育成に関する実態調査（企業）	139
遠隔教育プラットフォーム	225
演習教材	277
遠隔教育プラットフォームを使った遠隔授業の成果検証報告	299

事業成果報告書

令和3年度「専修学校における先端技術利活用実証研究」

遠隔教育によるチームプログラミング導入モデルの構築 事業成果報告

学校法人浦山学園
富山情報ビジネス専門学校



Agenda

1. 文科省委託事業の全体像
2. 事業実施の背景
3. 事業の全体像
4. 令和3年度事業内容
5. 事業評価
6. 令和4年度事業内容





1. 文科省委託事業の全体像

(1) 専修学校における先端技術利活用実証研究

専修学校における先端技術利活用実証研究		令和3年度予算額 (前年度予算額)	623百万円 353百万円
背景	<p>世界に先駆けて人口減少に直面する日本は、産業界等における生産性の向上が喫緊の課題であり、労働力世代の個々人のスキルアップ、技術継承が不可欠であるが、企業等の現場においては、その高い情報伝達能力や再現性から、既に研修等において先端技術（VR・AR等）を導入し、これらの問題に対処しようとする動きがみられ、海外においては先端技術の導入は教育分野においても広がりがつつある。</p> <p>また、新型コロナウイルス感染症の影響が拡大していく中で、専修学校における多様なメディアを高度に利用して行う授業（遠隔授業）は急速に拡大しつつあるが、これまで対面で実施されてきた実践的な職業教育と同等以上の教育的効果を高めるための質向上が課題となっている。</p> <p>⇒ 上記を踏まえると、職業人材の養成場面においても様々な先端技術の活用による教育方法等の改善が重要になる。</p>		
事業内容	■概要	<p>専修学校教育における職業人材の養成機能を強化・充実するため、産学が連携し実践的な職業教育を支える実習授業等における先端技術（VR・AR等）の活用方策について実証・研究を行うとともに、在宅等でも、専修学校における実践的な職業教育の質を落とすことなく提供するため、先端技術を活用した遠隔教育の実践モデルを構築する。また、分野横断型連絡調整会議を設置することにより、上記の各プロジェクトにおける成果に横串を刺し、事業の成果を体系的にまとめる。</p>	
	■スキーム	<p>①先端技術利活用・検証プロジェクト 【約327百万円(前年度 約327百万円)】(16プロジェクト×約20百万円) ○専修学校を中心として、産業界、行政を含めた協議体を各分野で構築 ○電子や実習授業等における先端技術の活用方策(教育手法への落とし込みに係る方策)について実証・研究</p> <p>分野横断連絡調整会議 【約47百万円(前年度 約23百万円)】(2事務所×約23百万円) ○各プロジェクトの進捗管理及び連絡調整 ○各プロジェクトの事業成果を体系的にまとめ、普及・定着方策を練り ○新たな技術開発動向や活用事例のリサーチ等</p> <p>新たな技術開発に関する示唆 産業界 動向リサーチ</p>	
成果	<p>多様な分野において先端技術を活用した効果的な教育手法、コンテンツ、カリキュラムが作成され、それらが専修学校における教育プログラムに導入されるとともに、在宅等でも、対面授業と同等以上の教育効果が得られることによって、職業人材の養成機能を強化・充実していくとともに、先端技術の技術革新や社会実装が触発される。</p>		





(2) 専修学校遠隔教育導入モデル構築プロジェクト

専修学校遠隔教育導入モデル構築プロジェクト
 (「専修学校における先端技術活用実証研究」の新規メニューとして実施)

令和3年度予算額 (案) 270百万円 (新規)

背景・課題
 新型コロナウイルス感染症の感染拡大の影響により、遠隔授業を実施・検討中の専門学校は、4月6日時点では23.6%に止まっていたところ、緊急事態宣言下の5月11日時点で84.6%に拡大。しかし、宣言解除後の6月1日時点には、実施している学校は49.9%に減少(文部科学省調べ)。遠隔授業の実施割合が減少した理由として、制度的に遠隔授業の実施が認められているものの、実践的な職業教育を行うという特色から、対面授業を前提に実習・実技にウエイトを置いたカリキュラムが構築されていること、これまで遠隔授業のノウハウが乏しいことなどが考えられる。
 また、遠隔授業を実施している学校では、遠隔授業の指導方法が未確立であるため指導内容は教員のITスキルに依存している状況にある。また、昨年度に比べて生徒の学習へのモチベーションの低下(「株」進研アド調べ)が指摘されるなど、遠隔授業の実施によって、これまでと同等の教育効果が得られているのかについては不安が聞かれる。このため、今後感染の長期化や新たな脅威が懸念されることから、専修学校において社会に必要不可欠な専門人材の供給を継続できる体制の構築が急務である。

事業内容
 ①在宅等でも、専修学校教育の質を落とすことなく提供するため、先端技術を活用した遠隔教育の実践モデルを開発し、その効果を検証する。
 ●件数・単価：12箇所×約210百万円
 ②分野横断型連絡調整会議を設置し、各取組の進捗管理及び連絡調整を行い、事業成果を体系的にとりまとめるとともに普及・定着方を検討、展開する。
 ●件数・単価：1箇所×約230百万円

遠隔教育導入モデルの構築！！
 新型コロナウイルス感染症の感染拡大の影響による、自宅学習などの環境下においても、専修学校における実践的な職業教育を提供するためのモデルを構築！

分業横断連絡調整会議
 ●各プロジェクトの進捗管理及び連絡調整
 ●各プロジェクトの事業成果を体系的にとりまとめ、普及・定着方を検討
 ●新たな技術開発動向や活用事例のリサーチ 等

アウトプット(活動目標)
 ●専修学校における遠隔教育プログラムの構築数
 ⇒ 12モデル
 ●プログラム受講者数
 ⇒ 600名 (12モデル×50名)

アウトカム(成果目標)
 初・期：新型コロナウイルス感染症が長期化した場合でも、社会に必要不可欠な専門人材を供給する教育体制の構築
 中長期：ハード・ソフト両面でのICT環境を整備することで、個別最適化された質の高い教育を提供

インパクト(国民・社会への影響)
 専修学校が、社会や企業ニーズに則した実践的な職業人材を輩出することにより、我が国の労働生産性の向上及び生涯を通じた学習機会の拡大に寄与



(2) 専修学校遠隔教育導入モデル構築プロジェクト

背景・課題

緊急事態宣言解除後、専門学校の遠隔授業の実施割合は84.6%から49.9%に低下

(理由)
 実習・実技にウエイトを置いたカリキュラム
 遠隔授業のノウハウが乏しい

(課題)
 教員のITスキルに依存
 学生のモチベーションの低下

専門人材の供給を継続できる体制の構築が急務

(2) 専修学校遠隔教育導入モデル構築プロジェクト



事業内容

在宅等でも、専修学校教育の質を落とすことなく提供するため、先端技術を活用した遠隔教育の実践モデルを開発し、その効果を検証する。
(12箇所)

分野横断型連絡調整会議を設置し、各取組の進捗管理及び連絡調整を行ない、事業成果を体系的にとりまとめるとともに普及・定着方策を検討、展開する。(1箇所)

令和3年度からの3か年事業



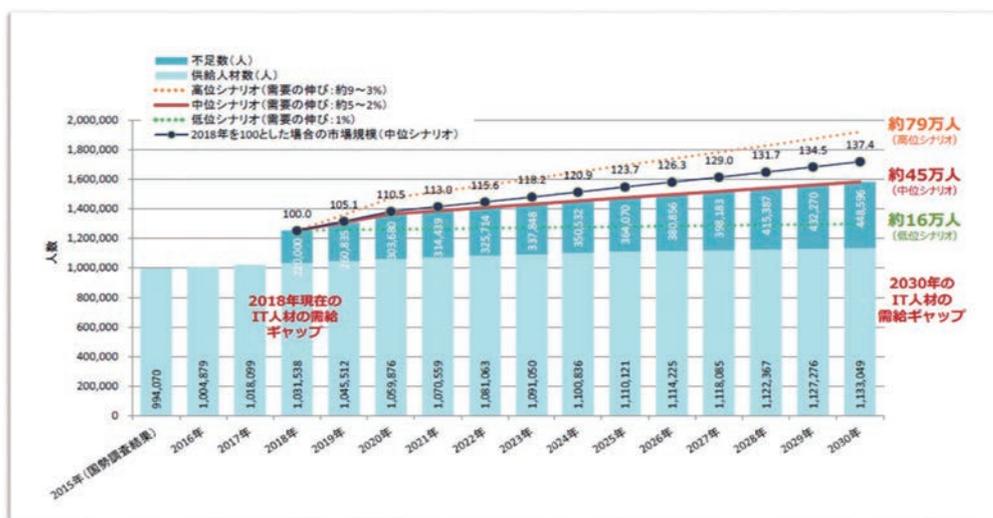
2. 事業実施の背景



(1)なぜ、プログラミング教育なのか？

要因	課題
外的要因	<ul style="list-style-type: none"> ・プログラマ人材不足 ・プログラミング教育の必修化
内的要因	<ul style="list-style-type: none"> ・プログラミング教育のレベル不足 ・演習授業の負担大
複合要因	<ul style="list-style-type: none"> ・実践的インターンシップの不足 ・学生と企業との接点不足

(2)プログラマ人材不足



2030年に最大で79万人の不足

経済産業省 IT人材需給に関する調査 より





(3) 小～高等学校までのプログラミング教育の必修化

小・中・高等学校別のポイント（総則及び各教科等）

- 小学校においては、文字入力など基本的な操作を習得、**新たにプログラミング的思考を育成**
各教科等の特質に応じて、児童がコンピュータで文字を入力するなどの学習の基盤として必要となる情報手段の基本的な操作を習得するための学習活動や、プログラミングを体験しながらコンピュータに意図した処理を行わせるために必要な論理的思考力を身に付けるための学習活動を計画的に実施することを明記。【総則】
- 中学校においては、技術・家庭科（技術分野）において**プログラミング、情報セキュリティに関する内容を充実**
「計測・制御のプログラミング」に加え、「ネットワークを利用した双方向性のあるコンテンツのプログラミング」等について学ぶ。【技術・家庭科（技術分野）】
- 高等学校においては、**情報科において共通必修科目「情報Ⅰ」を新設し、全ての生徒がプログラミングのほか、ネットワーク（情報セキュリティを含む）やデータベースの基礎等について学習**
「情報Ⅰ」に加え、選択科目「情報Ⅱ」を開設。「情報Ⅰ」において培った基礎の上に、情報システムや多様なデータを適切かつ効果的に活用し、あるいはコンテンツを創造する力を育成。【情報科】

文部科学省 新学習指導要領のポイント より

早ければ2025年にプログラミング教育を受けた
学生が専門学校へ入学してくる



(4) 専門学校でのプログラミング教育の現状

課題

- ・基礎から教えていて、2年間で実践レベルに到らない
- ・学生にレベル差があって教員の負担が大きい
- ・出来あがったプログラムの品質が低い

要因

- ・失敗できない教育環境
- ・個別対応できない教育環境
- ・プログラム品質保証への意識不足



プログラミングへの興味を失う
全体的に低いプログラミングレベル

(5) 産学連携課題



項目	課題
インターンシップ	・専用カリキュラム作成・実施の負担 ・セキュリティ問題
連携事業	・企業が求めるレベルへの未達
就職	・学生の実践力不足 ・実力を測る環境が無い(総合力で判断)

(6) 事業目標



(a) 対面と同等な教育効果を得るだけではなく、遠隔授業の特性を活かして

- ・学生のレベルに応じて楽しく学習できる環境を提供する
- ・企業レベルの高品質なプログラミングを実現する
- ・IT企業も参加できる環境を提供する
- ・教員が簡単に導入でき、学習状況を容易に管理できる環境を提供する
- ・手間のかかる演習指導の負担を下げる

(b) 教育環境を公開することで

- ・全国のIT教員が利用、交流し、プログラミング教育の向上を図る
- ・IT企業と学生が交流し、インターンシップ、コーオプ教育の実現、就職マッチングを実現する
- ・IT企業と教員が交流し、最新のIT知識、情報を教育に反映する



2. 事業の全体像

(1) 事業目標

(a) 対面と同等な教育効果を得るだけではなく、遠隔授業の特性を活かして

- ・学生のレベルに応じて楽しく学習できる環境を提供する
- ・企業レベルの高品質なプログラミングを実現する
- ・IT企業も参加できる環境を提供する
- ・教員が簡単に導入でき、学習状況を容易に管理できる環境を提供する
- ・手間のかかる演習指導の負担を下げる

(b) 教育環境を公開することで

- ・全国のIT教員が利用、交流し、プログラミング教育の向上を図る
- ・IT企業と学生が交流し、インターンシップ、コーオプ教育の実現、就職マッチングを実現する
- ・IT企業と教員が交流し、最新のIT知識、情報を教育に反映する



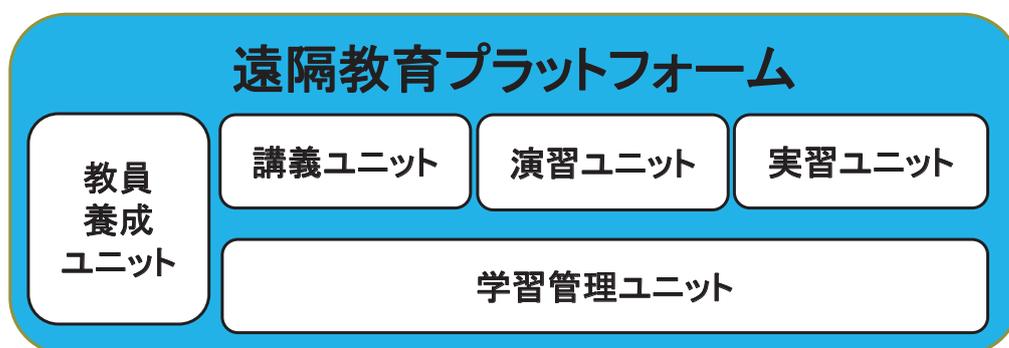


(2) 事業の概要

- クラウド上に無料または安価な既存サービス、アプリケーションを組み合わせて、遠隔教育プラットフォームを構築する。
- 学生にプラットフォーム上の演習課題を取り組ませ、対面授業と同等以上の教育効果を実現しているか、企業レベルのプログラミング能力に達しているかを検証、評価する。
- 遠隔教育プラットフォームを全国に普及させる。



(3) 遠隔教育プラットフォーム





3. 令和3年度事業内容

(1) 実施事業

調査事業

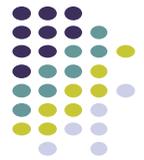
- 専門学校におけるプログラム開発教育に関する状況調査
- プログラマー育成に関する実態調査

開発・実証事業

- 演習ユニット(プロトタイプ)
- 学習管理ユニット(プロトタイプ)



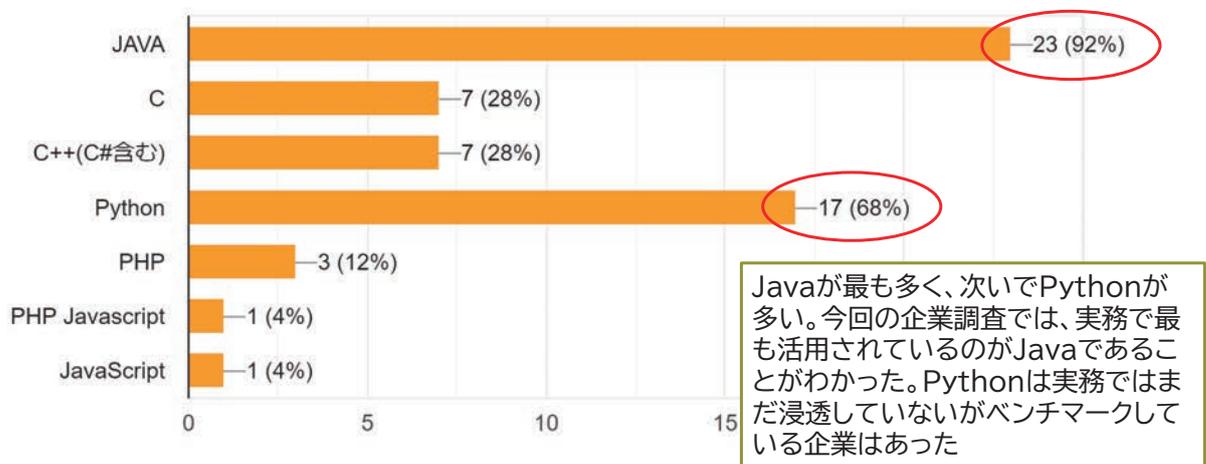
(2) 専門学校でのプログラミング教育に関する状況調査



項目	内容
目的	遠隔教育プラットフォームの学習方法、評価方法の傾向を知る
対象	全国の職業実践専門課程の認定を受けたIT系学科を設置する専門学校
内容	プログラミング教育における学習方法や評価方法等を調査する
方法	Webを利用したアンケート調査
回収率	22.4% (107校中24校回収)

(7) 全国の専門学校へのアンケート調査結果

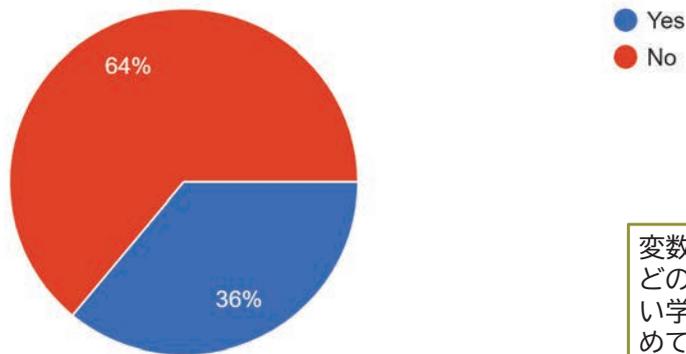
① 開発実践力習得のために、学習に力を入れているプログラミング言語は何ですか？(複数選択可)



(7) 全国の専門学校へのアンケート調査結果



② 変数の命名規則やコメントの付け方などコーディングルールを決めて演習していますか？

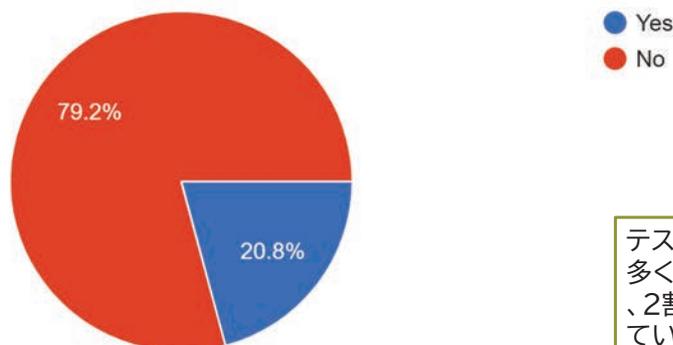


変数の命名規則やコメントの付け方などのコーディングルールは決めていない学校の方が多く、25校中16校が決めていない

(7) 全国の専門学校へのアンケート調査結果



③ テスト仕様書を作成してテストしていますか？

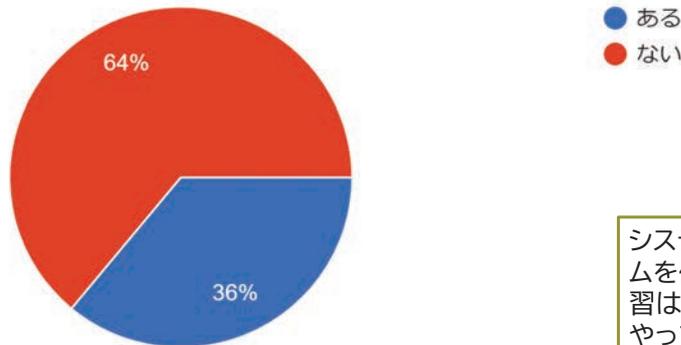


テスト仕様書を作成していない学校が多く約8割(19校)を占めている。一方、2割(5校)ではテスト仕様書を作成している

(7) 全国の専門学校へのアンケート調査結果



④ システムの一部となる部分のプログラムを作らせ、全体に組み込むような実習形態はありますか？

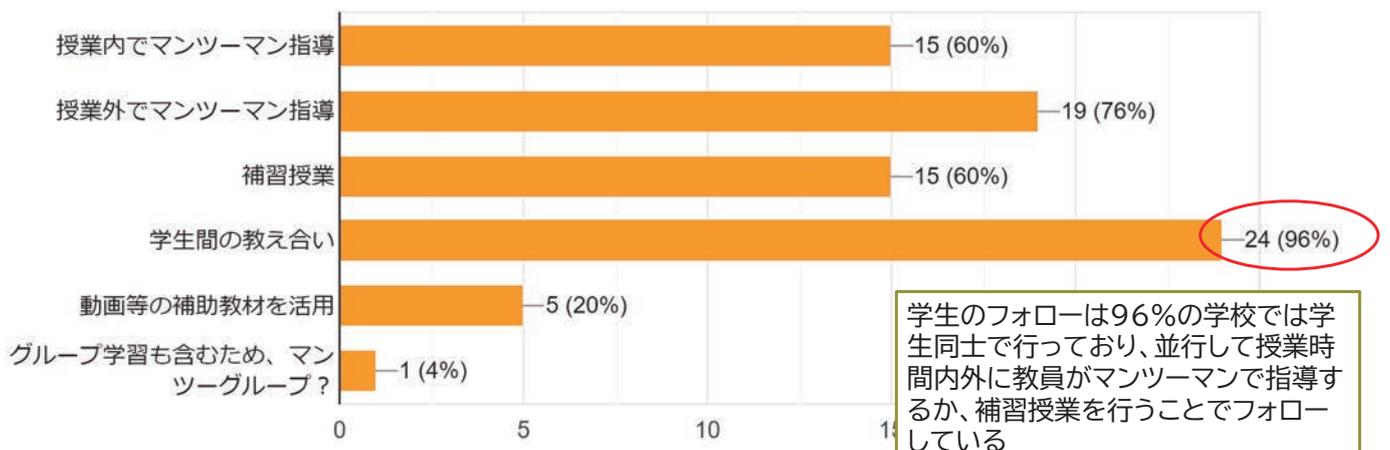


システムの一部となる部分のプログラムを作らせ、全体に組み込むような実習はやっていない学校のほうが多く、やっている学校は9校である

(7) 全国の専門学校へのアンケート調査結果



⑤ プログラミング課題が全くできない、手が動かないという学生に対するフォローはどのように行っていますか？(複数選択可)

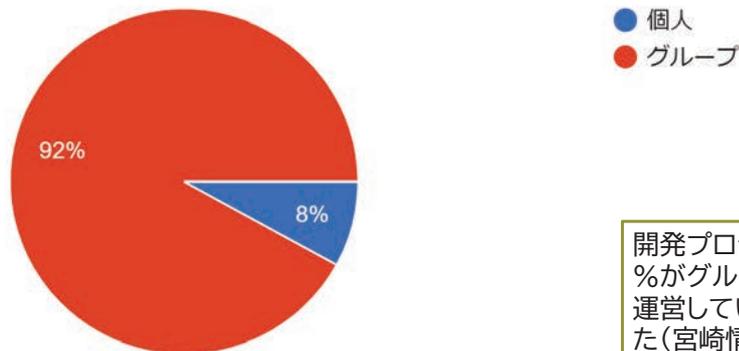


学生のフォローは96%の学校では学生同士で行っており、並行して授業時間内外に教員がマンツーマンで指導するか、補習授業を行うことでフォローしている

(7)全国の専門学校へのアンケート調査結果



⑥ 開発プロジェクト(卒業制作等)は学生が個々で運営していますか、グループで運営していますか？



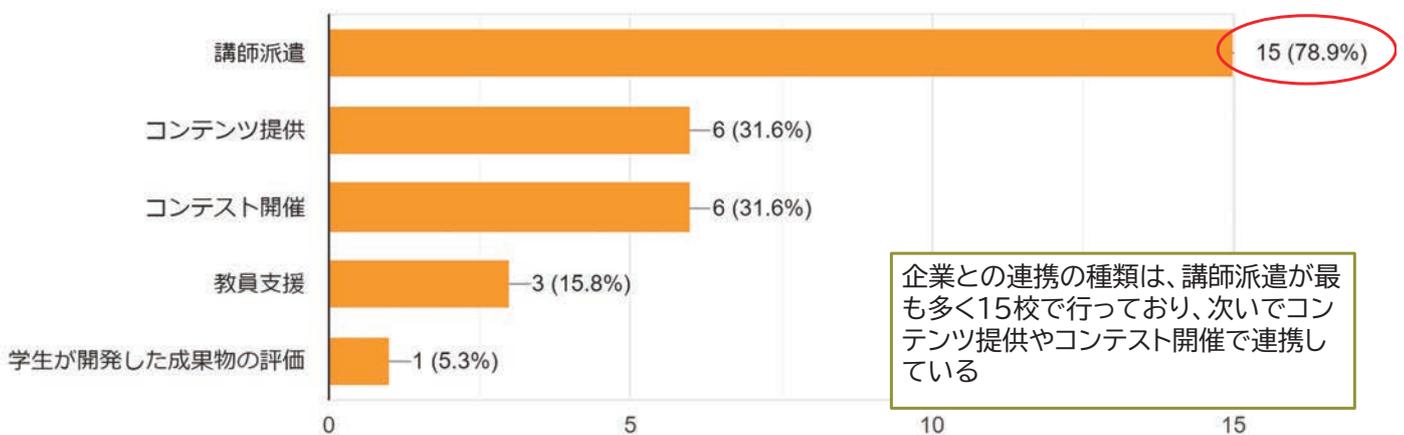
開発プロジェクト(卒業制作等)は92%がグループで運営しており、個人で運営している学校は2校(8%)であった(宮崎情報ビジネス医療専門学校・甲府市立甲府商科専門学校)

(7)全国の専門学校へのアンケート調査結果



⑦ プログラミングを含むシステム開発等で企業と連携していますか？

◆◆◆連携の種類◆◆◆



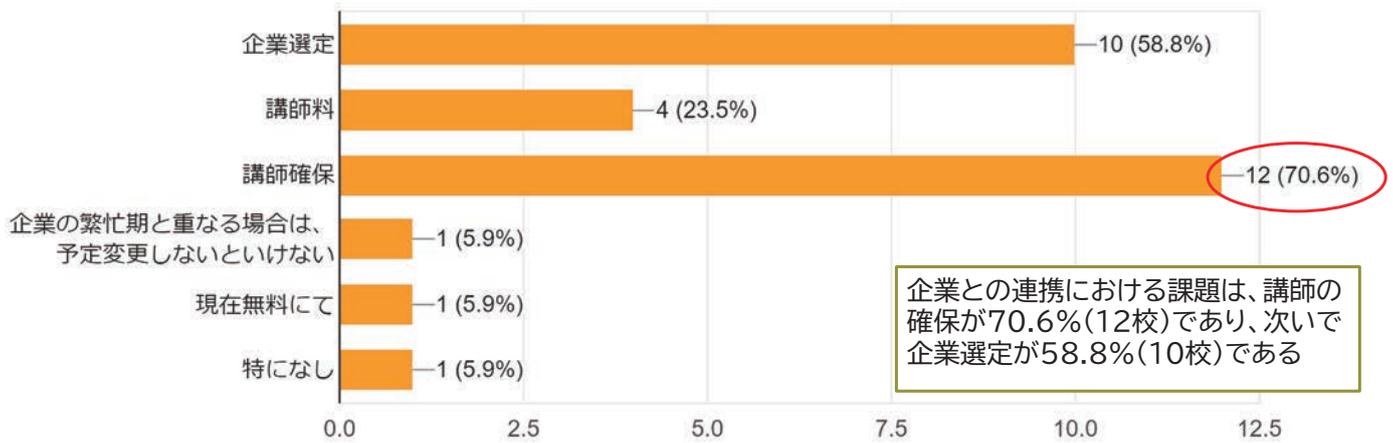
企業との連携の種類は、講師派遣が最も多く15校で行っており、次いでコンテンツ提供やコンテスト開催で連携している

(7)全国の専門学校へのアンケート調査結果



⑧ プログラミングを含むシステム開発等で企業と連携していますか？

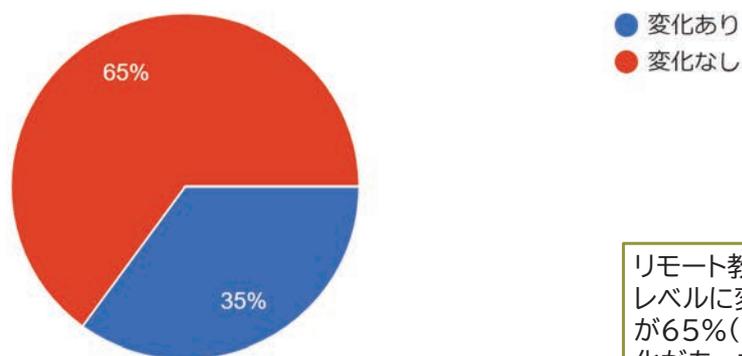
◆◆◆企業連携の課題◆◆◆



(7)全国の専門学校へのアンケート調査結果



⑨ リモート教育に伴い、開発実践力取得レベルに変化はありましたか？



リモート教育に伴い、開発実践力取得レベルに変化はなかったと考える学校が65%(13校)と大半を占めるが、変化があったとする学校も35%(7校)あった

(7)全国の専門学校へのアンケート調査結果



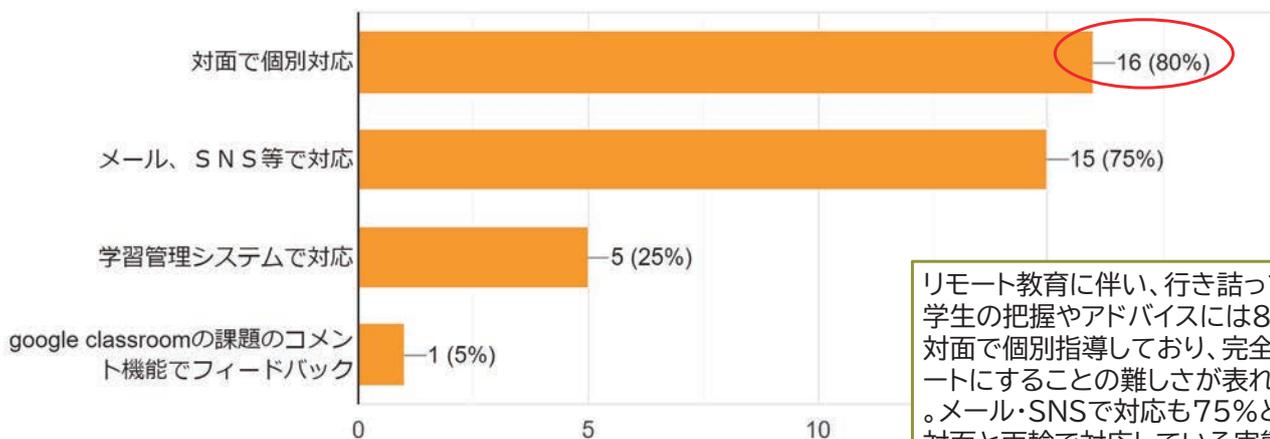
⑩リモート教育に伴い、開発実践力取得レベルに変化があった理由として考えられることを教えてください。

- リモートで、聞き流している学生のスキルは身につけていない
- 対面での打ち合わせが出来なくて、レベルが低下した
- 学生のモチベーションの維持、自己実現が難しい
- 対面での直接指導が減った。学生のモチベーションコントロールの低下。
- 対面なら困った時にすぐ、近くの学生や教員に相談できるがそれが、しづらい
- 一人一人の習熟度等をリアルタイムで把握できない。

(7)全国の専門学校へのアンケート調査結果



⑪ リモート教育に伴い、行き詰っている学生の把握やアドバイスはどのようにしていますか？(複数選択可)

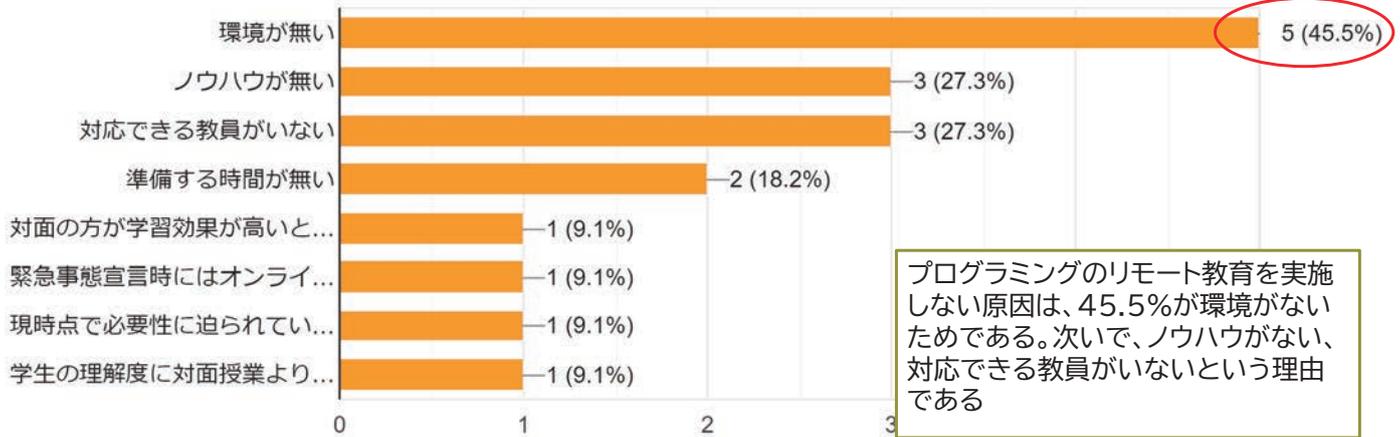


リモート教育に伴い、行き詰っている学生の把握やアドバイスには80%が対面で個別指導しており、完全にリモートにすることを難しさが表れている。メール・SNSで対応も75%と多く、対面と両輪で対応している実態が推測できる

(7) 全国の専門学校へのアンケート調査結果



⑫ プログラミングのリモート教育を実施しない原因または障害はどんなことですか？

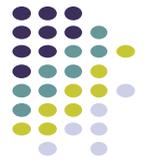


(8) プログラマ育成に関する実態調査



項目	内容
目的	プログラマが備えておくべき素養と要件を明確にする
対象	全国の6地域(富山、群馬、東京、香川、高知、福岡)の専門学校及び遠隔にてIT人材育成を実施している研修業者・IT企業
内容	専門学校と企業のプログラマ育成方法やプログラム品質の評価方法の実態およびギャップを調査する
方法	現地視察調査
調査数	専門学校:6校 研修業者:2社 IT企業:11社

(8) プログラマ育成に関する実態調査(学校調査編)



- リモート教育では、対面授業以上に学生の速度差・レベル差が顕著に出る傾向がある
- 教員側としては準備に余計時間がかかるようになった
- コーディング規約がない学校が多い
- ほとんどの学校でテストまではしっかりと取り組めていない
- フレームワークの授業はやっていない学校が多い

(9) プログラマ育成に関する実態調査(企業調査編)



- Javaはすべての調査対象企業で主要言語として使っている
- 新入社員教育ではJavaが多く採用されている
- リモート環境下で使われるコミュニケーションツールの利用経験を評価する企業があった
- プロジェクトごとにコーディングの基準を決めている企業がほとんどであった
- テストは全ての企業が重要視しており、テスト教育を強化して欲しいとする企業があった

(9) プログラマ育成に関する実態調査(企業調査編)



- プログラマのスキルレベルの指標を持っている企業は、ほとんど無かった
- 入社時に求められるプログラミングレベルを規定している企業は無かった
- 学校でのプログラミング経験は一定の評価はあるものの、採用に関してはコミュニケーション能力やその他の部分が評価され採用に結びつくことが多い

(10) 令和3年度調査結果の反映



調査結果	対応状況
Java言語	対応済み
コーディングルール(規約)	次年度以降対応予定
テスト設計	次年度以降対応予定
システム改修に対応した教材	対応済み
学生間での教え合い	対応済み
グループ開発	対応済み
企業連携	次年度以降対応予定
学生のモチベーション維持	次年度以降対応予定
学生-教員間のコミュニケーション	対応済み

(10) 令和3年度調査結果の反映



調査結果	対応状況
課題のリアルタイム管理	対応済み
教員の教材開発負担の低減	対応済み
フレームワーク利用	対応済み
企業のプログラミングスキルの低評価	次年度以降対応予定

(11) 遠隔教育プラットフォームの開発



演習ユニット

システム	内容
プログラミング演習	課題管理とオブジェクト指向設計に基づいたフレームワークを活用した高速開発の開発環境
テスト支援システム	テスト設計、テスト項目及びテスト結果の管理 テスト結果、プログラム不良からプログラム品質を評価
コミュニケーション	学生と教員間、学生間のチャット、テレビ会議等のコミュニケーション手段の提供、コミュニケーション結果の保存 学生からの想定質問に対するチャットボット
プロジェクト管理	チーム編成支援と工程予実績を管理、評価

(11)遠隔教育プラットフォームの開発



学習管理ユニット

内容
参加学生のエントリー、ID発行、演習を通じた学習進捗・学習評価・学習履歴等のプラットフォームを利用するための基本情報を管理

(11)遠隔教育プラットフォームの開発



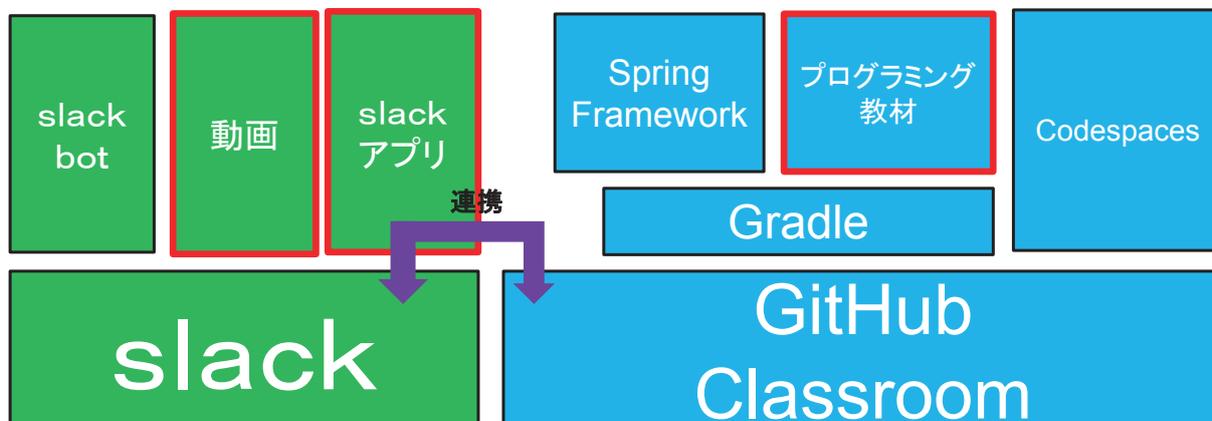
学習管理システム比較

学習管理システム	メリット	デメリット
UMU	<ul style="list-style-type: none">AI機能を持ち、機能豊富	<ul style="list-style-type: none">プログラミングに特化していない使用料有料
Google Classroom	<ul style="list-style-type: none">多くの教育機関で使用されている使用料無料	<ul style="list-style-type: none">プログラミングに特化していない
GitHub Classroom	<ul style="list-style-type: none">プログラミングに特化している使用料無料GitHub自体が多くの開発現場で使用されている	<ul style="list-style-type: none">コミュニケーション機能が少ない



(11)遠隔教育プラットフォームの開発

プラットフォームの構成



赤枠: 追加開発



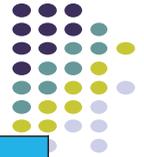
(11)遠隔教育プラットフォームの開発

各アプリケーションの役割

アプリケーション	概要	役割
slack	ビジネスチャットツール	<ul style="list-style-type: none"> 教員－学生間のコミュニケーション
slackアプリ	slack上で動作する追加機能	<ul style="list-style-type: none"> slack初期設定の自動化 課題の進捗管理 評価の登録
slack bot	メッセージの自動返答	<ul style="list-style-type: none"> 定型質問の自動回答
GitHub Classroom	教育機能を持ったソフトウェア開発プラットフォーム	<ul style="list-style-type: none"> 課題管理 提出課題の自動採点
Gradle	自動環境作成ツール	<ul style="list-style-type: none"> 学生への開発環境の提供
Spring Framework	Java言語開発用フレームワーク	<ul style="list-style-type: none"> フレームワークを使用したプログラミング環境の提供
Codespaces	クラウド上のプログラム開発環境	<ul style="list-style-type: none"> 学生のプログラミング環境

(11)遠隔教育プラットフォームの開発

各アプリケーションの標準機能と連携機能



アプリケーション	機能
slack	教員-学生間のコミュニケーション 学生間のコミュニケーション チャットbot
Github Classroom	授業用のクラス作成、教員・学生のエントリ、課題登録 課題用のURL発行 課題のフィードバック 課題へのエントリ 課題の対応と提出、修正、再提出
連携	授業のチャンネル作成 課題のチャンネル作成と配布 提出された学生の課題情報の自動連携 課題の進捗と採点の確認、評価の登録 ポータル画面の表示(割り当てた授業、課題の表示)

(11)遠隔教育プラットフォームの開発

ポータル画面



教員用ポータル画面



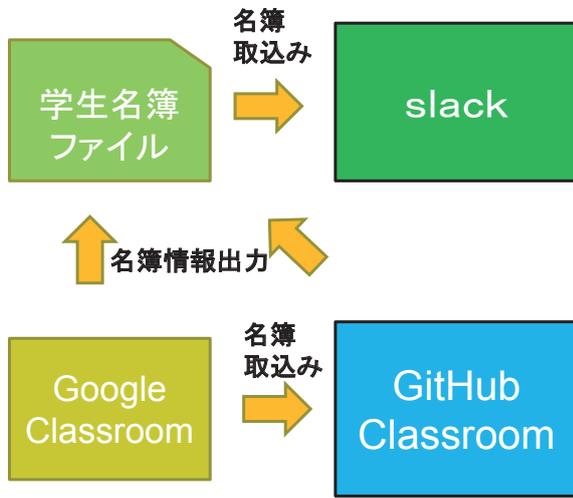
学生用ポータル画面



(11)遠隔教育プラットフォームの開発



環境導入

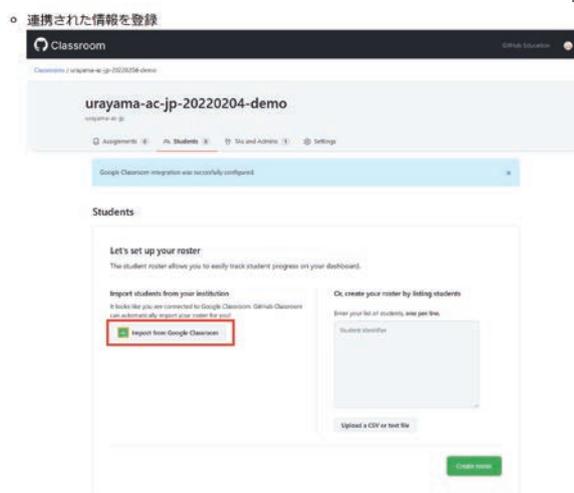
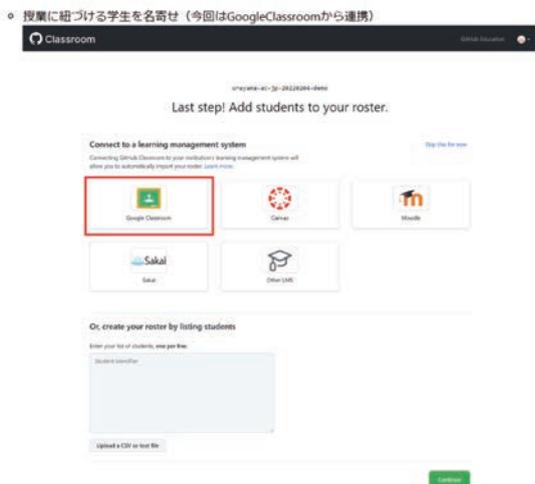


自動生成
実習用クラス
授業チャンネル
課題チャンネル
プログラム開発環境

(11)遠隔教育プラットフォームの開発

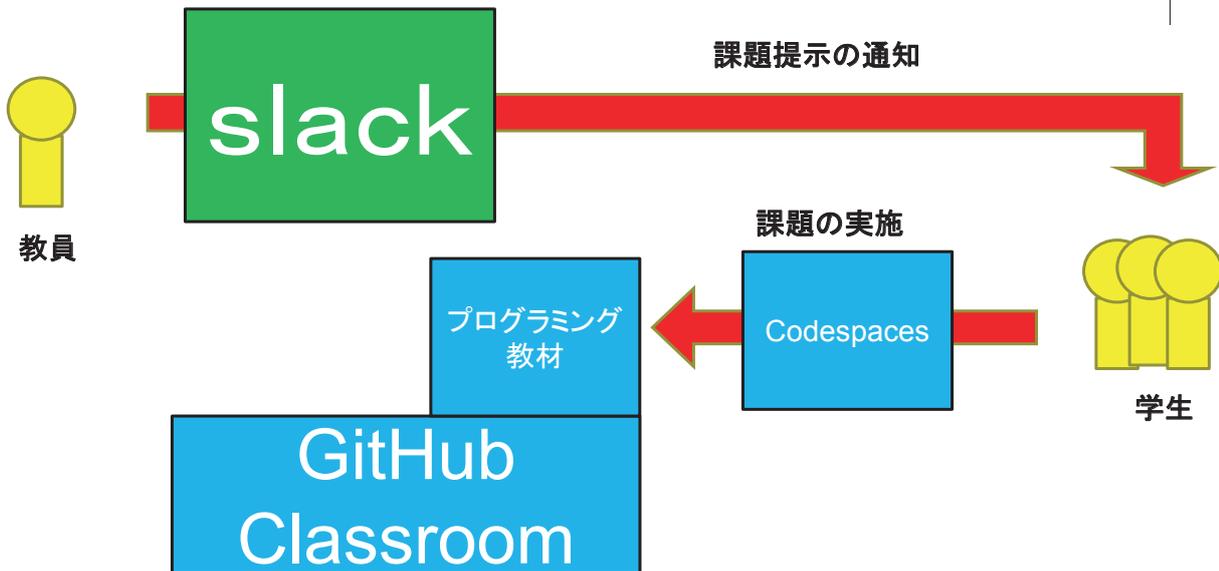


環境設定画面



(11)遠隔教育プラットフォームの開発

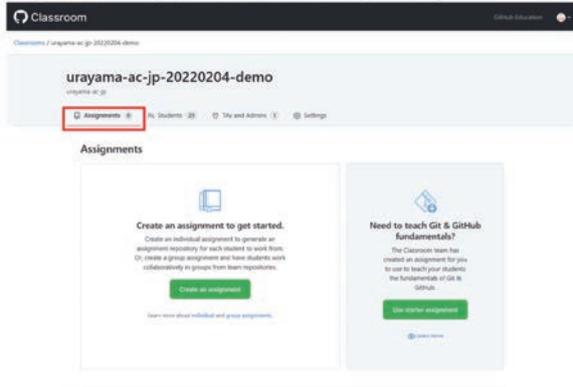
演習課題の提示と実施



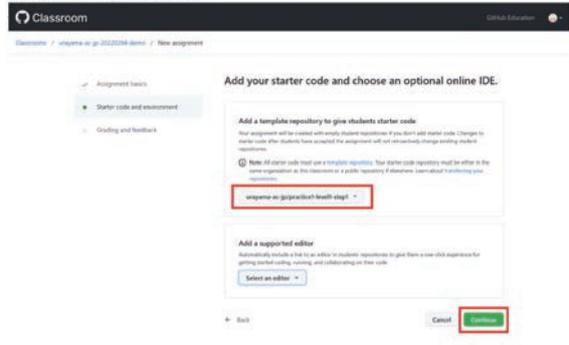
(11)遠隔教育プラットフォームの開発

課題設定画面

- 課題の登録を行っていく(課題はクラスを作った後で作成することも可能)



- 学生に配布するリポジトリを選択(ここで選択されたリポジトリがコピーされて、学生が課題を行うリポジトリが作成される)

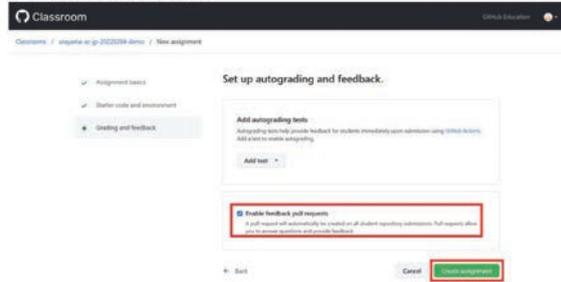


(11)遠隔教育プラットフォームの開発

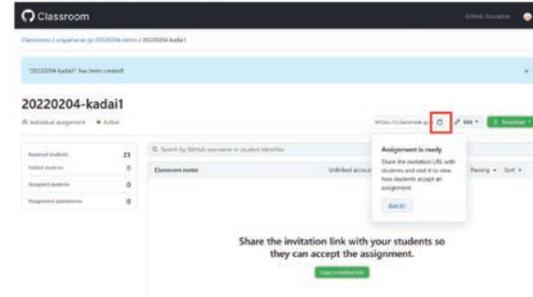
課題設定画面



フィードバックを行う設定を追加



課題を配布するためのURLが発行されるのでコピー ※別途プラットフォームを使って配布



(11)遠隔教育プラットフォームの開発

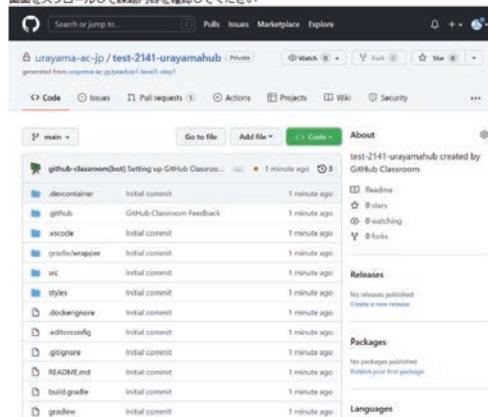
課題実施画面



1. Slackの課題チャンネルから赤枠の課題URLをクリック



6. 以下のGitHubの画面が表示されることを確認 画面をスクロールして課題内容を確認してください

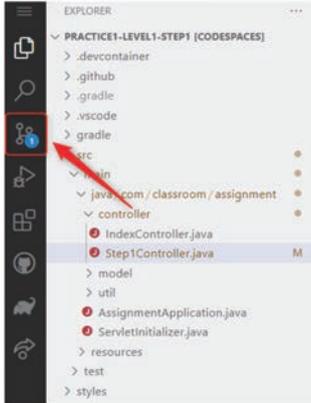


(11)遠隔教育プラットフォームの開発

課題提出画面



1. 木の枝のようなマークを押します。



```
Step1Controller.java M X
src > main > java > com > classroom > assignmen
1 package com.classroom.assignment
2
3 import org.springframework.stere
4 import org.springframework.ui.Mo
5 import org.springframework.web.b
6 import org.springframework.web.b
7
8 @Controller
9 @RequestMapping("/step1")
10 public class Step1Controller {
11
12     @GetMapping
13     public String index(Model mode
14         model.addAttribute("title",
15             return "step1/index";
16     }
17 }
18
```

6. チェックボタンを押してコミットしましょう。



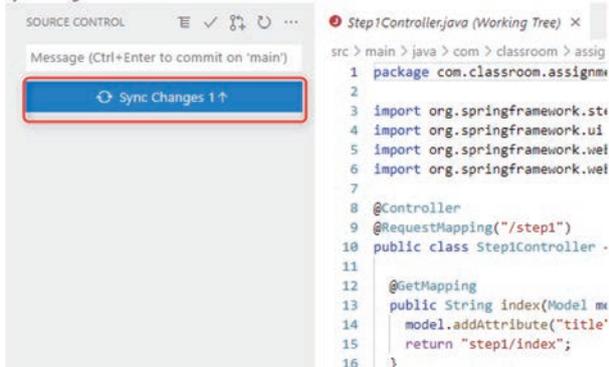
```
Step1Controller.java (Working Tree) M X
src > main > java > com > classroom > assignm
1 package com.classroom.assignm
2
3 import org.springframework.ste
4 import org.springframework.ui
5 import org.springframework.wel
6 import org.springframework.wel
7
8 @Controller
9 @RequestMapping("/step1")
10 public class Step1Controller .
11
12 @GetMapping
13 public String index(Model mc
14     model.addAttribute("title"
15     return "step1/index";
16 }
```

(11)遠隔教育プラットフォームの開発

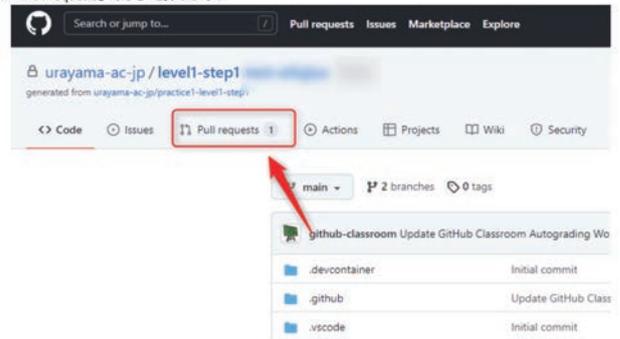
課題提出画面



7. Sync Changesボタンを押してコミットをプッシュしましょう。

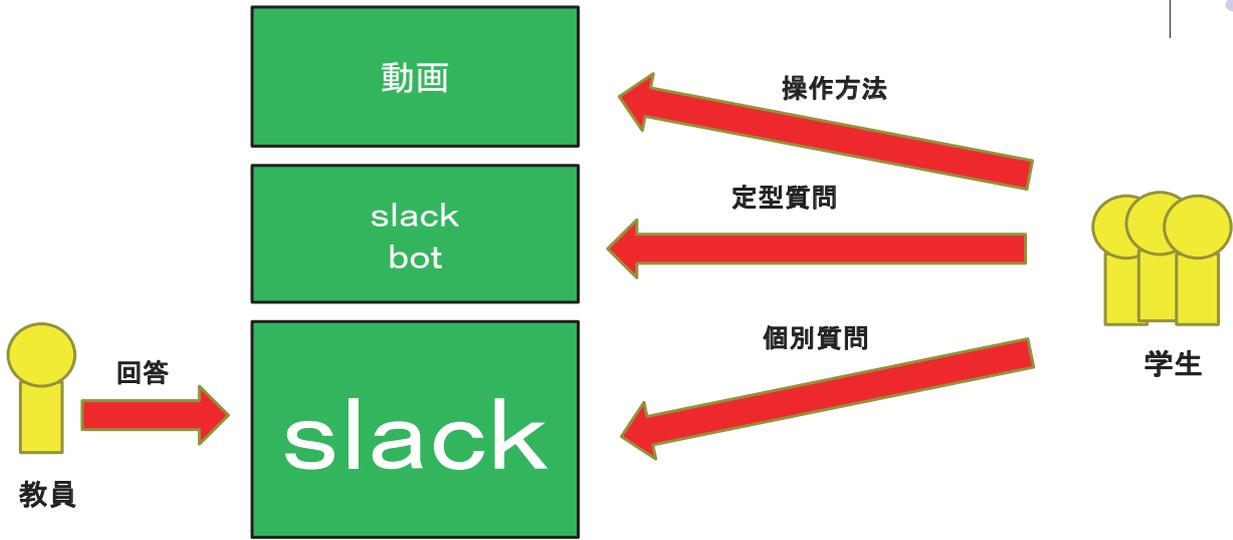


14. 「Pull requests」ボタンを押します。



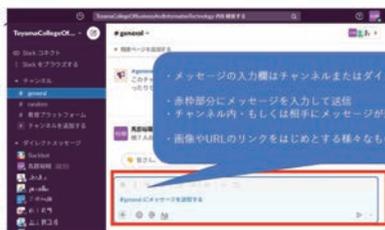
(11)遠隔教育プラットフォームの開発

学生－教員間コミュニケーション



(11)遠隔教育プラットフォームの開発

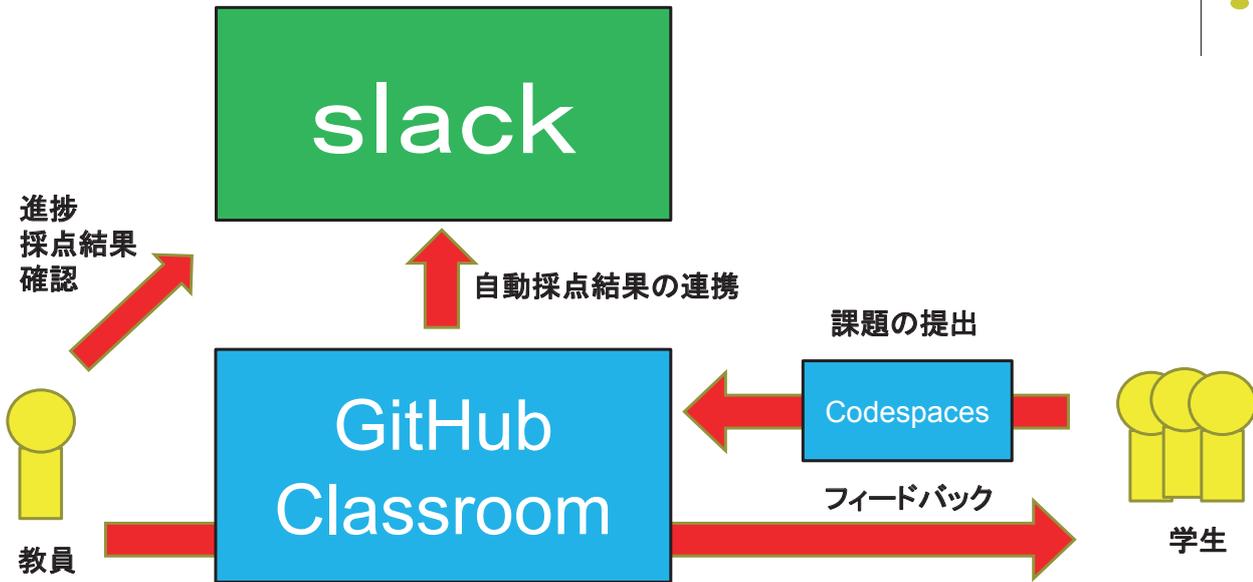
学生－教員間コミュニケーション



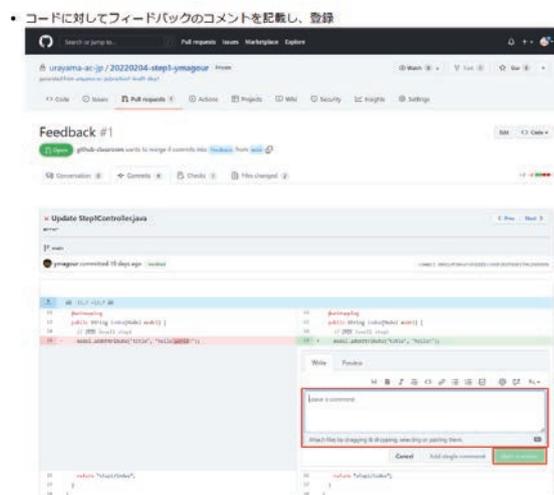
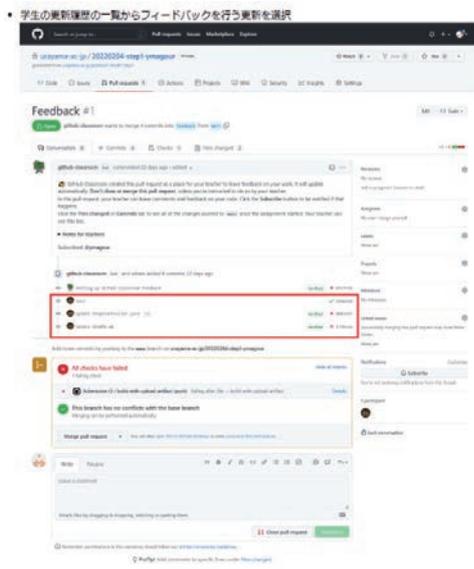
チャットボット



(11)遠隔教育プラットフォームの開発 課題評価とフィードバック



(11)遠隔教育プラットフォームの開発 課題評価とフィードバック





(11) 遠隔教育プラットフォームの開発

プログラミング演習システム

項目	内容
機能	課題管理とオブジェクト指向設計に基づいたフレームワークを活用した高速開発の開発環境
言語	Java言語
課題	2段階レベルの課題
課題数	各レベル 4本

(12) 教材の開発



レベル	レベル概要	演習課題
1	実際に動作するベースプログラムに対して、一部の命令を書き換えることにより、要件定義を満たすプログラム	1. フレームワークを利用したプログラムの実行と表示文言の変更 2. GETパラメータの取得 3. 画面デザイン(Bootstrapの使用) 4. バリデーション
2	実際に動作するベースプログラムに対して、一部分の処理を参考に追加することにより、要件定義を満たすプログラム	1. 入力フォームの追加 2. 表示項目(エンティティ)の追加 3. サービスクラスタの作成 4. データベースの利用
3	実際に動作するベースプログラムに対して、新しい処理を追加することにより、要件定義を満たすプログラム	今年度開発なし
4	すべてのプログラムを作り上げることにより、要件定義を満たすプログラム	今年度開発なし

(12)教材の開発



- 演習課題

画面に新しい入力フォームを追加する(レベル1)



(12)教材の開発



- 解答

【修正前】

```

src/main/java/com/classroom/assignment/model/request/Comment.java
4
5 private String name;
6 private String content;
7
8 public Comment(String name, String content) {
9     this.setName(name);
10    this.setContent(content);
11 }
12
13 public String getName() {
14
15 }
16
17 @-25,4 +27,12 @@ public String getContent() {
18     public void setContent(String content) {
19         this.content = content;
20     }
21 }
    
```

【修正後】

```

4
5 private String name;
6 private String content;
7 private String mail;
8
9 public Comment(String name, String content,String address) {
10    this.setName(name);
11    this.setContent(content);
12    this.setMail(mail);
13 }
14
15 public String getName() {
16
17 }
18
19 public void setContent(String content) {
20    this.content = content;
21 }
22
23 public String getMail() {
24    return mail;
25 }
26
27 public void setMail(String mail) {
28    this.mail = mail;
29 }
30 }
    
```



(12)教材の開発

- テスト

- 追加したフォーム(メールアドレス)に値を設定しリクエスト
- 表示された画面にフォームに設定した内容が含まれていること

```
@Test
public void mailItemFormExistenceConfirmation() throws Exception {
    // given
    // when
    // then
    this.mockMvc.perform(
        MockMvcRequestBuilders.post("/step3/comment").param("mail", "test_mail").andDo(print())
        .andExpect(status().isOk())
        .andExpect(content().string(containsString("test_mail"))));
}
```

(13)遠隔教育プラットフォームの実証

項目	内容
目的	プログラミング学習システムとして求められる学習効果、利便性があるか確認する
実施場所	富山情報ビジネス専門学校
参加者	情報システム学科 1年生 約40名
方法	対面学習者20名、プラットフォームによる学習者20名に区分し、学習過程や成果物によって取得するアセスメント、コメント、開発速度、プログラム品質、プログラム稼働状況等の項目について比較検証する。



(13)遠隔教育プラットフォームの実証



実証講座の内容

授業形式	対象者	期間	実施内容	実施方法
遠隔	富山情報ビジネス専門学校 1年情報システム学科 23名	2022年2月7日 ～2月10日	レベル1教材2本 レベル2教材2本	各アプリケーションの説明を対面で受けて、自宅で開催内の好きな時間に演習課題を解く。質問等はslackで行う。
対面	富山情報ビジネス専門学校 1年情報システム学科 22名	2022年2月8日 9:00～12:00		講師の指示のもと、演習課題を解く。質問等はその場で講師が回答する。

(13)遠隔教育プラットフォームの実証



課題ごとの完了数と品質の比較

形式	対象者	課題	完了		未完了		未着手
			バグなし	バグ修正後 再提出	バグあり	未提出	
遠隔	22人(※)	課題1(レベル1)	20人	0人	0人	2人	0人
		課題2(レベル1)	18人	0人	1人	3人	0人
		課題3(レベル2)	12人	0人	1人	9人	0人
		課題4(レベル2)	10人	1人	0人	11人	0人
対面	22人	課題1(レベル1)	21人	0人	0人	1人	0人
		課題2(レベル1)	19人	0人	0人	3人	0人
		課題3(レベル2)	8人	0人	0人	14人	0人
		課題4(レベル2)	0人	0人	0人	0人	22人

※ 23名中、1名がPCトラブルにより不参加となったため対象外とし対象者は22名

対面は環境作成に時間が掛かり、課題3、4の実施時間を十分に取れなかった。

(13)遠隔教育プラットフォームの実証

課題ごとの完了までに要した時間の比較



形式	課題	課題の完了者	課題の開始から完了（提出）になるまでに要した時間					
			～30分未満	30分～1時間未満	1時間～2時間未満	2時間～4時間未満	4時間～8時間未満	8時間以上
遠隔	課題1(レベル1)	20人	3人	7人	1人	4人	1人	4人
	課題2(レベル1)	18人	5人	2人	3人	4人	0人	4人
	課題3(レベル2)	12人	3人	7人	2人	0人	0人	0人
	課題4(レベル2)	11人	7人	1人	1人	0人	1人	1人
対面	課題1(レベル1)	21人	21人	0人	0人	0人	0人	0人
	課題2(レベル1)	19人	19人	0人	0人	0人	0人	0人
	課題3(レベル2)	8人	※1					
	課題4(レベル2)		※2					

※1 授業時間内で説明だけ行い、課題は宿題としたため計測不可
 ※2 授業時間内では時間が足りず、未実施

(13)遠隔教育プラットフォームの実証

課題結果の考察



- 対面では、授業内でレベル1の課題1、2とレベル2の課題3の説明までしか行えなかった。そのため、課題3については有志による宿題とし、課題4については未実施となった。
- レベル1の課題においては、課題1の完了者が対面では21/22人で遠隔が20/22人、課題2の完了者が対面では19/22人で遠隔18/22人となり、ほぼ同等の結果が得られた。レベル2の課題3の完了者が対面では8/22人で遠隔が12/22人、課題4の完了者は遠隔が11/22人(対面は未実施)だった。レベル2は、レベル1に比べ完了者が減少する傾向が見られ、対面の学生から「3問目の難易度が急に上がって難しかったです」という意見もあった。課題3の完了者数を比較すると遠隔のほうが多い結果となった。これは、対面は宿題だったことで先生のフォローはなし、遠隔はSlackでのフォローがあった差と見えるが、遠隔の学生から課題3に関する問い合わせは特になかったため、フォロー有無は影響しなかったと考えられる。また対面では課題3の提出を有志としたため完了者が減ったと思われるが、それを鑑みても遠隔での授業は対面と同等の成果があったと思われる。
- 遠隔と対面で課題完了までの時間を比較すると、対面は課題1、2ともに30分以内での完了(提出)に至ったが、遠隔では30分以内で完了できた学生は、課題1で3人、課題2で5人のみだった。このことから対面に比べ遠隔のほうが課題に時間を有したことが分かる。この要因としては、以下が挙げられる。
 1. 学生が課題に対してしっかりと時間をかけて取り組んでいた
 2. 課題を行う環境(プラットフォーム)に慣れていなかった
 3. 課題について理解しにくい、分からない、詰まった部分が多々あった
- 上記1.については、時間に縛られないという遠隔授業のメリットであり問題ないと考えられる。上記2.については、慣れるまでにある程度時間は必要と考えられるが、プラットフォームの使いやすさが向上することで改善される時間とも言える。上記3.については、アンケートで「教員への質問がしにくい」という意見があったことから、手が止まったときの対応(Slackを使って報告)がなかったため、結果的に放置または解決までに時間を要してしまっただと思われる。今後は、質問のし易さを改善することで、遠隔で課題に要する時間の短縮を図ることが可能と考えられる。

(13)遠隔教育プラットフォームの実証

フレームワークの理解度と課題の完了数の関係を比較



形式	アンケート回答者	課題の完了数	理解できなかった	あまり理解できなかった	理解できた	使えた	使いこなせた
遠隔	21名	0個	1人	1人	0人	0人	0人
		1個	0人	2人	0人	0人	0人
		2個	0人	2人	3人	0人	0人
		3個	0人	0人	0人	1人	0人
		4個	0人	3人	4人	4人	0人
対面	18名	0個	0人	0人	0人	0人	0人
		1個	0人	0人	1人	0人	0人
		2個	0人	4人	4人	3人	0人
		3個	0人	0人	4人	2人	0人
		4個	未実施				

(13)遠隔教育プラットフォームの実証

フレームワークの理解度と課題の完了数の関係の考察



- フレームワークを使ったプログラミングについて「理解できた」「使用できた」と回答した学生は、対面が77.8%(14/18人)で遠隔が57.1%(12/21人)だった。対面では、課題の1、2について全員実施できたが、課題3については宿題とし有志の学生のみ実施、課題4は未着手となってしまった。そのため、アンケートへの回答が課題1、2を終えた上での回答となる学生が多くなったと考えられ、「理解できた」「使用できた」の割合が増えた可能性があることを考慮する必要がある。また、遠隔では初期の段階で課題を4つ提示していたため、完遂できなかった学生については「理解できなかった」「あまり理解できなかった」を選ぶ割合が高まったことが考えられる。
- 遠隔で「理解できなかった」「あまり理解できなかった」と回答した学生の課題完了数を見ると0/4個だった学生が2人、1/4個できた学生が2人、2/4個できた学生が2人、4/4個できた学生が3人だった。4/4個できた学生が「あまり理解できなかった」と回答しているが、これはフレームワークに関する理解が進まないまま、課題に臨んでいたためだと考えられ、なぜフレームワークを利用するかという部分と各課題ごとの説明をより充実させていく必要がある。
- 0/4個～2/4個できた6名の学生が「理解できなかった」「あまり理解できなかった」と回答しているが、うち1名の学生からアンケートにて「プログラムの意味がいまいち理解できず、問題が解けない箇所があった。」との回答があったため、プログラム部分で躓いたものと思われる。加えて、Slackで質問や詰まっている旨のコメントはなかったため、Slackを活用した質問へのハードルが高かったものと思われる。これは初期段階でSlackを使った質問のチュートリアルやHow Toの資料を作成することで対応できるものと考えられる。他の5名の学生については、GitHub、Codespaces、Slackに対して、それぞれ「理解できない」、「あまり理解できなかった」と回答していることが多かった。このことから、プラットフォームのツール部分での躓きが学習へのモチベーションを下げてしまい、結果それがフレームワークの理解度の下げることにつながってしまったと考えられる。今後は、プラットフォームの利用をフォローする説明資料や動画の拡充を図る必要がある。

(13)遠隔教育プラットフォームの実証

対面と遠隔(Slackとチャットbot)の質問数と内容を比較



- 対面:計 35回
 - デバッグが動きません(同様の質問が15回)
 - 修正するソースの場所が分からない(同様の質問が7回)
 - 実行後、ポートの競合により画面が見れない(同様の質問が2回)
 - 何を提出すればよいか、また提出ファイルが分からない。(同様の質問が4回)
 - 課題にある例題の実行方法(URLの書き換え方について)が分からない(同様の質問が3回)
 - 編集対象のファイルの場所について(同様の質問が3回)
 - 修正対象となるファイルを教えてください
- 遠隔:11回(うち、チャットbotへの質問数は0回)
 - 自分の名前を選ぶ画面ってどこにありますか？ログインしたあとに違う画面がでます。
 - ログイン画面が表示されているんですけど、usernameがすいません、忘れてしまってどんな形式でした？
 - 一応手順通りやって終わりましたがこれで大丈夫ですかね？
 - いつでも実施していいとのことなんですけど基本的に期限などはありますか？
 - この画面になってしまいました(codespacesの上限)
 - インストールのダイアログが出てきません
 - すみません、私もインストールのダイアログが出ません。数回立ち上げなおしたんですけどダイアログが出なかったです。
 - (課題の実行に関して)これってどうやればいいですか
 - フィードバック失敗したんですけどどうしたらいいですか
 - すみません、この赤い×はどうしたら直りますか？
 - 虫と再生ボタンを押したのですがOpen in Browserが表示されません。どうしたらいいですか？

(13)遠隔教育プラットフォームの実証

対面と遠隔(Slackとチャットbot)の質問数と内容を比較



- 対面:計 35回
 - デバッグが動きません(同様の質問が15回)
 - 修正するソースの場所が分からない(同様の質問が7回)
 - 実行後、ポートの競合により画面が見れない(同様の質問が2回)
 - 何を提出すればよいか、また提出ファイルが分からない。(同様の質問が4回)
 - 課題にある例題の実行方法(URLの書き換え方について)が分からない(同様の質問が3回)
 - 編集対象のファイルの場所について(同様の質問が3回)
 - 修正対象となるファイルを教えてください
- 遠隔:11回(うち、チャットbotへの質問数は0回)
 - 自分の名前を選ぶ画面ってどこにありますか？ログインしたあとに違う画面がでます。
 - ログイン画面が表示されているんですけど、usernameがすいません、忘れてしまってどんな形式でした？
 - 一応手順通りやって終わりましたがこれで大丈夫ですかね？
 - いつでも実施していいとのことなんですけど基本的に期限などはありますか？
 - この画面になってしまいました(codespacesの上限)
 - インストールのダイアログが出てきません
 - すみません、私もインストールのダイアログが出ません。数回立ち上げなおしたんですけどダイアログが出なかったです。
 - (課題の実行に関して)これってどうやればいいですか
 - フィードバック失敗したんですけどどうしたらいいですか
 - すみません、この赤い×はどうしたら直りますか？
 - 虫と再生ボタンを押したのですがOpen in Browserが表示されません。どうしたらいいですか？

(13)遠隔教育プラットフォームの実証

アンケート結果の考察



- 教育プラットフォーム(GitHub、Codespaces、Slack)の利用時間について、平均で10時間20分(中央値9時間)という結果が得られた。対面での授業(3時間)と比較すると3倍以上の時間となった。この要因として、自分の好きなタイミングとペースで学習できていたことが挙げられる。普段の対面授業と今回の遠隔での授業を比べてみて良かった点のアンケートでこの意見が多く挙がっており、遠隔授業のメリットが活かせていたと言える。また別の要因として学生は今回初めてプラットフォーム利用するため資料確認、操作に慣れる、他の学生の状況を伺っていたなどである程度時間が増えたことが考えられる。
- GitHubについて「理解できた」「使用できた」と回答した学生は、81% (17/21人)に達した。また、実際に課題1と2では、それぞれ20人と18人の学生が課題に取り組みしており、GitHubについては一定の使いやすさ・分かり易さがあったと言える。これらの要因としては、GitHubに関する説明の資料で一定の理解が得られた、またはGitHubの操作(コミット、プッシュ)に関する手順が詳細に記載されていたことで、躓くことが少なかったことが考えられる。「理解できなかった」「あまり理解できなかった」と回答した学生については、GitHubだけでなくCodespacesやSlackに関する理解度も低い傾向にあった。このことから今回用意した資料や説明だけでは不十分だったとも考えられるため、事前作業段階で詳細な説明、もしくは遠隔での授業開始時にチュートリアルを設けるなどして、確実に触れる・動かせる状態にまでもっていく必要があると思われる。
- Codespacesについて「理解できた」「使用できた」と回答した学生は、61.9% (13/21人)で、GitHubの81%に比べると約19%も下回る結果となった。この要因としては、Codespacesを立ち上げた際に手順とは異なった動きをしたケースが多かったことが挙げられる。学生から手順にあるインストール時のダイアログが表示されない(再現性がなくいまだ原因不明)、Codespacesが立ち上がらない(1つの課題で複数個同時に立ち上げていたことが原因)という報告があった。GitHubでは、このようなイレギュラーな事象は特に発生しなかったことから、アンケートの理解度に差が生まれたものと思われる。また、約8割の学生がCodespacesを使って課題1と2を完了できていることから、Codespacesの操作については特に問題はなかったと言える。

(13)遠隔教育プラットフォームの実証

アンケート結果の考察



- Slackについて「理解できた」「使用できた」と回答した学生は、66.7% (14 /21人)で、「理解できなかった」「あまり理解できなかった」と回答した学生は33.3%(7 /21人)だった。「理解できなかった」「あまり理解できなかった」と回答した7名のうち、Slackの利用(コメントやリアクション)が何もなかった学生は2人だった。これらの学生への対応としては、事前作業の段階で、Slackのチュートリアルを実施し、確実に触れてもらうことで、利用へのハードルを下げるのが有効と考える。Slackを利用はされたが、「理解できなかった」「あまり理解できなかった」と回答した学生については、実証実験の中で誤った運用(課題1に関する質問を課題4のところへコメント)をする学生がいたことから、Slackを含むプラットフォームを使う上でルールが分からず、結果それが理解度の低下に繋がったと思われる。プラットフォームの利用する上でのルールを明確にした上で、その場所について周知(もしくは固定)することで解消できるものとする。
- 対面と遠隔の質問数を比較したところ、対面が3時間で35回に対して遠隔は4日間で11回だった。遠隔が大きく質問の回数を減らした要因として、Slackの特徴であるやり取りした内容が履歴として残り共有されたことで、同じ質問を人が減ったことが挙げられる。学生からのアンケートでも同様の回答があり、これはSlackをうまく活用できていたと言える。その一方で、対面に比べて質問しづらかったという意見も見受けられたため、先述したSlackを利用する上でのハードルを下げる試みが必要と考える。
また、今回Slackに学生からの質問に回答するチャットbotを設けたが、受け付けた質問は0件だった。これは、チャットbotに何を質問すればいいかが分からず、先生に聞いてしまったほうが早いとなってしまったと考えられる。今後、チャットbotを運用していくにあたり、今回の検証で学生から出た質問をチャットbotで回答できるようすること、またその質問のQA例を提示して利用方法を周知するような対応を行う必要がある。



5. 事業評価

成果



- 利用費、運用費が無料のプラットフォームを開発することができた
- 導入の手間の少ないプラットフォームを開発することができた
- 対面と同等な教育効果を出すことができた
- フレームワーク利用したプログラミングを学生に理解させることができた
- プラットフォームに対する学生の参加性の高さを確認できた

課題と対応



- 遠隔教育プラットフォームを利用できない学生が少数ながらいた
→利用説明等の動画を強化する
- 教員－学生間に二通りのコミュニケーション手段を使用している
→企業での標準形式を優先するか、教育を優先するか検討する
- 企業は学生に高いプログラミングスキルを求めている
→企業での即戦力レベルのプログラミングスキルレベルを策定する



6. 令和4年度事業内容

令和4年度事業内容



事業分野	内容
調査	<ul style="list-style-type: none">• 企業での即戦力レベルのプログラミングスキルの調査
開発	<ul style="list-style-type: none">• 演習ユニットの完成 動画、プログラミング規約、品質管理教材の追加• 学習管理ユニットの完成 チャットボットの対応項目の追加、チーム開発機能の強化• 実習ユニットのプロトタイプ
実証	<ul style="list-style-type: none">• 演習ユニット、学習管理ユニットの検証 全国3校程度で実施• 実習ユニットの検証

専門学校におけるプログラム開発
教育に関する状況調査

専門学校におけるプログラム開発教育に関する状況調査

学校法人浦山学園 富山情報ビジネス専門学校

本アンケートは、プログラマー育成に関する状況を統計的に調査し、学習領域、教育方法、評価方法等を整理する目的で実施するものです。専門学校で実施されているプログラマー育成方法とプロのプログラマーとして必要とされている能力を比較する際の基礎データとして活用します。その結果につきましては、本事業完了時にフィードバックさせていただきます。

アンケートのご記入には、プログラミングのリモート教育を実施している学校様においては概ね 20 分程度、実施していない学校様においては概ね 10 分程度のお時間を要するかと思います。お忙しいところ恐縮ですが、ご協力をお願い申し上げます。

(1) 全校共通

- ① 開発実践力習得のために、学習に力を入れているプログラミング言語は何ですか？
複数選択可
JAVA、C、C++(C#含む)、Python、その他()
- ② 検定取得のために、学習に力を入れているプログラミング言語は何ですか？
複数選択可
JAVA、C、Python、アセンブラ、その他()
- ③ 文法理解以外のプログラミング演習内容とその時間数は？
複数選択可
教科書課題の演習、独自課題の演習、PBL、企業連携課題の演習
時間数()
- ④ 変数の命名規則やコメントの付け方などコーディングルールを決めて演習していますか？
◆Yes/No
- ⑤ プログラミングの際は事前にプログラム仕様書を作成していますか？
◆Yes/No ※Yesの場合→⑥ Noの場合→⑦
- ⑥ プログラム仕様書を作成してプログラミングを実施している場合、仕様書は教員が作成しますか、学生が作成しますか？
◆教員が作成する/学生が作成する
- ⑦ テスト仕様書を作成してテストしていますか？
◆Yes/No ※Yesの場合→⑧ Noの場合→⑨
- ⑧ テスト仕様書を作成してテストを実施している場合、仕様書は教員が作成しますか、学生が作成しま

すか？

◆教員が作成する/学生が作成する

- ⑨ システムの一部となる部分のプログラムを作らせ、全体に組み込むような実習形態はありますか？
◆ある/ない
- ⑩ 学生から提出されたプログラミング課題の確認をどのようにおこなっていますか？
複数選択可
ソースファイルを確認
実行ファイルを確認
実行結果を画像(スリーンショット)で提出
入力状況を画面録画して動画形式で提出
その他()
- ⑪ 学生から提出されたプログラミング課題の内容をどのように学生にフィードバックしていますか？
複数選択可
対面で個別フィードバック
メール、SNS等でフィードバック
学習管理システムでフィードバック
その他()
- ⑫ 学生に与える課題のボリューム感はどの程度で設定していますか？
授業時間内に終わる
授業時間外に多少の時間確保が必要
授業時間外に大幅な時間確保が必要
- ⑬ プログラミング課題が全くできない、手が動かないという学生に対するフォローはどのように行っていますか？
複数選択可
授業内でマンツーマン指導
授業外でマンツーマン指導
補習授業
学生間の教え合い
動画等の補助教材を活用
その他()
- ⑭ プログラミング課題を宿題として学校外で取り組ませていますか？
◆取り組ませている/取り組ませない ※取り組ませている場合→⑮
-

- ⑮ プログラミング課題を宿題として自宅で行っているとした場合、学生の疑問・質問はどのように対応していますか？

複数選択可

対面で個別対応

メール、SNS等で対応

学習管理システムで対応

その他()

- ⑯ 一般的に提供されている Web 学習サービス(paiza ラーニング等)を導入されていますか？

◆Yes/No Yes の場合、採用しているサービス名と方法

◆形式:

希望者が自発的に取り組む

学校で推奨している

希望者に学校から配布する

全員に取り組ませている

- ⑰ 開発プロジェクト(卒業制作等)は学生が個々で運営していますか、グループで運営していますか？

◆個人/グループ

- ⑱ どのような開発手法をプロジェクトで採用していますか？

◆ウォーターフォール型/アジャイル型/プロトタイプ型/スパイラル型/その他

- ⑲ エンドユーザーを想定して、開発したプログラムを利用したフィードバックや追加の改修を行う事まで実施していますか？

◆Yes/No

- ⑳ プログラミングを含むシステム開発等で企業と連携していますか？連携している場合、以下のことについて教えてください。

◆連携の種類:講師派遣/コンテンツ提供/コンテスト開催/教員支援

◆実施形式:対面/リモート/ハイブリッド/インターンシップ

◆実施頻度:定期的(時間割に組み込む)/イベント的(スポット開催)

◆企業連携の課題:企業選定/講師料/講師確保/その他()

(2)プログラミングのリモート教育を実施「している」学校

プログラミングのリモート教育を実施「している」学校は、以下のアンケートにもご協力をお願いいたします。実施していない学校は、(3)に進んでください。

- ① リモート教育に伴い、新たに導入したツールはありますか？
◆Yes/No Yes の場合、導入したツール名
- ② リモート教育に伴い、新たに利用することになった学習管理システムはありますか？
◆Yes/No Yes の場合、利用しているシステム名
- ③ リモート教育に伴い、ペアプログラミングを実施または検討していますか？
◆実施中/検討中/どちらもしていない
- ④ リモート教育に伴い、学習習熟度や検定取得率に変化はありましたか？また、その理由として考えられることを教えてください。
◆変化あり/変化なし 理由:
- ⑤ リモート教育に伴い、開発実践力取得レベルに変化はありましたか？また、その理由として考えられることを教えてください。
◆変化あり/変化なし 理由:
- ⑥ リモート教育に伴い、行き詰っている学生の把握やアドバイスはどのようにしていますか？
複数選択可
対面で個別対応
メール、SNS等で対応
学習管理システムで対応
その他()
- ⑦ リモート教育に伴い、対面教育と比べてボリュームに変化がありましたか？
◆増えた/減った/変化はなかった
- ⑧ リモート教育に伴い、対面教育と比べてボリュームに変化のあった場合、課題は全員が完了するレベル感の内容ですか？
◆完了する/完了しない
- ⑨ リモート教育の実施有無を広報情報として利用していますか？
◆Yes/No

(3)プログラミングのリモート教育を実施「していない」学校

- ① プログラミングのリモート教育を実施しない原因または障害はどんなことですか？
必要性が無い
-

環境が無い
ノウハウが無い
対応できる教員がいない
準備する時間が無い

以上です。ご協力ありがとうございました。

専門学校におけるプログラム開発教育 に関する状況調査

アンケート調査の概要

調査名	専門学校におけるプログラム開発教育に関する状況調査
調査目的	プログラマー育成に関する状況を統計的に調査し、学習領域、方法、評価方法等を整理する。
調査対象	職業実践専門学校課程の認定を受けたIT系学科を設置する専門学校107校
調査手法	Google formによるネットワークアンケート調査
調査項目	教育領域、教育設計、実施方法、学習評価方法、遠隔教育実施概要 遠隔教育事例等
分析内容 (集計項目)	上記の調査項目について、統計的に集計し、専門学校で行われているプログラマー育成学習の概要を整理する。
活用手法	専門学校で実施されているプログラマー育成とプロのプログラマーとして必要としている能力を比較する際の基礎データとする。

アンケート調査の概要

【アンケート発送日】

2021年11月12日(金)

内容:郵送によるアンケートページ(Google form)のご案内

【アンケート発送件数】

全国専門学校 107校

【アンケート回答件数】

送付先107校中**24校**(22.4%)

文部科学省委託事業

回答のあった専門学校・・・24校

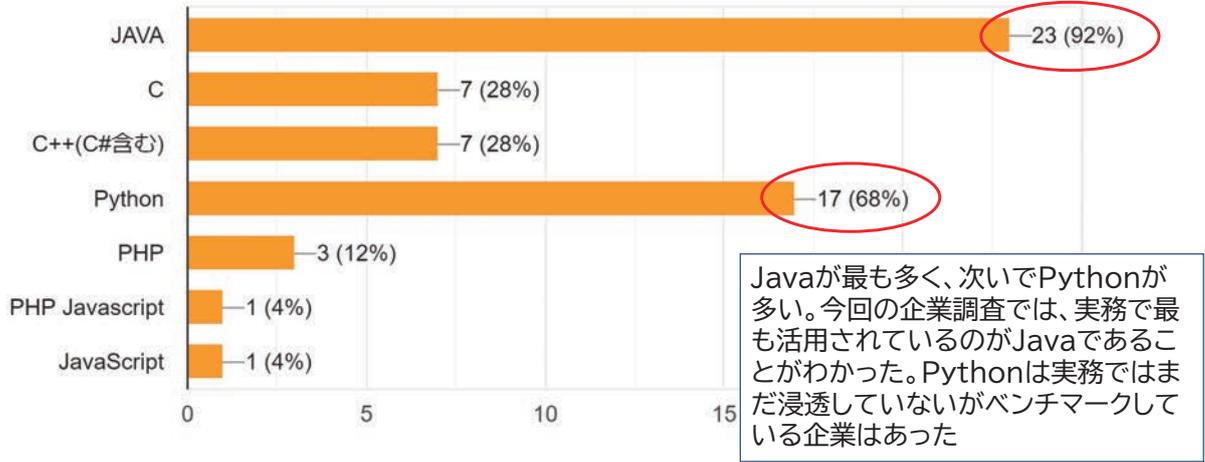
- ・ アルスコンピュータ専門学校
- ・ 穴吹ビジネス専門学校
- ・ 富山情報ビジネス専門学校
- ・ 日本電子専門学校
- ・ 専門学校徳島穴吹カレッジ
- ・ 麻生情報ビジネス専門学校
- ・ 熊本電子ビジネス専門学校
- ・ 大原簿記情報医療専門学校
- ・ YIC情報ビジネス専門学校
- ・ 浜松未来総合専門学校
- ・ 専門学校中央情報大学校
- ・ 東北電子専門学校
- ・ 専門学校岡山情報ビジネス学院
- ・ 船橋情報ビジネス専門学校
- ・ 専門学校アレック情報ビジネス学院
- ・ 新潟高度情報専門学校
- ・ 学校法人情報文化学園アーツカレッジヨコハマ
- ・ 河原電子ビジネス専門学校
- ・ 宮崎情報ビジネス医療専門学校
- ・ 盛岡情報ビジネス&デザイン専門学校
- ・ 中央情報専門学校
- ・ 甲府市立甲府商科専門学校
- ・ KCS北九州情報専門学校
- ・ 専門学校穴吹コンピュータカレッジ

文部科学省委託事業

(1) 全校共通

① 開発実践力習得のために、学習に力を入れているプログラミング言語は何ですか？(複数選択可)

25 件の回答

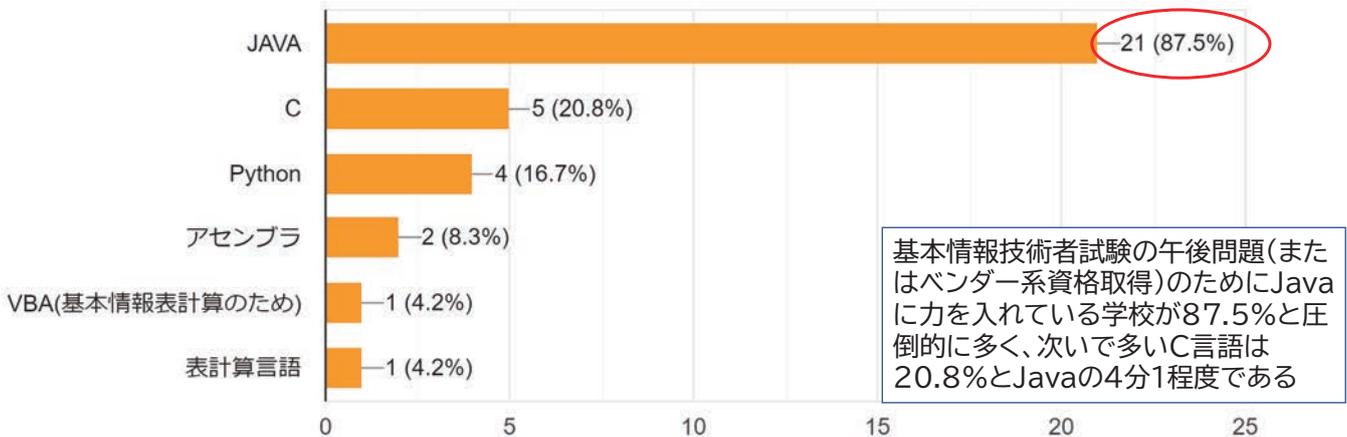


文部科学省委託事業

(1) 全校共通

② 検定取得のために、学習に力を入れているプログラミング言語は何ですか？(複数選択可)

24 件の回答

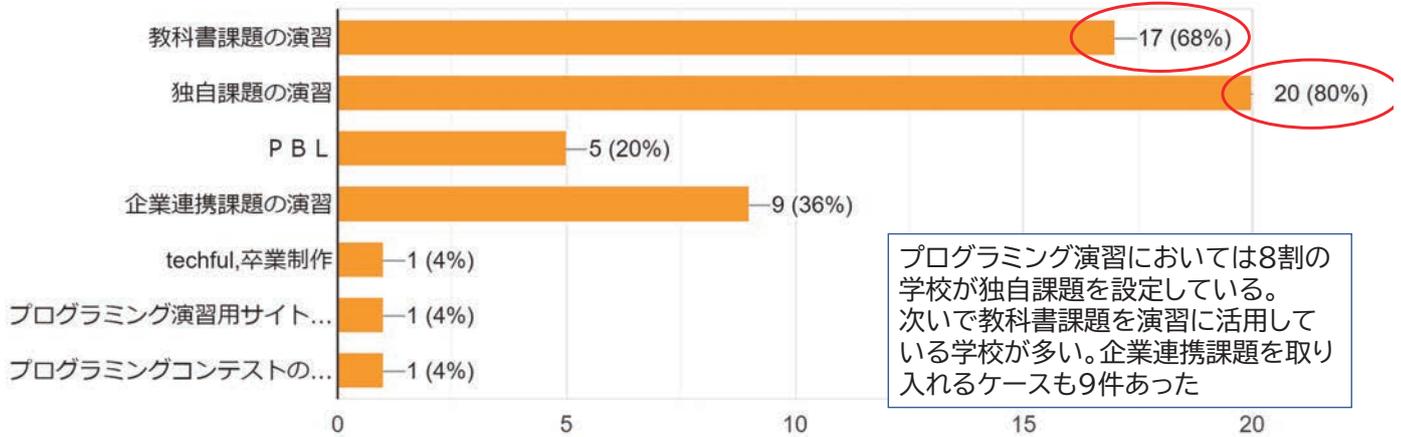


文部科学省委託事業

(1) 全校共通

③ 文法理解以外のプログラミング演習内容は？(複数選択可)

25 件の回答



文部科学省委託事業

(1) 全校共通

③-2 文法理解以外のプログラミング演習の時間数は？ 22 件の回答

- 100時間
- 144時間
- 240時間
- 10時間/週程度(状況による)
- 週6時間
- 週30時間
- 週30時限中20時限程度
- 300時間
- 全体の50%
- 週20時間
- 360時間
- 半年で50時間程度
- 100時間以上
- 週3時間
- 週で2限分
- 学科によって異なります。開発系の2年課程なら、1年次114時間、2年次342時間です。
- 約80時間
- 400h程度
- 240時間
- およそ800時間前後
- 136時間
- 6コマ(1コマ70分)

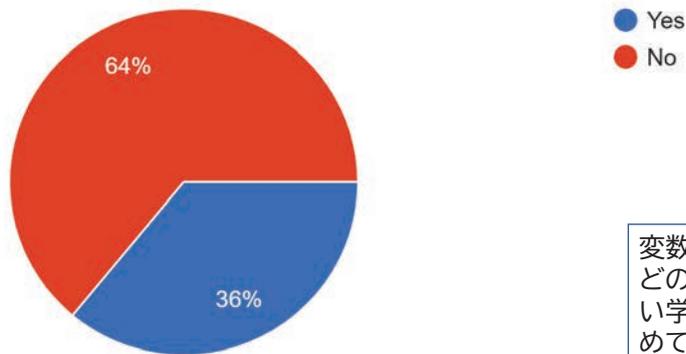
文法理解以外のプログラミング演習時間は100時間から800時間まで学校によって幅がある。

文部科学省委託事業

(1) 全校共通

④ 変数の命名規則やコメントの付け方などコーディングルールを決めて演習していますか？

25 件の回答



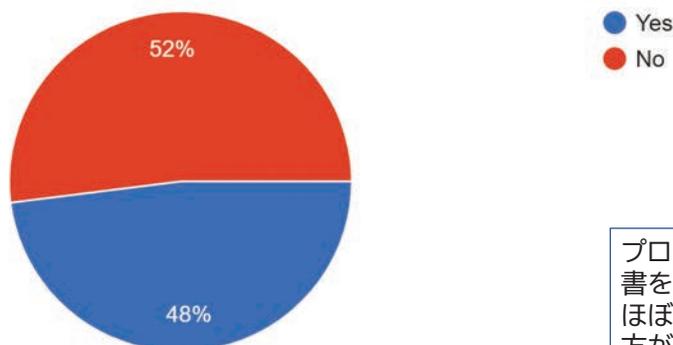
変数の命名規則やコメントの付け方などのコーディングルールは決めていない学校の方が多く、25校中16校が決めていない

文部科学省委託事業

(1) 全校共通

⑤ プログラミングの際は事前にプログラム仕様書を作成していますか？

25 件の回答



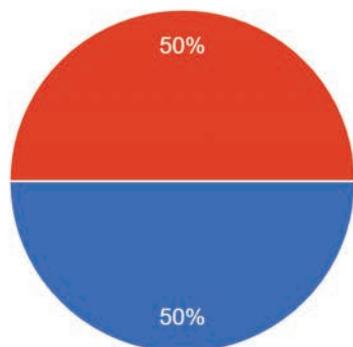
プログラミングの際にプログラム仕様書を作成する学校と作成しない学校はほぼ半数ずつだが、作成しない学校の方が1校多い

文部科学省委託事業

(1) 全校共通

⑥ プログラム仕様書を作成してプログラミングを実施している場合、仕様書は教員が作成しますか、学生が作成しますか？

14 件の回答



● 教員が作成する
● 学生が作成する

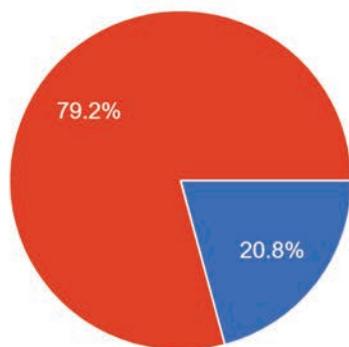
仕様書を作成する場合、学生が作るケースと教員が作るケースはちょうど50%ずつである。つまり約4分の1の7校では学生が仕様書を作成している

文部科学省委託事業

(1) 全校共通

⑦ テスト仕様書を作成してテストしていますか？

24 件の回答



● Yes
● No

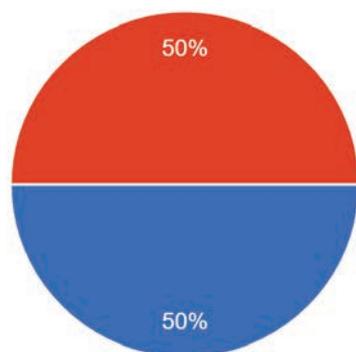
テスト仕様書を作成していない学校が多く約8割(19校)を占めている。一方、2割(5校)ではテスト仕様書を作成している

文部科学省委託事業

(1) 全校共通

⑧ テスト仕様書を作成してテストを実施している場合、仕様書は教員が作成しますか、学生が作成しますか？

8 件の回答



● 教員が作成する
● 学生が作成する

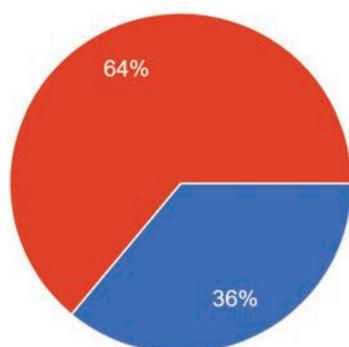
テスト仕様書を作成している学校では、学生が仕様書を作るケースと教員が作るケースでちょうど半数ずつである

文部科学省委託事業

(1) 全校共通

⑨ システムの一部となる部分のプログラムを作らせ、全体に組み込むような実習形態はありますか？

25 件の回答



● ある
● ない

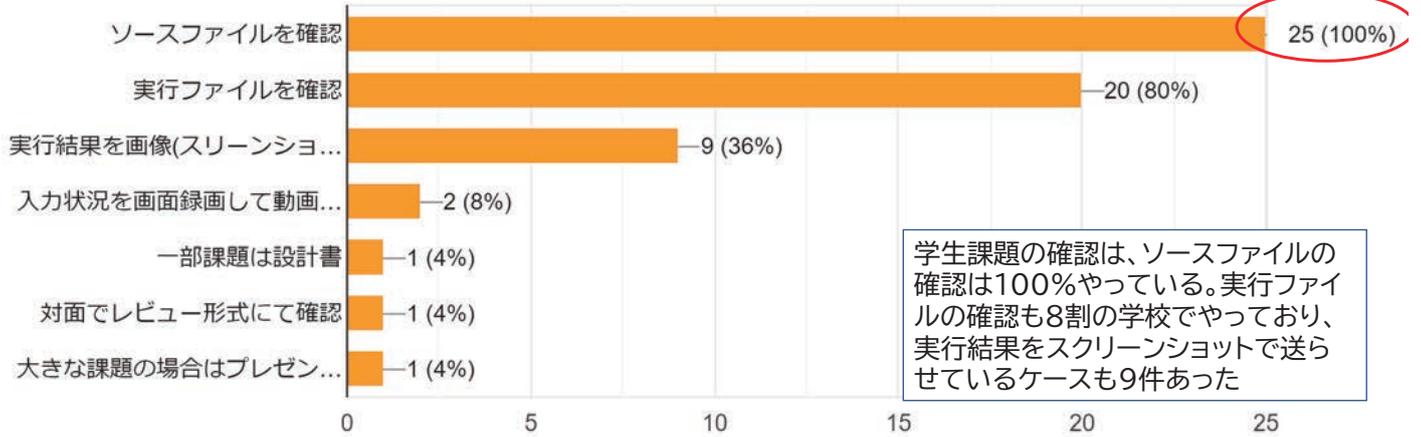
システムの一部となる部分のプログラムを作らせ、全体に組み込むような実習はやっていない学校のほうが多く、やっている学校は9校である

文部科学省委託事業

(1) 全校共通

⑩ 学生から提出されたプログラミング課題の確認をどのようにおこなっていますか？(複数選択可)

25 件の回答

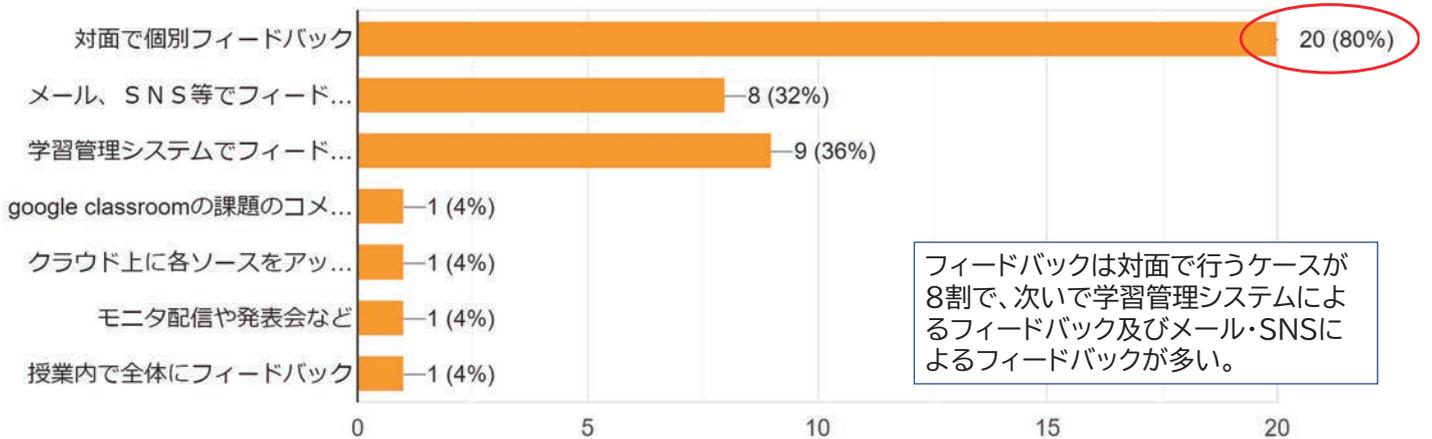


文部科学省委託事業

(1) 全校共通

⑪ 学生から提出されたプログラミング課題の内容をどのように学生にフィードバックしていますか？(複数選択可)

25 件の回答

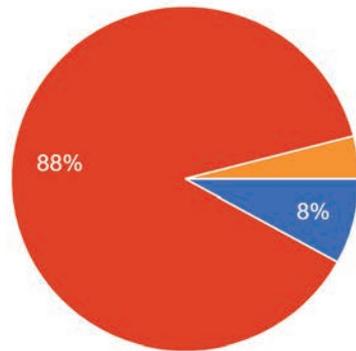


文部科学省委託事業

(1) 全校共通

⑫ 学生に与える課題のボリューム感はどの程度で設定していますか？

25 件の回答



- 授業時間内に終わる
- 授業時間外に多少の時間確保が必要
- 授業時間外に大幅な時間確保が必要

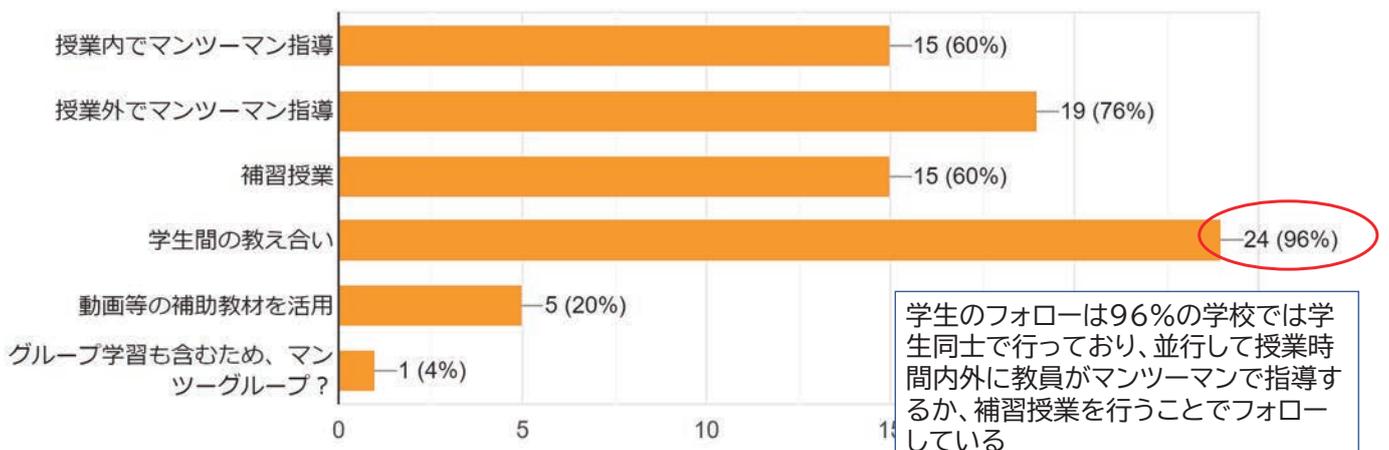
学生に与える課題のボリューム感は、授業時間外にも取り組みを要するボリュームで与えている学校が8割を超える(22校)

文部科学省委託事業

(1) 全校共通

⑬ プログラミング課題が全くできない、手が動かないという学生に対するフォローはどのように行っていますか？(複数選択可)

25 件の回答

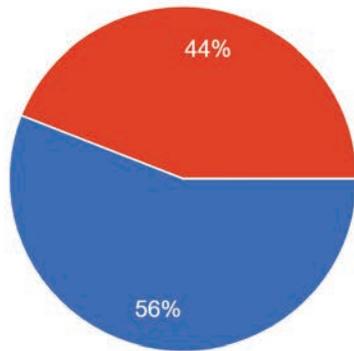


学生のフォローは96%の学校では学生同士で行っており、並行して授業時間内外に教員がマンツーマンで指導するか、補習授業を行うことでフォローしている

文部科学省委託事業

(1) 全校共通

⑭ プログラミング課題を宿題として学校外で取り組ませていますか？
25 件の回答



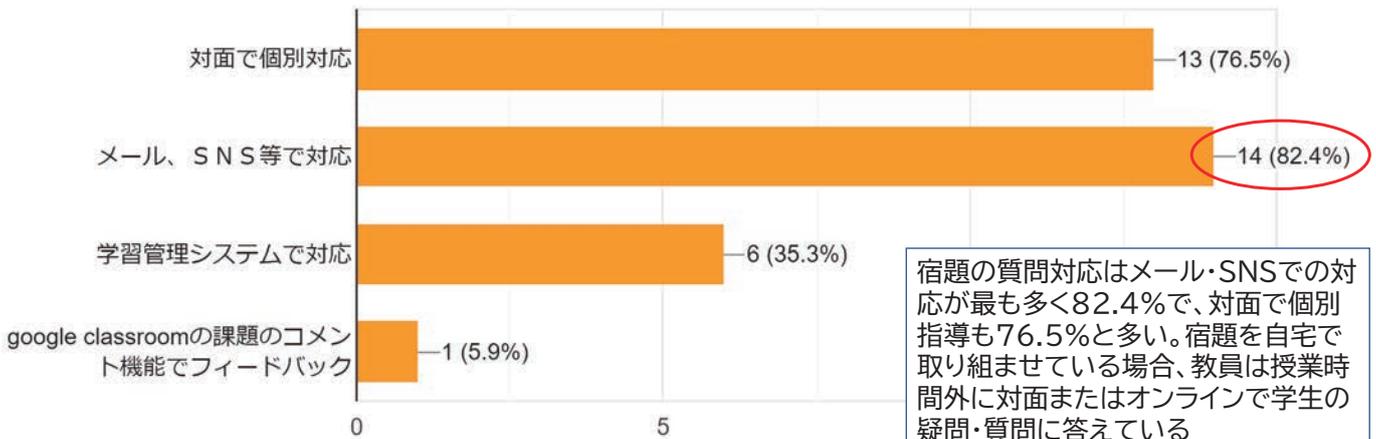
● 取り組ませている
● 取り組ませない

プログラミング課題を宿題として学校外でも取り組ませている学校が3校多く14校で、取り組ませない学校が11校である

文部科学省委託事業

(1) 全校共通

⑮ プログラミング課題を宿題として自宅に取り組ませているとした場合、学生の疑問・質問はどのように対応していますか？(複数選択可)
17 件の回答



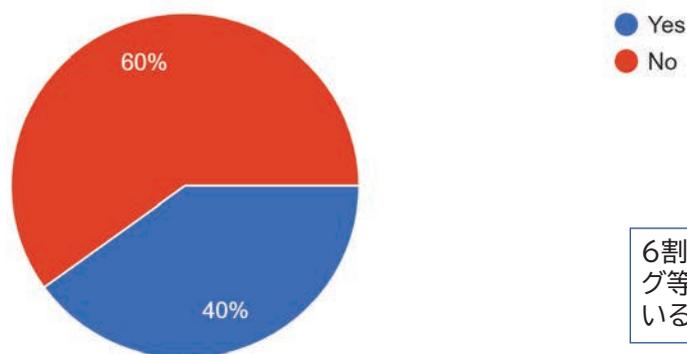
宿題の質問対応はメール・SNSでの対応が最も多く82.4%で、対面で個別指導も76.5%と多い。宿題を自宅で行っている場合、教員は授業時間外に対面またはオンラインで学生の疑問・質問に答えている

文部科学省委託事業

(1) 全校共通

⑩ 一般的に提供されているWeb学習サービス(paizaラーニング等)を導入されていますか？

25 件の回答



6割の学校(15校)でpaizaラーニング等のWeb学習サービスを活用している。

文部科学省委託事業

(1) 全校共通

⑩-2 上記Yesの場合、採用しているサービス名と方法

10 件の回答

- Paiza 7件
- TechFul 6件
- Moodle 1件
- AtCoder 1件
- 本校独自のWeb学習サービス 1件
- Teams 1件

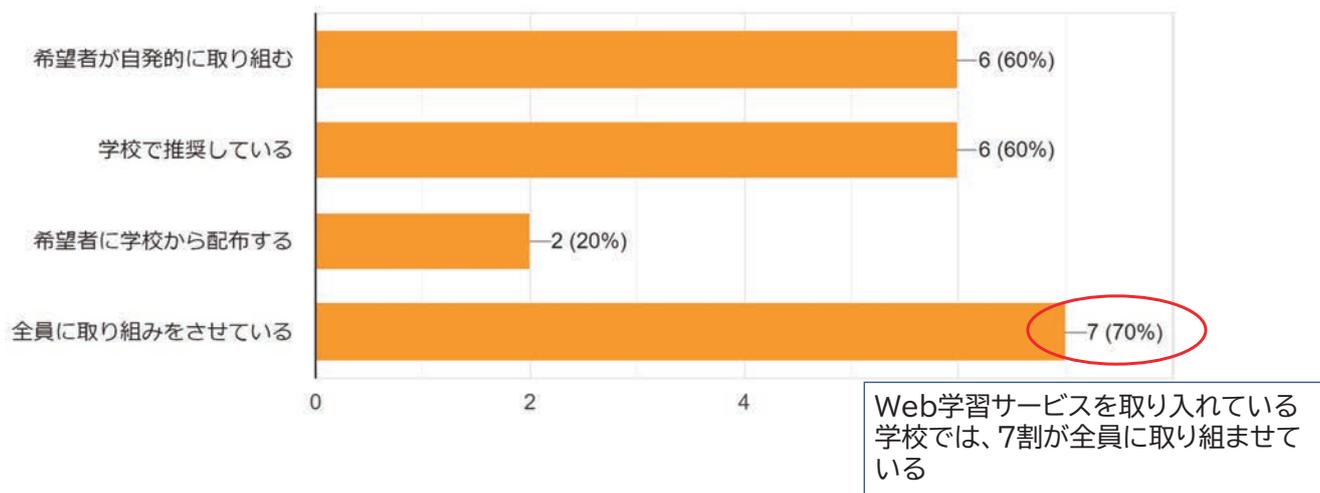
Web学習システムを使っている事例ではPaizaが最も多く7件で、次いでTechFulが6件となっている。本校独自のWeb学習サービスというものも1件あった(麻生情報ビジネス専門学校)

文部科学省委託事業

(1) 全校共通

⑩-3 上記採用しているサービスの利用方法

10 件の回答

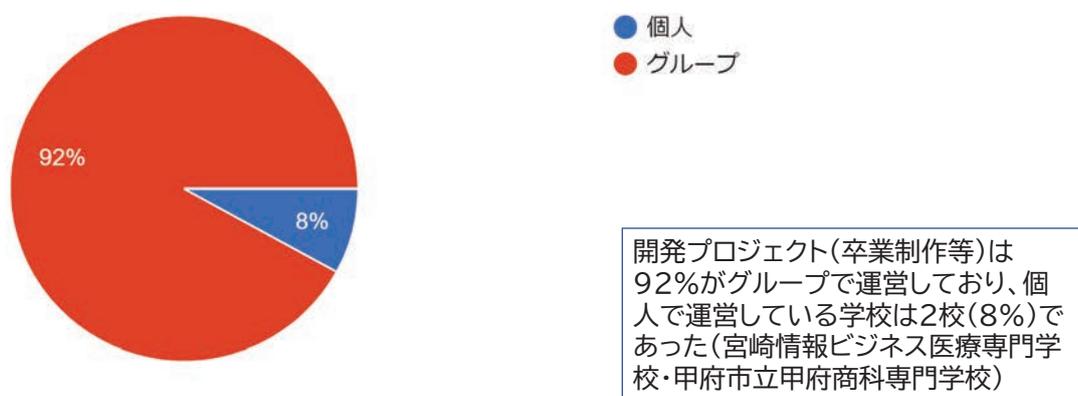


文部科学省委託事業

(1) 全校共通

⑰ 開発プロジェクト(卒業制作等)は学生が個々で運営していますか、グループで運営していますか？

25 件の回答

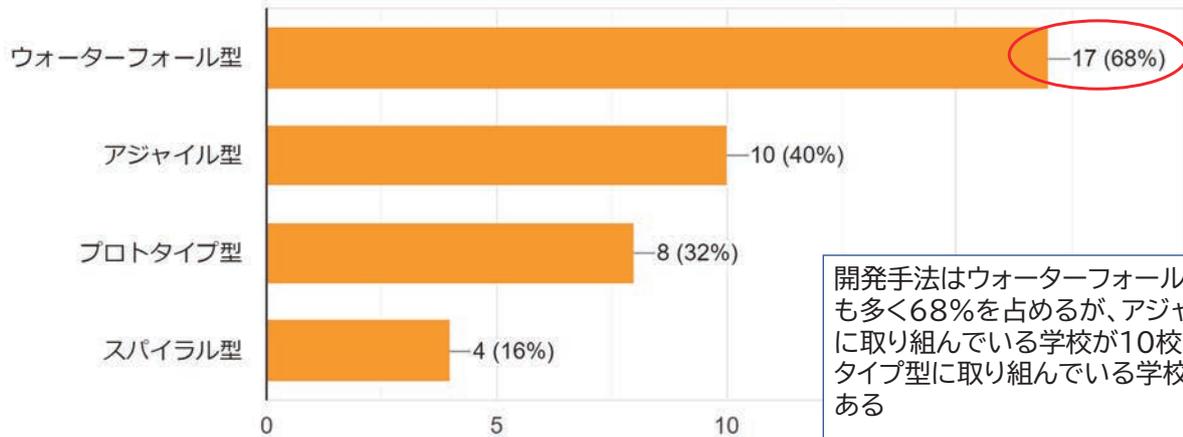


文部科学省委託事業

(1) 全校共通

⑱ どのような開発手法をプロジェクトで採用していますか？

25 件の回答

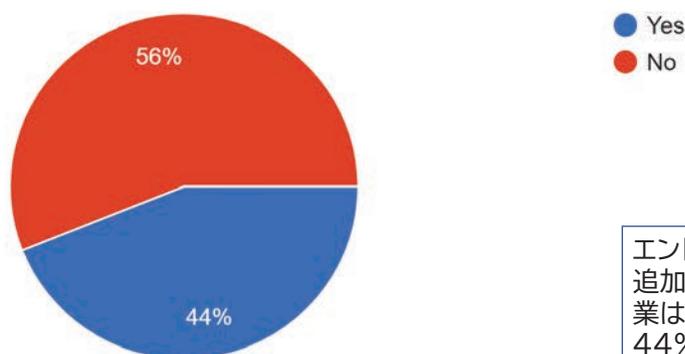


文部科学省委託事業

(1) 全校共通

⑲ エンドユーザーを想定して、開発したプログラムを利用したフィードバックや追加の改修を行う事まで実施していますか？

25 件の回答



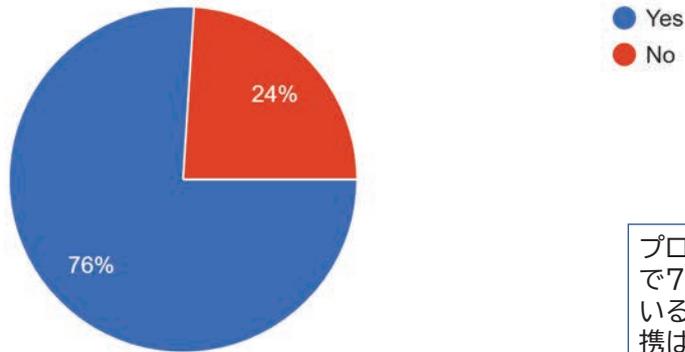
エンドユーザーによるフィードバックや追加の改修まで想定した実践的な授業は56%(14校)がやっていないが、44%(11校)ではやっている

文部科学省委託事業

(1) 全校共通

⑳ プログラミングを含むシステム開発等で企業と連携していますか？

25 件の回答



プログラミングを含むシステム開発等で76%(19校)が企業との連携をしているが、24%(6校)では企業との連携はしていない

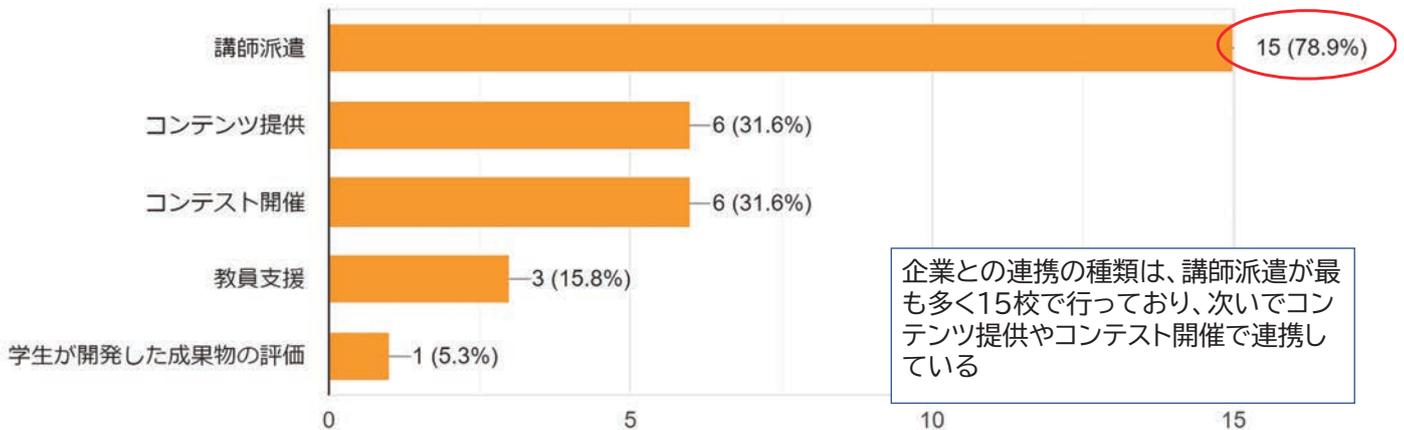
文部科学省委託事業

(1) 全校共通

⑳-1 プログラミングを含むシステム開発等で企業と連携していますか？

◆◆◆連携の種類◆◆◆

19 件の回答



企業との連携の種類は、講師派遣が最も多く15校で行っており、次いでコンテンツ提供やコンテスト開催で連携している

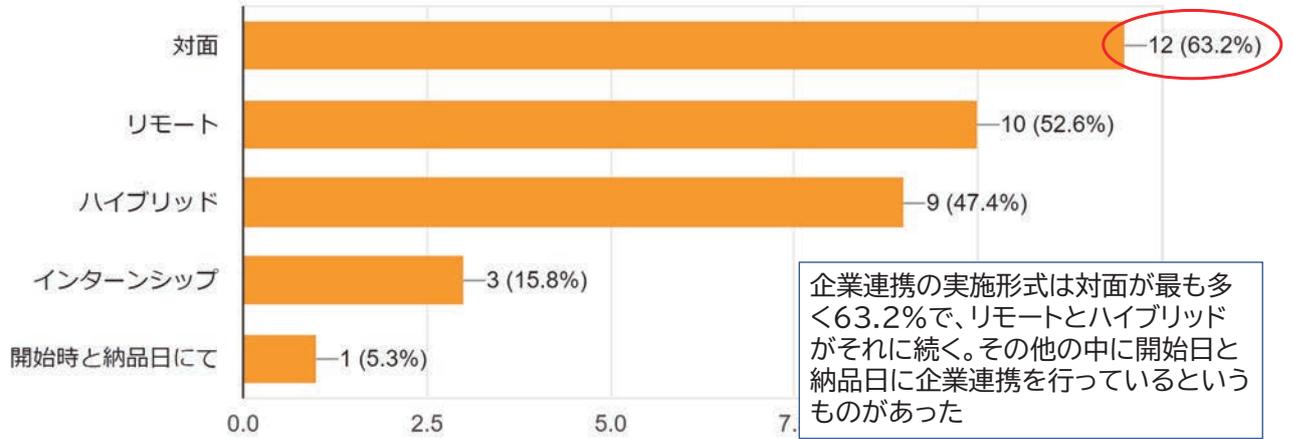
文部科学省委託事業

(1) 全校共通

⑳-2 プログラミングを含むシステム開発等で企業と連携していますか？

◆◆◆実施形式◆◆◆

19 件の回答



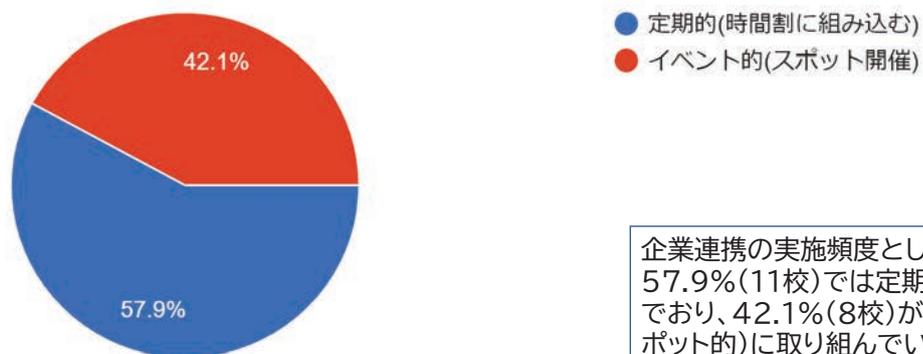
文部科学省委託事業

(1) 全校共通

⑳-3 プログラミングを含むシステム開発等で企業と連携していますか？

◆◆◆実施頻度◆◆◆

19 件の回答



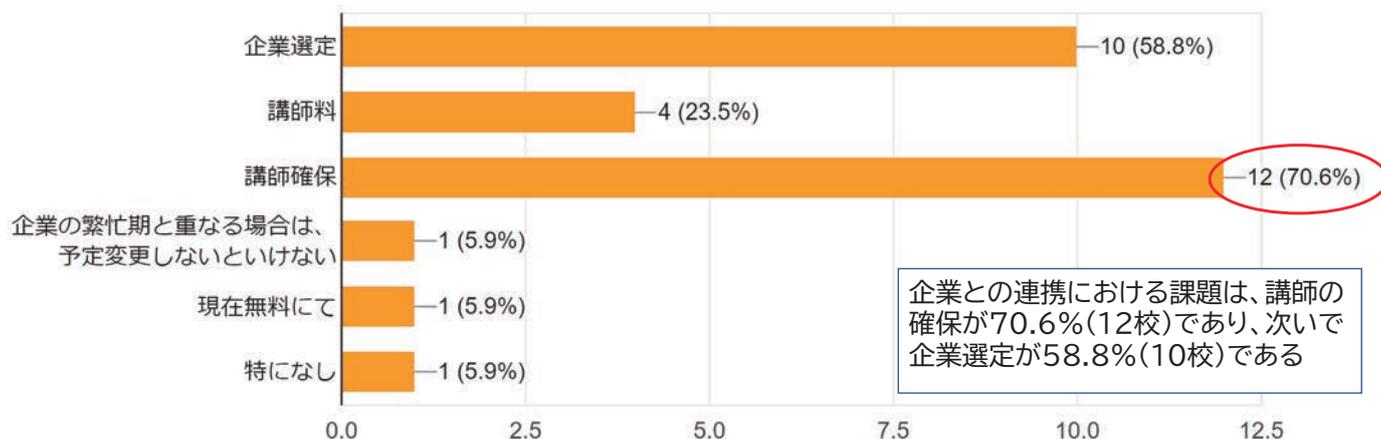
文部科学省委託事業

(1) 全校共通

⑳-4 プログラミングを含むシステム開発等で企業と連携していますか？

◆◆◆企業連携の課題◆◆◆

17件の回答



文部科学省委託事業

(1) 全校共通

全校共通質問事項に関するまとめ(1)

【開発言語】

- 開発実践力習得のために、92%の学校がJavaに力を入れている。
- 次いで68%の学校でPythonが選ばれている
- 87.5%の学校で、検定取得のためにJavaの学習に力を入れている

【演習方法・時間】

- 80%の学校で演習には独自課題を設定している。
- 次いで68%の学校で教科書課題を演習に取り入れている。
- 演習時間は学校によって大きく差があり、100時間から800時間と幅がある

【演習内容】

- 演習の際に64%の学校がコーディングルールを決めていない
- 演習の際に52%の学校がプログラム仕様書を作成していない
- プログラム仕様書を作成している学校のうち50%で、学生が仕様書を作っている

文部科学省委託事業

(1) 全校共通

全校共通質問事項に関するまとめ(2)

【テスト演習】

- 79.2%の学校ではテスト仕様書を作成していない
- テスト仕様書を作成している20.8%の学校のうち50%で学生がテスト仕様書を作成している

【組み込みプログラムの演習】

- 64%の学校でシステムの一部となる部分のプログラムを全体に組み込む演習はやっていない

【学生課題の確認方法】

- ソースファイルの確認は100%の学校で行っている
- 80%の学校で実行ファイルの確認も行っている
- 課題内容の学生へのフィードバックは、80%の学校で対面で個別で行っている

【学生に与える課題のボリューム】

- 88%の学校では授業時間外に多少の時間の確保が必要な量を与えている

文部科学省委託事業

(1) 全校共通

全校共通質問事項に関するまとめ(3)

【課題フォロー体制】

- プログラミング課題が全くできない、手が動かない学生のフォローは、96%の学校が学生同士で行っており、並行して授業時間内外に教員がマンツーマンで指導するか、補習授業で行っている
- 56%の学校でプログラミング課題を宿題として学校外でも取り組ませている
- 宿題の質問対応はメール・SNSでの対応が最も多く82.4%で、対面で個別指導も76.5%と多い

【Web学習サービスの利用状況】

- 60%の学校ではWeb学習サービスを導入していない
- 40%のWeb学習サービスを導入している学校では、Paiza(7件)、TechFUL(6件)を導入しており、独自のWeb学習サービスを導入している学校も1校ある(麻生情報ビジネス専門学校)
- Web学習サービスを導入している学校では、70%が全員に取り組みをさせている

文部科学省委託事業

(1) 全校共通

全校共通質問事項に関するまとめ(4)

【開発プロジェクト(卒業制作)について】

- 開発プロジェクト(卒業制作)は**92%**の学校がグループで運営している
- 開発手法は**68%**でウォーターフォール型を採用しており、アジャイル型も**40%**で採用されている
- **56%**の学校ではエンドユーザーを想定したフィードバックや追加の改修は行っていない

【企業連携】

- **76%**の学校ではシステム開発等で企業と連携している
- 連携の種類は講師派遣が最も多く、**78.9%**の学校で実施している
- 次に多い企業連携の種類は、コンテンツ提供とコンテスト開催でどちらも**31.8%**の学校で実施している
- 企業連携の実施形式としては**63.2%**が対面であり、リモートも**52.6%**の学校で実施している
- 実施頻度としては**57.9%**が定期的(時間割に組み込む)に実施している
- 企業連携における課題は、**70.6%**の学校で講師確保が課題となっており、次いで**58.8%**の学校で企業選定が課題となっている

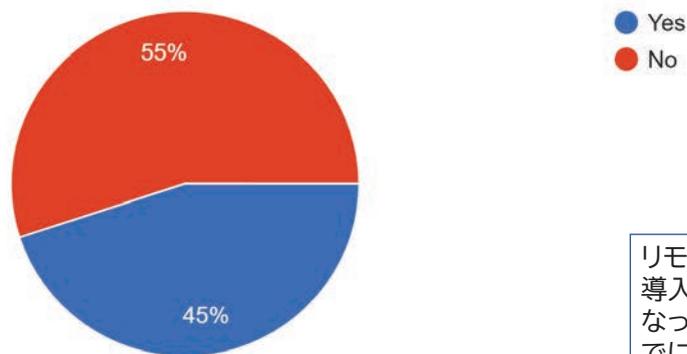
文部科学省委託事業

(2) プログラミングのリモート教育を実施 「している」学校への質問

(2)プログラミングのリモート教育を実施「している」学校

① リモート教育に伴い、新たに導入したツールはありますか？

20 件の回答



リモート教育に伴い、新たにツールを導入した学校は45%と半数以下となっている。リモート教育の環境はすでに整っていたと推測できる

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

①-2 Yesの場合のツール名

9 件の回答

- Microsoft Teams 6件
- Zoom 3件
- Webex 2件
- Moodle 1件
- Slack 1件
- GoogleMeet 1件
- グーグルアプリケーション 1件
- webex-meeting 1件

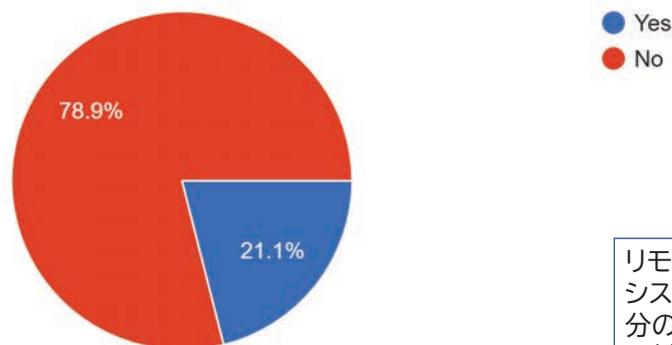
リモート教育ツールはMicrosoft Teamsが6件と最も多く、次いでzoomが3件であった。教育現場ではMicrosoft Teamsの活用が進んでいると推測できる

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

② リモート教育に伴い、新たに利用することになった学習管理システムはありますか？

19 件の回答



リモート教育に伴い、新たに学習管理システムを導入した学校は21.1%と4分の1以下となっている。リモート教育に対応する学習管理システムはすでに導入されていたと推測できる

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

②-2 上記Yesの場合、利用しているシステム名

5 件の回答

- GoogleClassroom 5件
- Google meet 1件

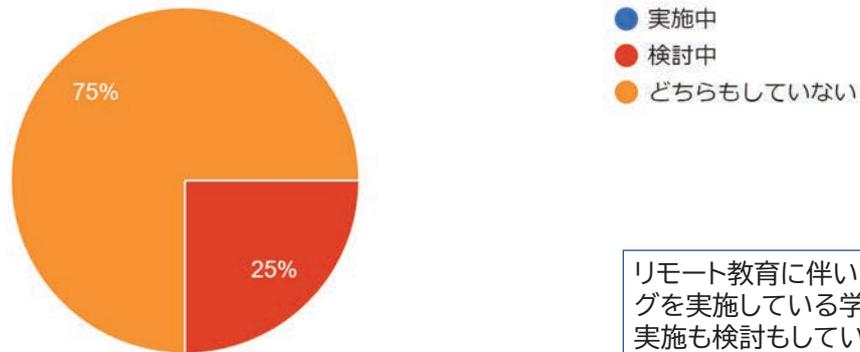
リモート教育に伴い、新たに導入した学習管理システムは、GoogleClassroomが5件と最多で、Google meetも1件あった。学習管理にはGoogle製品がよく使われていると言える

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

③ リモート教育に伴い、ペアプログラミングを実施または検討していますか？

20 件の回答



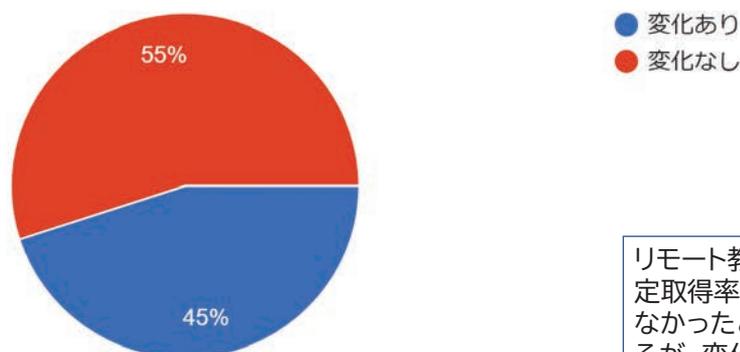
リモート教育に伴い、ペアプログラミングを実施している学校は1校もなく、実施も検討もしていない学校が75%と大半を占める。リモート教育とペアプログラミングは関連しないと考えられている

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

④ リモート教育に伴い、学習習熟度や検定取得率に変化はありましたか？また、その理由として考えられることを教えてください。

20 件の回答



リモート教育に伴い、学習習熟度や検定取得率に変化があったとする学校となかったとする学校は約半数ずつであるが、変化がなかったとする学校のほうが1校だけ多かった

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

④-2リモート教育に伴い、学習習熟度や検定取得率に変化があった理由として考えられることを教えてください。

13件の回答

- 個々の課題時間が増えて、スキルが向上した
- 国家試験取得率が低下。検定対策がリモートになりメリハリがなくなった。
- 対面と比較すると、学習の経過の把握が難しいため。
- 学生のモチベーションの維持、自己実現が難しい1年生は馴れていない為難しい
- 動画コンテンツ、スケジュール管理、得点管理などのマネジメントを徹底した
- 対面での直接指導が減った。学生のモチベーションコントロールの低下。
- 定期試験の得点状況などに大きな変化はなかった。
- 学習意欲の低い学生を十分フォローできないので低下傾向(オンラインだけの場合)
- リモートだと理解度が把握しにくいいため、対面に比べて理解度が下がる傾向がある。
- 一人一人の習熟度等をリアルタイムで把握できない。
- リモート授業の時間を最小限にしたため、はっきりとした変化が現れるほどではなかった。しかし、長引けば学力格差が顕著になったと思われます。

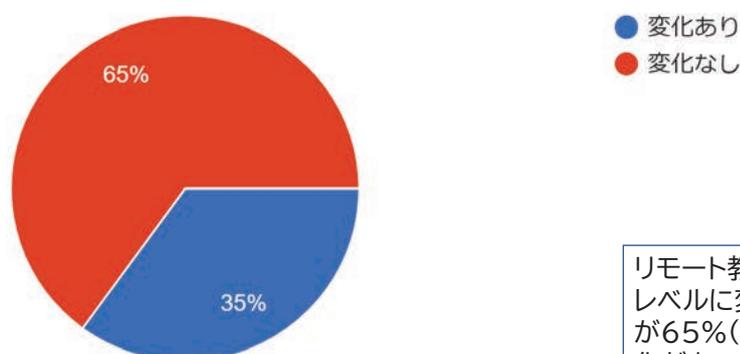
赤字:プラス効果
青字:マイナス効果

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

⑤ リモート教育に伴い、開発実践力取得レベルに変化はありましたか？

20件の回答



リモート教育に伴い、開発実践力取得レベルに変化はなかったと考える学校が65%(13校)と大半を占めるが、変化があったとする学校も35%(7校)あった

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

⑤-2リモート教育に伴い、開発実践力取得レベルに変化があった理由として考えられることを教えてください。

13件の回答

- リモートで、聞き流している学生のスキルは身につけていない
- 対面での打ち合わせが出来なくて、レベルが低下した
- プログラミング・システム開発はリモートでもそこまで習熟度に大きな違いは感じない。
- 学生のモチベーションの維持、自己実現が難しい
- 開発実践力については現在取り組んでいるものの成果によって判断できるため、現段階では大きな変化は感じられない。
- 対面での直接指導が減った。学生のモチベーションコントロールの低下。
- ハイブリッド型が中心なので、対面授業とほとんど変わりなく教育できている
- 対面なら困った時にすぐ、近くの学生や教員に相談できるがそれが、しづらい
- リモートだと理解度が把握しにくいいため、対面に比べて理解度が下がる傾向がある。
- 一人一人の習熟度等をリアルタイムで把握できない。

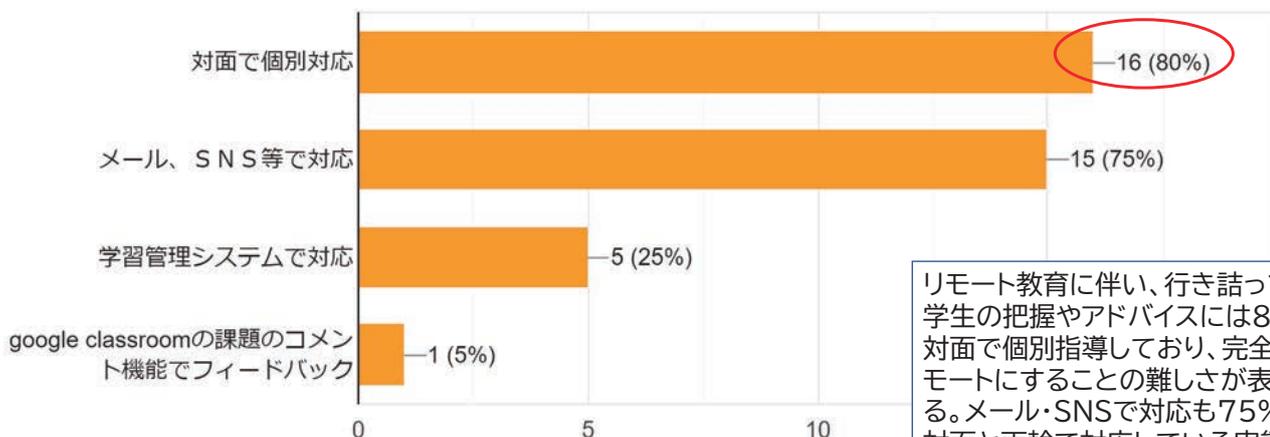
青字:マイナス効果
黒字:どちらでもない

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

⑥ リモート教育に伴い、行き詰っている学生の把握やアドバイスはどのようにしていますか？(複数選択可)

20件の回答



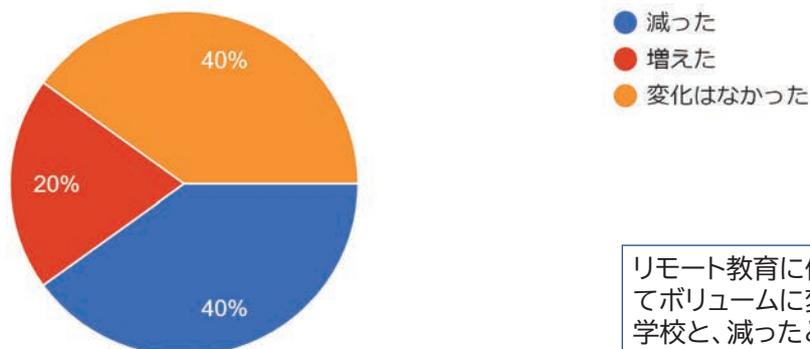
リモート教育に伴い、行き詰っている学生の把握やアドバイスには80%が対面で個別指導しており、完全にリモートにするものの難しさが表れている。メール・SNSで対応も75%と多く、対面と両輪で対応している実態が推測できる

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

⑦ リモート教育に伴い、対面教育と比べてボリュームに変化がありましたか？

20 件の回答



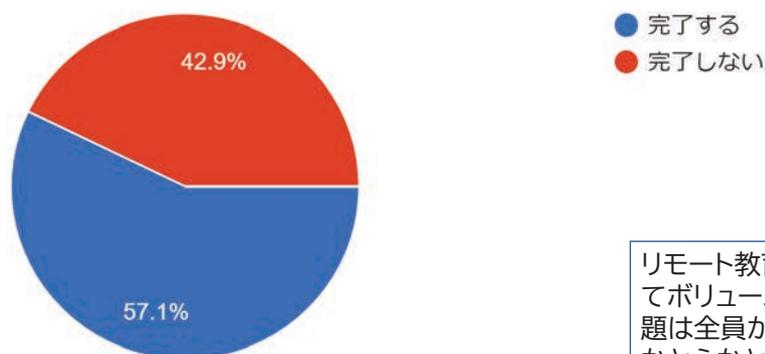
リモート教育に伴い、対面教育と比べてボリュームに変化はなかったとする学校と、減ったとする学校がそれぞれ40%(8校)ずつで、20%(4校)ではボリュームが増えたと回答している

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

⑧ リモート教育に伴い、対面教育と比べてボリュームに変化のあった場合、課題は全員が完了するレベル感の内容ですか？

14 件の回答



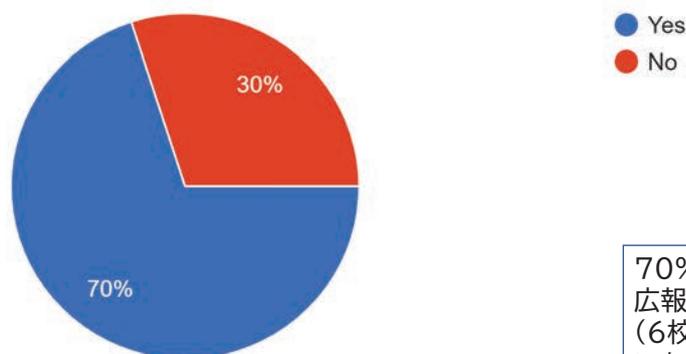
リモート教育に伴い、対面教育と比べてボリュームに変化のあった場合、課題は全員が完了するレベル感の内容かどうかという質問には、57.1%(8校)が完了すると答え、42.9%(6校)が完了しないと答えている

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

⑨ リモート教育の実施有無を広報情報として利用していますか？

20 件の回答



70%(14校)がリモート教育の実施を広報情報として利用しているが、30%(6校)では広報情報としては利用していない

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

プログラミングのリモート教育を実施「している」学校に対する質問事項に関するまとめ(1)

【リモート教育ツール】

- リモート教育に伴い、新たなツールを導入した学校は**45%**であり、**55%**は導入していない
- 導入したツールはMicrosoft Teamsが最も多く(**6件**)、次いでzoomが多い(**3件**)
- リモート教育に伴い、新たに利用することとなった学習システムはないと答えた学校が**78.9%**である
- リモート教育に伴い、新たに利用することとなった学習システムがあると答えた**21.1%**の学校では、GoogleClassroomまたはGooglemeetを導入している

【ペアプログラミングの実施検討状況】

- ペアプログラミングについては実施している学校はなく、**75%**の学校が実施も検討もしていない
- **25%**の学校でペアプログラミングを検討している

文部科学省委託事業

(2)プログラミングのリモート教育を実施「している」学校

プログラミングのリモート教育を実施「している」学校に対する質問事項に関するまとめ(2)

【リモート教育の効果】

- リモート教育に伴い、**55%**の学校では学習習熟度や検定取得率に変化がなかった
- 変化があったとする**45%**の学校では、ポジティブな変化が**2件**、ネガティブな変化が**8件**あった
- リモート教育に伴い、**65%**の学校では開発実践力取得レベルに変化がなかった
- 変化があったとする**35%**の学校では、ネガティブな変化が**7件**あり、ポジティブな変化はなかった

【リモート教育に伴う変化】

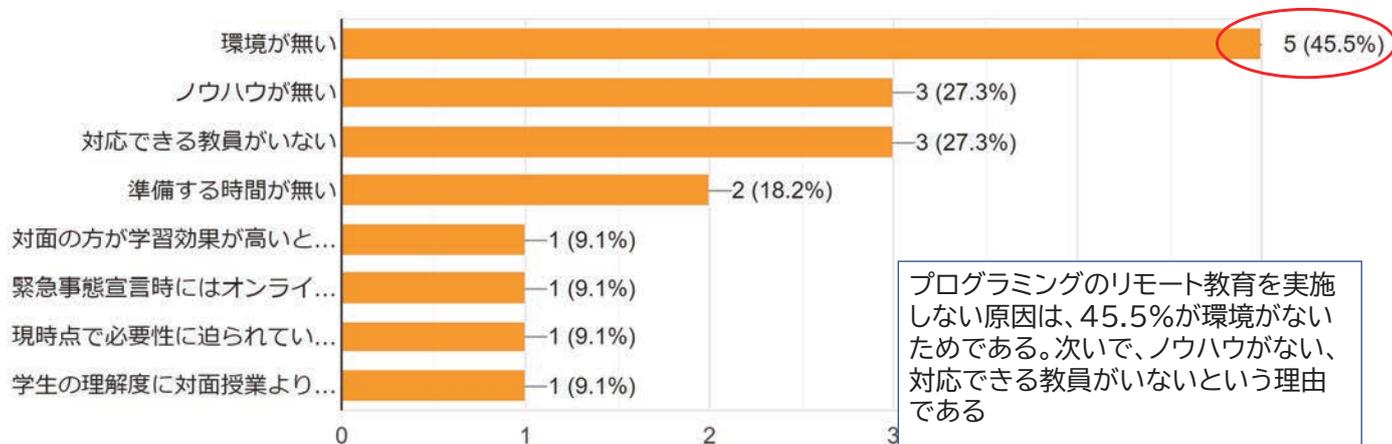
- リモート教育に伴い、行き詰まっている学生の把握やアドバイスは**80%**の学校で対面で個別対応しており、並行して**75%**の学校でSNSやメールで対応している
- リモート教育に伴い、対面教育と比べてボリュームに変化があった学校は**60%**で、**20%**が増えた、**40%**が減ったとしている
- ボリュームに変化があったとする学校では、**57.1%**で課題は全員が完了するレベル感の内容となっている
- リモート教育の実施有無を**70%**の学校では広報情報として利用している

文部科学省委託事業

(3)プログラミングのリモート教育を実施「していない」学校への質問

(3)プログラミングのリモート教育を実施「していない」学校

① プログラミングのリモート教育を実施しない原因または障害はどんなことですか？
11件の回答



文部科学省委託事業

(3)プログラミングのリモート教育を実施「していない」学校

プログラミングのリモート教育を実施「していない」学校に対する質問事項に関するまとめ(1)

【リモート教育を実施しない原因または障害】

- プログラミングのリモート教育を実施しない原因は、45.5%が環境がないためである。
- 次いで、ノウハウがない、対応できる教員がいないという理由が、どちらも27.3%である

文部科学省委託事業

総括

総括

調査目的

プログラマー育成に関する状況を統計的に調査し、学習領域、方法、評価方法等を整理する。

【調査対象校の全体像】

(遠隔学習用プラットフォームを制作するにあたってのターゲット・モデル)

共通事項(学習領域、方法、評価方法等)

- 【開発言語】:Java、Python
- 【演習方法・時間】:独自課題を設定、100時間以上
- 【演習内容】:コーディングルール・プログラム仕様書は作成していない
- 【テスト演習】:テスト仕様書は作成していない
- 【組み込みプログラムの演習】:組み込む演習はやっていない
- 【学生課題の確認方法】:ソースファイルの確認・実行ファイルの確認
- 【学生に与える課題のボリューム】:授業時間外に多少の時間の確保が必要
- 【課題フォロー体制】:学生同士のフォロー体制を構築、授業時間内外に教員がマンツーマンで指導、補習授業
- 【Web学習サービスの利用状況】:Web学習サービスは導入していない
- 【開発プロジェクト(卒業制作)について】:グループで運営、ウォーターフォール型
- 【企業連携】:システム開発等で企業と連携している

総括

【リモート教育の効果と課題】

効果	課題
<ul style="list-style-type: none"> • 個々の課題時間が増えて、スキルが向上した • 動画コンテンツ、スケジュール管理、得点管理などのマネジメントを徹底できる 	<ul style="list-style-type: none"> • 国家試験取得率が低下 • 学習の経過の把握が難しい。 • 学生のモチベーションの維持、自己実現が難しい • 学生のモチベーションコントロールの低下。 • 学習意欲の低い学生を十分フォローできない • 理解度が下がる • 習熟度等をリアルタイムで把握できない • 長引けば学力格差が顕著になった

文部科学省委託事業

総括

【リモート教育導入による開発実践力取得レベルの変化】

問題なし	問題あり
<ul style="list-style-type: none"> • プログラミング・システム開発はリモートでもそこまで習熟度に大きな違いは感じない • 開発実践力については現在取り組んでいるものの成果によって判断できるため、現段階では大きな変化は感じられない • ハイブリッド型が中心なので、対面授業とほとんど変わりなく教育できている 	<ul style="list-style-type: none"> • リモートで、聞き流している学生のスキルは身につけていない • 対面での打ち合わせが出来なくて、レベルが低下した • 学生のモチベーションの維持、自己実現が難しい • 対面での直接指導が減った。学生のモチベーションコントロールの低下。 • 対面なら困った時にすぐ、近くの学生や教員に相談できるがそれが、しづらい • リモートだと理解度が把握しにくいいため、対面に比べて理解度が下がる傾向がある。 • 一人一人の習熟度等をリアルタイムで把握できない。

文部科学省委託事業

以上の課題・問題点を解決し、効果を最大化する遠隔教育プラットフォームの開発が、今後の専門学校IT教育における重要テーマであることを確認できた。

プログラマ育成に関する実態調査
(学校調査編)

プログラマ育成に関する実態調査(学校調査編)

調査期間 令和3年11月12日～令和3年12月16日

1. 麻生情報ビジネス専門学校
2. 中央情報大学校
3. 穴吹コンピュータカレッジ
4. 日本電子専門学校
5. 高知情報ビジネス&フード専門学校
6. 富山情報ビジネス専門学校

麻生情報ビジネス専門学校

1. ヒアリング日時

令和3年11月12日(金)16:30 ~17:15

2. 訪問者

富山情報ビジネス専門学校 教務部 学科長 山田 太

高知情報ビジネス&フード専門学校 副校長 中川 隆

専門学校穴吹コンピュータカレッジ 副校長 産学連携センター 戸倉 潤也

3. 対応者

麻生情報ビジネス専門学校 福岡校 教務部 システム系 村上 香代 (Web系学科教員)

4. ヒアリング場所

麻生情報ビジネス専門学校 福岡校



5. 学校情報

(1) 基本情報

学校法人名	麻生情報ビジネス専門学校 福岡校
-------	------------------

所在地	〒812-0016 福岡市博多区博多駅南 2-12-32
Web サイト	https://asojuku.ac.jp/abcc/
設立	1986年(昭和61年)4月
募集定員数	360名
教育理念	1, 変化する情報化社会において、体系だった幅広い知識と専門分野の高い技術を持った社会人を育成します。 2, 知識や技術を最大限に発揮するための「人間力」を高めるために、思いやりと自立心を育成します。 3, 豊かな人間性と社会で必要とされる知識・技術を備えた人材を育成し、社会の発展に寄与します。
設置学科	情報工学科、情報システム専攻科、情報システム科、ネットワーク・セキュリティ科、電子システム工学科、ゲームクリエイター専攻科、経理専攻科、経理科、ビジネスエキスパート科、情報ビジネス科、経営ビジネス科、国際ITエンジニア科、国際ビジネス科、日本語科(2年)

(2)沿革

1939年(昭和14年)	3月 麻生太賀吉、中堅技術者養成のため完全無料の研修所「麻生塾」を創立
1947年(昭和22年)	4月 麻生塾小学校を設立、12月 麻生塾を財団法人に組織変更、緒方義人、理事長に就任(2代)
1948年(昭和23年)	3月 学制改革により新制高校麻生塾工業高等学校設立
1951年(昭和26年)	3月 学校法人麻生塾に組織変更
1954年(昭和29年)	1月 麻生五郎、理事長に就任(3代)
1956年(昭和31年)	3月 採鉱科実習室竣工
1957年(昭和32年)	4月 麻生塾創立15周年記念式挙行、笠親会より寄贈の校門落成し贈呈式
1960年(昭和35年)	4月 飯塚病院正看護婦養成所設置、飯塚病院内に特殊学級設置 7月 機械科実習工場竣工
1961年(昭和36年)	12月 斯道文庫を慶応義塾大学に継承、設置
1962年(昭和37年)	2月 柳俊二、理事長に就任(4代)
1964年(昭和39年)	6月 麻生塾創立25周年記念式典挙行
1966年(昭和41年)	4月 飯塚病院付属看護婦養成所を飯塚病院付属高等看護学院に改称
1969年(昭和44年)	10月 麻生太賀吉、藍綬褒章正四位勲二等瑞宝章受章

1970年(昭和45年)	4月 工業化学科設置
1972年(昭和47年)	3月 小学校閉校式、原田富平、理事長に就任(5代)
1974年(昭和49年)	7月 麻生太郎、理事長に就任(6代)
1979年(昭和54年)	4月 飯塚病院付属高等看護学院を飯塚病院付属看護専門学校に改称、麻生塾創立40周年記念式典挙 行 12月 麻生泰、理事長に就任(7代)
1980年(昭和55年)	2月 飯塚病院付属看護専門学校創立20周年記念式典挙 行 4月 飯塚病院付属看護専門学校を学校法人 麻生塾麻生飯塚看護専門学校に組織変更
1982年(昭和57年)	3月 麻生塾工業高等学校閉校式
1986年(昭和61年)	3月 麻生飯塚看護専門学校 新校舎竣工、香椎浜男子寮竣工 4月 麻生電子ビジネス専門学校開校(1号館)
1988年(昭和63年)	2月 機械科設置福岡キャンパス2号館竣工
1989年(昭和64年・ 平成元年)	4月 麻生飯塚看護専門学校を麻生医療福祉専門学校に改称 麻生医療福祉専門学校 看護棟竣工、麻生教育サービス株式 会社設立
1991年(平成3年)	4月 専修学校麻生外語観光カレッジ開校(3号館)
1994年(平成6年)	3月 香椎浜女子寮(セピアハイツ)竣工
1996年(平成8年)	4月 麻生法律専門学校麻生医療福祉専門学校北九州校 麻生テクニカルデザインカレッジ、戸畑に開校
1997年(平成9年)	4月 麻生医療福祉専門学校福岡校開校(4号館)
1998年(平成10年)	4月 麻生工科専門学校開校(2号館)
2000年(平成12年)	3月 福岡キャンパス5号館竣工、麻生外語観光カレッジ移転
2001年(平成13年)	4月 麻生リハビリテーション専門学校開校(1号館) 麻生テクニカルデザインカレッジを麻生電子ビジネス専門学校 北九州校に改称
2002年(平成14年)	3月 福岡キャンパス6号館竣工 4月 麻生法律専門学校を麻生公務員専門学校に改称 専修学校麻生ビューティーカレッジ開校
2003年(平成15年)	4月 麻生電子ビジネス専門学校を福岡校、北九州校ともに麻生情 報ビジネス専門学校に改称
2004年(平成16年)	4月 北九州キャンパス小倉校舎竣工 麻生情報ビジネス専門学校北九州校および麻生公務員専門学 校移転、麻生公務員専門学校を麻生公務員専門学校北九州校

	に改称 9月 福岡キャンパス 7号館 竣工、麻生工科専門学校 移転
2005年(平成17年)	4月 麻生公務員専門学校福岡校開校(2号館) 麻生工科専門学校を麻生工科デザイン専門学校に改称
2006年(平成18年)	2月 麻生リハビリテーション専門学校 2号館竣工
2008年(平成20年)	4月 専門学校麻生工科自動車大学校開校、あゆみらい保育園開園 8月 麻生外語観光カレッジの新実習棟「ASO SKY CUBE」竣工(8号館)
2009年(平成21年)	4月 ASO インターナショナルハウス(留学生寮)竣工 専修学校麻生外語観光カレッジを麻生外語観光&製菓専門学校に改称、麻生医療福祉専門学校を麻生看護医療専門学校に改称、麻生工科デザイン専門学校を麻生建築&デザイン専門学校に改称、麻生医療福祉専門学校北九州校を専門学校麻生医療福祉&観光カレッジに改称
2010年(平成22年)	3月 麻生公務員専門学校福岡校新校舎移転(9号館)4月 瀋陽麻生外語培训学校開校
2012年(平成24年)	4月 麻生リハビリテーション専門学校を専門学校麻生リハビリテーション大学校に改称、麻生看護医療福祉専門学校を専門学校麻生看護大学校に改称 9月 北九州キャンパス西小倉校舎竣工、麻生公務員専門学校北九州校移転麻生電子ビジネス専門学校開校
2013年(平成25年)	8月 はぐみらい保育園開園
2014年(平成26年)	4月 麻生健、理事長に就任(8代) 8月 BINUS-ASO School of Engineering 開設(インドネシア)
2015年(平成27年)	3月 学校法人麻生塾 Headquarters 竣工(10号館)総合図書館開設 4月 株式会社麻生キャリアサポート設立
2016年(平成28年)	9月 麻生専門学校グループ(11号館)竣工
2018年(平成30年)	4月 ASO ポップカルチャー専門学校 開校
2019年(平成31年)	4月 ASO 高等部 開校
2020年(令和2年)	4月 麻生美容専門学校 福岡校 福岡キャンパス移転

6. インタビュー要約

同校には、ネットワーク系・Web系・AI/IoT系の学科があり、村上先生はWeb系2年制

学科の2年生の担任でPHPの授業を担当。現在は2年生後期のECサイト作成演習を指導中。



【習得するプログラム言語】

- JavaScript(1年前期/後期、Vue.js を活用した Web フロント制作等)
- Java(1年後期/2年前期/2年後期)
- CSS(1年前期/後期)
- HTML(1年前期/後期)
- PHP(2 年前期:データベース連携/2年後期:チーム開発でPHP連携の演習)
- HTML/CSS/JavaScript(1 年前期～)
- JavaScript フレームワーク(2 年後期:Vue.js を活用した Web フロント制作等)
- PHP(2 年前期/2 年後期 :データベース連携)
- Java(1年後期/2年前期/2年後期)⇒3 年前期 JSP サーブレット
- システム開発演習(チームで EC サイト開発、PHP)

【開発教育について】

- 基本情報技術者試験については1 年前期で修了し、1 年後期からは開発演習中心となる
- 国家試験対策は、1 年次は必須の対策授業、2 年次からは選択授業を行う
- 2 年生からはチームでデザインやコードを分担して初めての演習を行う。2 年後期に初めてのチーム演習(設計～実装、プレゼン)を行い、企業連携授業で 3 回のレビュー(外部設

計レビュー、DB 設計レビュー、プレゼンテーション)を実施、企業担当者に評価してもらう

- 現在の 2 年生は 23 名で、4 チームに分けてリーダーを選出し、リーダーとドラフト会議を実施している
- 全体では 6 クラス 180 名。チーム編成はリーダーによるドラフト形式で行った(今年度は 2 年制 2 クラス、3 年制 3 クラス、4 年制 1 クラスの編成)
- プログラムを書けるのは 1 チーム 6 人中 2~3 人
- 学科は 2 年制・3 年制・4 年制があり、4 年制学科は 4 回の開発経験を積む

【遠隔授業について】

- 現在のところ遠隔授業週 3 回・リアル授業週 2 回の割合である
- プログラムができない学生の理由は、授業への参加姿勢による
- コロナの影響により 1 年次より遠隔授業だったため、下位層(取り組まない層)が多い
- 1 年前期でアルゴリズムの基本を理解しないまま進むと、それ以降の問題が全く解けなくなる
- 遠隔授業では、最大 2 クラス合同で 80 名の授業を 1 名の教員が受け持っていた
- 遠隔授業では、最初と最後に出欠確認をするが、途中応答のない学生もいる
- 遠隔授業では、Microsoft teams を使っている
- 遠隔での演習授業において、連絡のつかなくなる学生もいる
- オンラインとリアルのハイブリッド授業も展開している

【遠隔授業における課題解決について】

- 習熟別に指導ができるとうい(下位層の救い上げにパワーを割いてしまうので、上位層の意欲の高い学生に対応できていない)
- 講義中は上位層の学生の期待にこたえられていないため、時間外に対応している
- 上位層・下位層のそれぞれに対応できる授業・カリキュラムができればよい
- 上位層の学生には成功体験を積み重ねて能力を伸ばしたい
- 下位層は諦めている学生が多いが、十分にパワーを注いでいない。下位層には成功体験を積んで自信をつけて欲しい。下位層の対応時間が多く、上位層に十分にパワーを注いでいないように感じている(上位層には、応用課題、追加課題で対応している)
- クラウド上の教育プラットフォームにレベル別の課題を用意しておき、学生が各自のペースで習得できる遠隔教育システムがあれば、プログラミングの授業の課題解決になる(講義を 90 分遠隔授業で行うのは教員側も学生側も大変なので、演習を遠隔授業でやる)

【下位層の課題】

- 迷惑をかけたくないという思いがあり、自分から質問できない
- 自分から何かのアクションを起こすことにハードルがある

- 手を上げる勇気がない
- 教育プラットフォームに FAQ を用意しておくことで、自分で解決できるので自信につなげることができる。課題演習の時は Web 検索を許可しているが的外れな検索をしている学生がいる。FAQ を用意しておけばピンポイントで調べることができる

【チームプログラミングについて】

- チーム編成(メンバーの選定)が最も難しい
- リーダーは嫌だという学生もいる
- 人間関係でも揉めるケースもある
- レベル別のチームにした際に、上位層のチームがお互いに自己主張が強く崩壊するケースがあり、逆に下位層のチームからおもしろい企画が出て、アプリコンテストで入賞するなどの成果につながることもある

【その他授業に関して】

- プログラミングレベルの評価→各授業で評価基準はあるが、各評価項目の裁量は科目担当者による
- 授業におけるコーディングの規則→科目によるが、現 2 年生 PHP ではない
- プログラミングテストの授業→3 年次にテスト技法の授業がある。2 年後期次のシステム開発演習でも一部導入がある
- 企業連携→2 年後期で企業連携の授業があり、設計・コーディング開始段階で企業の人に入ってもらう
- 2 年後期システム開発演習での方針→「動くものをしっかりと作る」ことに重きを置いている

【参考:高知情報ビジネス専門学校での事例】

- スパイラルモデルでの課題演習を取り入れており、基本14ステップを習得する仕組みになっている
- ステップ1が修了すると、ステップ2はステップ1の課題にプログラムを追加していく課題内容になっている
- 上位層はステップ14の上にさらに課題が用意されており、終わらない仕組みになっている
- 下位層は最低でもステップ 7 まで進むカリキュラムになっており、ステップ 7 は C 言語で名簿を仕様書通りにリスト処理するプログラム作成ができるレベルとなっている
- 2年制課程にチーム開発の授業は取り入れていない

中央情報大学校

1. ヒアリング日時

令和3年11月16日(火) 10:30 ~11:30

2. 訪問者

富山情報ビジネス専門学校 教務部 学科長 山田 太

3. 対応者

専門学校中央情報大学校 事業開発室 プロジェクトリーダー 小澤 慎太郎

専門学校中央情報大学校 教務部 情報教育課 課長兼学年課長 内池 雄

4. ヒアリング場所

専門学校中央情報大学校

5. 学校情報

(1) 基本情報

学校法人名	専門学校中央情報大学校
所在地	〒370-0841 群馬県高崎市栄町 13-2
Web サイト	https://www.chuo.ac.jp/cid/department/ict-design
設立	1942年(昭和17年)
募集定員数	285名
教育目標	高度な専門性: 目指す職業で活躍するために高度な専門知識・技術・資格を身につけている 豊かな人間性: 自ら考え、工夫し、成長しようとする力や、何事にも挑戦しようとする心 本学生が目指すもの ● やって、みて、考える。失敗から学ぶ心と方法を知る 一生の間を通じて、仕事を学ぶ“心”と“方法”が分かる人間 ● スペシャリストとして不可欠な知識と技能を身につける 未来を担う人材として必要な実務知識と能力要件を備え、それを資格で証明できる人間 ● 豊かな人間性を兼ね備える 生き生きと職場で活躍し育っていくために必要な人間性を兼ね備えた人間

設置学科	高度 ICT デザイン学科、情報システム学科、クリエイティブデザイン学科、ポップカルチャー学科、ネット動画クリエイター学科、電気通信学科、国際エンジニア学科、日本語学科
------	--

(2)沿革

1942年(昭和17年)	有坂作太郎が「前橋服装女学院」を創立し、初代校長に就任
1952年(昭和27年)	学校の目的を商業人の育成とし「前橋商業学校」に校名変更
1974年(昭和49年)	第二代学校長に中島芳子が就任
1976年(昭和51年)	創立35年を記念して経理本科(昼・夜間部各1年制)・経理速成科(昼・夜間部6ヶ月制)を設置し、総合経理専門校として歩みだす
1988年(昭和63年)	「中央情報経理専門学校」に改名
1993年(平成5年)	有坂中央学園理事長に中島利郎が就任
1998年(平成10年)	「中央工学院専門学校」「中央会計ビジネス専門学校」を開校
1998年(平成10年)	「中央会計ビジネス専門学校」を「中央情報経理専門学校太田校」に校名変更 「中央高等専門学院」をクラーク記念国際高等学校と提携し開校する
2000年(平成12年)	学校法人関東早稲田育英会(現 学校法人中央総合学園)の運営を当学園グループが継承
2001年(平成13年)	「高崎ビューティモード専門学校」を開校 高等教育機関国内で初めて、ISO9001 認証取得
2003年(平成15年)	「中央情報経理専門学校太田校」を「中央医療歯科専門学校」に改組
2004年(平成16年)	「群馬法律専門学校」を「群馬法科ビジネス専門学校」に改組
2005年(平成17年)	「高崎ペットワールド専門学校」を開校 「中央情報経理専門学校高崎校」(旧 早稲田カレッジオブビジネス)に校名変更
2006年(平成18年)	「中央工学院専門学校」を「中央工科デザイン専門学校」に校名変更
2009年(平成21年)	学校法人設楽学園 桐生ビジネス専門学校の運営を当学園グループが継承
2010年(平成22年)	「群馬法科ビジネス専門学校桐生校」(旧 桐生ビジネス専門学校)に校名変更
2011年(平成23年)	「中央農業グリーン専門学校」を開校

	「群馬法科ビジネス専門学校桐生校」を開校 高等教育機関国内で初めて、ISO29990 認証取得
2012年(平成24年)	9月学園創立70周年を迎える
2014年(平成26年)	中央高等専門学院を中央高等専修学校前橋校と中央高等専修 学校桐生校に改編し、開校 文部科学省職業実践専門過程認定
2015年(平成27年)	厚生労働省専門実践教育訓練講座指定 「中央農業グリーン専門学校 食農調理学科」6次産業化スキル を身につける調理師養成施設として指定を受ける
2018年(平成29年)	中央医療歯科専門学校高崎校 厚生労働大臣指定歯科衛生士養 成所として指定を受ける 前橋市が公募した「旧嶺小学校活用事業」に提案し認められる グローバル人材育成を目指す英語体験施設「イングリッシュビレ ッジ MAEBASHI」が開村
2018年(平成30年)	「中央情報経理専門学校高崎校」を「中央情報大学校」に校名変 更 「中央農業グリーン専門学校」を「中央農業大学校」に校名変更
2019年(令和元年)	「中央動物看護専門学校」が「群馬サファリ・ワールド」と職業教 育連携
2020年(令和2年)	「前橋東洋医学専門学校(新校名:中央スポーツ医療専門学校)」 の運営を当学園グループが継承



6. インタビュー要約



【習得するプログラム言語】

- Java } ... ①企業の基幹業務で使われているため
- C 言語 } ... ②他の言語に行きやすいため
- Python・・・流行もあり広報面も意識して取り入れている
- アセンブラ

【言語選択の理由】

- ウェイトが最も高いのは Java(必須科目)で、C 言語と Python は選択制
- Java は教えやすい
- Java は企業からの評価される
- ベンダー系資格「オラクル JAVA」取得は履歴書に書ける
- オラクル Java ブロンズは目標 100%、合格率は 80～90%、シルバーは受験者の 50%

【資格と就職の関係性】

- 資格に関する企業の評価は、国家資格を評価する企業、ベンダー系資格を評価する企業、資格はいらぬという企業など多様である
- 都内でも Python 系の資格が必要だという企業は少ない。仕事としてはまだ活用されている場面が少ない
- Python は主に AI 系の研究室、RPA 開発、スクレイピング系で使われている
- Excel を動かすのを VBA でやるくらいなら Python でやる。Python は特徴がなく、ライトなユーザーがちょっと学べば使える
- Excel の RPA には使いやすい。ライブラリもたくさんある。ただし Java や C 言語に比べて処理スピードが遅い

【プログラム教育について】

- 高校教育(日商・全商)には Python が入っている。一般的なプログラミング教育には導入しやすいが、本格的なシステム開発は Java の方が向いている
- プログラミングを勉強することを目的にすると Python は向いていない。解決したいことがある場合は Python が有効。Java は基幹業務系の開発に向いている
- 誰もがプログラミングを習得する時代が来る
- 言語を習得する場合は、文法や知識・用法を先に覚えるよりどんどん作らせた方がいい
- 1から作らせるのではなく、機能を追加する形で習得するといふ
- 課題が見えている場合 Python が選ばれやすい。Java ではできないけれど Python ならできることもある。Excel を開くのも1行でできる。

【カリキュラムについて(2年制学科・学生数 150~200 名)】

- 1年前期:国家資格対策(情報処理技術者試験)
- 1年後期:Java 習得 12月までにオラクル Java ブロンズ、年明けからシルバーレベル
- 2年前期:オラクル Java シルバーレベル、プログラム作成演習
- 2年後期:チームプロジェクトとして開発
- イベント(CID EXPO や学園祭)向けに動くものをつくる
- チーム編成は 3~10人以上、Java を使う開発は 7~10 チーム
- 開発テーマは、今年度は教員側で出題した(パワードスーツ駆動、ルービックキューブ攻略ロボット等体験型の課題)
- フォローする教員数は4名、技術支援は個別対応で行っている

【遠隔授業について】

- コロナ対策のため、完全オンラインで授業を行ったのは2ヶ月間
- あとはハイブリッド授業(半数が自宅から、半数が教室で受講)が現在も継続中
- オンライン授業を導入し、成績は2極化が進んだ。中間層~上位層は伸び率が上がった

が、下位層はリアルで個別対応が必要であることが明白になった

- 今後、完全リアル授業となっても録画映像の限定公開は継続する
- グループウェアは Google クラウドルームを使っている
- 演習は、基本文法を授業で押さえた後、先生が作った問題を解かせている
- インPUTは学校の授業で行い、定着のための演習は自宅学習で行う。成果の評価は作成したソースコードをスクリーンショットで提出させている。課題提出で点数がつく
- コーディング規則の制定は科目によって違う。C 言語と Java ではルールが違うので、厳密なルールは定めていないが、Web 系は文法の作法が強いので規則を重視している
- ソースコードにコメントは書かせている。基本情報技術者試験の疑似言語に役立つ。最初の演習で10行~20行のプログラムであれば、すべてにコメントを書かせ、徐々にコメントの量を減らすようにしている
- 一部の学生にはソースコードを読ませることもよくさせている。例えば3人で20分間お互いのソースコードを読ませることで、人によって思考が違うことを理解できる
- 上位の学生には、AtCoder でソースコードを読ませている
- ものづくりの前段階で、論理的思考がプログラミングで表現できる段階までは以下のオンライン学習コンテンツを使っている
 - 初級:Paiza...500問くらいの問題を早い学生は2週間で解く
 - 中級:TechFUL...数学系でコンテスト向き。答えを見せてくれない点が難点
 - 上級:AtCoder...世界中の人が使っている
- オンライン学習コンテンツでの学習後、設計書の作成からプログラム制作実習に移行する
- オンラインでは GoogleMeet、Webex、zoom などのツールを使っている。ブレイクアウトルームを使う場合zoomを使い。各ブレイクアウトルームの進捗は共有スプレッドシートに書き込んでもらい、教員はスプレッドシートをチェックしている

【企業からの評価】

- 企業ごとに特徴が異なるので、授業はベースづくりで吸収力を高めるのが教育だと考えている
- 人としての総合力やどんな仕事でも対応できる力を、プログラミング教育を通して培っていくことが目標
- 地元企業は技術よりも資格取得でどれだけ頑張ってきたかを評価している傾向にある
- 東京の企業は技術を求める傾向にある
- Paiza で採用になるケースは技術を求めているケース
- 非情報関連企業の IT 技術者採用では技術力を求められるが、IT 業界への就職においてはそこまで技術力は求められない(入社後教育)
- 製造・サービス業への就職は1割程度(IT 系学科においては 90~95%は IT 業界への就職。コロナの影響で、8:2 で県内就職となっている。以前は 4:6~2:8 で都内就職が

多かった)

【品質テストについて】

- 単体テスト、結合テストは実施している
- 卒業研修や CID EXPO 出展作品では、正しく動くかどうか確認する程度
- テスト設計まではしていない
- 負荷テストはやっていない

【演習授業における課題等】

- 課題発見のプロセスを授業に取り入れたい(半年～1年で課題発見をテーマにした授業を立ち上げたい)
- 下位層のフォローアップが出来ていない。7～8割の学生には充実感と苦勞を乗り越える経験を与えられているが、わからない学生は苦しい思いをしているだろうと思う
- 自然発生的にメンタルケアを受け持つ先生も出てきており、その先生のおかげで救われている面もある

[参考]CID EXPO 2/26(土)～27(日) Gメッセで開催予定(80周年記念行事の一環)

- 学生の発表の場であり、企業にも見ていただく
- 企業出店ブースもあり、企業 PR も OK
- 参加者主体の e スポーツ大会開催
- プログラミングコンテストは高校生から職業人(3～5年目の社会人)まで参加し、問題は学生がつくる

穴吹コンピュータカレッジ

1. ヒアリング日時

令和3年12月2日(木) 15:00~16:00

2. 訪問者

富山情報ビジネス専門学校 教務部 学科長 山田 太

専門学校中央情報大学校 事業開発室 プロジェクトリーダー 小澤 慎太郎

日本電子専門学校 エンジニア教育部長 ケータイ・アプリケーション科 科長 大川 晃一

3. 対応者

専門学校穴吹コンピュータカレッジ 教務部教務課 課長 塩田 和正

4. ヒアリング場所

専門学校穴吹コンピュータカレッジ

5. 学校情報

(1) 基本情報

学校法人名	専門学校穴吹コンピュータカレッジ
所在地	〒760-0017 香川県高松市番町 2-4-14
Web サイト	https://web.anabuki-college.net/department/computer/
設立	昭和60年4月
募集定員数	60名
教育理念	職業教育を通して地域社会に貢献する人材を養成する
設置学科	情報システム学科[3年制 男女] AIテクノロジー学科[3年制 男女] ネットワークセキュリティ学科[2年制 男女] 情報ビジネス学科[2年制 男女]

(2) 沿革

1985年(昭和60年)4月	四国情報ビジネス学院を開校(高松)
1986年(昭和61年)4月	四国情報ビジネス学院が香川県知事より専修学校として認可され、穴吹情報ビジネス専門学校と校名を変更(高松)、徳島情報ビジネス学院を開校(徳島)
1986年(昭和61年)12月	穴吹コンピュータサービス株式会社設立(高松)

1987年(昭和62年)3月	学校法人穴吹学園設立(徳島)
1987年(昭和62年)4月	徳島情報ビジネス学院が徳島県知事より専修学校として認可され、穴吹電子ビジネス専門学校と校名を変更(徳島)
1987年(昭和62年)10月	穴吹コンピュータサービス株式会社徳島営業所開設(徳島)、穴吹コンピュータ教習所開設(徳島)
1988年(昭和63年)3月	穴吹進学ゼミナール高松本校設立(高松)
1988年(昭和63年)4月	米国 HEALD INSTITUTE OF TECHNOLOGY(SAN FRANCISCO)と姉妹校提携を結ぶ
1988年(昭和63年)7月	穴吹進学ゼミナール仏生山校開設(高松)
1988年(昭和63年)12月	学校法人穴吹学園設立(福山) 穴吹進学ゼミナール徳島本校設立(徳島)
1989年(平成元年)4月	穴吹コンピュータ専門学校・穴吹国際ビジネス専門学校を開校(福山)
1989年(平成元年)10月	米国エバンスビル大学(インディアナ州)と姉妹校提携及び留学提携を結ぶ
1991年(平成3年)1月	豪州グリフィス大学(クイーンズランド州)と姉妹校提携及び留学提携を結ぶ
1991年(平成3年)4月	学校法人穴吹学園設立(高松) 穴吹デザイン専門学校を開校(広島)
1991年(平成3年)10月	穴吹学園日本語学校が財団法人日本語教育振興協会から認定を受ける(高松)
1991年(平成3年)11月	韓国学校法人東西学園慶南専門大学・東西工科大学と姉妹校提携を結ぶ(韓国)
1992年(平成4年)4月	穴吹情報ビジネス専門学校を、穴吹情報専門学校と校名を変更(高松)、専門学校穴吹カレッジオブビジネス・専門学校穴吹カレッジオブデザイン・穴吹学園日本語学校を開校(高松)
1993年(平成5年)2月	学校法人穴吹学園台北事務所開設(台湾)
1993年(平成5年)4月	穴吹電子ビジネス専門学校を、専門学校穴吹カレッジと校名を変更(徳島)
1995年(平成7年)3月	英国ミドルゼックス大学(ロンドン近郊)と姉妹校提携及び留学提携を結ぶ
1996年(平成8年)4月	台湾忠信学校(新竹市)と教育連携校提携を結ぶ
1996年(平成8年)8月	豪州ボンド大学(クイーンズランド州)と姉妹校提携及び留

	学提携を結ぶ 社会人対象穴吹カレッジキャリアアップスクールを開講 (高松・徳島・福山・広島)
1998年(平成10年)1月	株式会社穴吹職業紹介センター設立(高松)
1999年(平成11年)1月	穴吹テレコム株式会社設立(高松)
1999年(平成11年)2月	株式会社穴吹カレッジサービス設立(高松)、株式会社穴吹カレッジサービス徳島営業所、福山営業所開設
1999年(平成11年)3月	サポート校高松高等学院設立
1999年(平成11年)9月	TAC 高松校開校
1999年(平成11年)11月	台湾台北市私立喬治高級工商職業学校(台北市)と教育連携校提携を結ぶ 台湾私立中山高級工業商業職業学校(高雄縣)と教育連携校提携を結ぶ
1999年(平成11年)12月	台湾台北市立松山高級商業家事職業学校(台北市)と教育連携校提携を結ぶ
2000年(平成12年)4月	TAC 徳島校開校 TAC 福山校開校
2000年(平成12年)8月	株式会社穴吹総合教育企画設立(高松)
2001年(平成13年)9月	TAC 岡山校開校、UPGARAGE 高松店オープン
2001年(平成13年)11月	穴吹学園上海事務所を開設(中華人民共和国)
2002年(平成14年)3月	UPGARAGE 松山店オープン(松山)
2002年(平成14年)4月	専門学校穴吹リハビリテーションカレッジ・専門学校穴吹工科カレッジ・専門学校穴吹ビューティーカレッジを開校(高松)、穴吹情報専門学校を、専門学校穴吹コンピュータカレッジと校名を変更(高松) 専門学校穴吹カレッジオブビジネスを、専門学校穴吹ビジネスカレッジと校名を変更(高松) 専門学校穴吹カレッジオブデザインを、専門学校穴吹デザインカレッジと校名を変更(高松)
2002年(平成14年)8月	TAC 姫路校開校(姫路)
2002年(平成14年)10月	穴吹カレッジグループ新 CI 制定
2002年(平成14年)12月	UPGARAGE 広島中央店オープン(広島)
2003年(平成15年)4月	穴吹ビューティ専門学校を開校(福山)
2003年(平成15年)6月	穴吹進学ゼミナール屋島校開設(高松)
2004年(平成16年)5月	学校法人穴吹学園香港事務所を開設(中華人民共和国)

2004年(平成16年)9月	穴吹カレッジグループと辰巳法律研究所が法科大学院進学、司法試験対策講座で教育提携
2005年(平成17年)4月	専門学校穴吹医療福祉カレッジを開校(高松)
2006年(平成18年)4月	専門学校穴吹動物看護カレッジを開校(高松) 穴吹調理師専門学校を開校(福山) 高松高等学院が、香川県教育委員会より技能教育施設として指定を受ける。
2006年(平成18年)6月	台湾台北縣立鶯歌高級工商職業学校(台北縣)と教育連携校提携を結ぶ
2006年(平成18年)10月	ロシア カムチャツカ国立大学(カムチャツカ)と姉妹校提携を結ぶ
2006年(平成18年)10月	フランストゥーレーヌエステティック学院(トゥール)と姉妹校提携を結ぶ
2007年(平成19年)4月	専門学校穴吹医療カレッジを開校(高松)
2007年(平成19年)10月	穴吹デザインプラス開設(高松)
2008年(平成20年)4月	専門学校穴吹医療福祉カレッジを、専門学校穴吹パティシエ福祉カレッジと校名を変更(高松) 専門学校穴吹福祉医療カレッジ・専門学校穴吹情報公務員カレッジを開校(徳島) 専門学校穴吹カレッジを、専門学校穴吹デザインビューティカレッジと校名を変更(徳島) 穴吹動物専門学校を開校(福山) 穴吹コンピュータ専門学校を、穴吹情報デザイン専門学校と校名を変更(福山) 穴吹国際ビジネス専門学校を、穴吹医療福祉専門学校と校名を変更(福山) 穴吹調理師専門学校を、穴吹調理製菓専門学校と校名を変更(福山) 東進衛星予備校高松常磐町校開設(高松) 東進衛星予備校徳島通町校開設(徳島)
2008年(平成20年)11月	韓国ソウル市漢陽工業高等学校と教育連携校提携を結ぶ
2009年(平成21年)10月	穴吹コンピュータサービス株式会社を、株式会社穴吹進学ゼミナールと社名を変更(高松)
2009年(平成21年)11月	UPGARAGE 岡山倉敷店オープン(岡山)
2010年(平成22年)10月	ベトナム フエ外国語大学(フエ市)と姉妹校提携を結ぶ
2010年(平成22年)11月	東進衛星予備校高松レインボー校開設(高松)
2011年(平成23年)2月	中国 江西科技師範学院(南昌市)と姉妹校提携を結ぶ

2011年(平成23年)9月	ベトナム ダナン外国語大学(ダナン市)と姉妹校提携を結ぶ
2012年(平成24年)4月	専門学校穴吹医療カレッジを、穴吹医療大学校と校名を変更(高松) UPGARAGE 広島五日市店オープン(広島) ベトナム フォンドン大学(ハノイ市)と姉妹校提携を結ぶ
2014年(平成26年)3月	穴吹学園(18校 55学科)が実践的な職業教育に組織的に取り組む学科(「職業実践専門課程」として文部科学大臣より認定される(文部科学省告示第59号))
2014年(平成26年)8月	ベトナム ホンバン大学(ホーチミン市)と姉妹校提携を結ぶ
2015年(平成27年)2月	ロシア オレンブルグ国立大学(オレンブルグ州)と姉妹校提携を結ぶ
2015年(平成27年)9月	ベトナム社会主義共和国国立ハノイ医療短期大学(ハノイ市)と姉妹校提携を結ぶ
2015年(平成27年)9月	ベトナム社会主義共和国私立ビンズオン大学(ビンズオン省)と姉妹校提携を結ぶ
2016年(平成28年)9月	カンボジア王国私立カンボジアメコン大学(プノンペン市)と姉妹校提携を結ぶ
2017年(平成29年)3月	ベトナム社会主義共和国私立タンタイ大学(ハノイ市)と姉妹校提携
2017年(平成29年)7月	あなかれほいくえん高松開園(高松)
2017年(平成29年)8月	あなかれほいくえん福山開園(福山)
2017年(平成29年)9月	あなかれほいくえん徳島開園(徳島)
2018年(平成30年)4月	専門学校穴吹デザインビューティカレッジを、専門学校穴吹調理デザインビューティカレッジと校名を変更(徳島)
2020年(令和2年)4月	専門学校穴吹調理デザインビューティカレッジ・専門学校穴吹福祉医療カレッジ・専門学校穴吹情報公務員カレッジを専門学校徳島穴吹カレッジに統合する(徳島)
2021年(令和3年)4月	せとうち観光専門職短期大学開学(高松) 穴吹学園高等学校開校(高松) 穴吹動物専門学校・穴吹ビューティ専門学校を、穴吹情報デザイン専門学校・穴吹医療福祉専門学校に統廃合し、穴吹情報デザイン専門学校を穴吹ビジネス専門学校と校名を変更、穴吹医療福祉専門学校を穴吹国際みらい専門学校と校名を変更(福山)



6, ヒアリング要約



【教えているプログラム言語】

- Java…オブジェクト指向のカチツとした言語を一つ押さえておこうという意向(以前は C 言語をやっていた)、アルゴリズムを積み上げていってオブジェクト指向に持っていく。1 年生は 90 分で週3コマやっている(基本情報技術者試験は表計算で受験している)。2 年次では Java サブレットもやっている(週 2 コマ)
- Monaca…アプリ開発環境を利用している(コードも書く)
- JavaScript
- PHP…2 年次

【オンライン授業について】

- 昨年 4 月～5 月連休明けまではオンラインだったが、その後は対面授業でやっている
- 長期休み明け2週間はオンラインで授業を行うことが学校で決まっている
- 教員は学校に来て教室から配信する
- オンライン授業だと学生の状況がわからない(どこで詰まっているのかが判断できない)
- 対面授業の方がやりやすいと感じている

【授業の目標】

- 学生の層が二極化しており、できる学生は先に進むが出来ない学生は全く進まない
- 最低クリアして欲しい最プログラミングの終課題は、6割7割はクリアできるレベルに設定している
- エラー処理などはできていない(正常ルートがちゃんと動くレベル)
- 卒業前の課題でグループ開発があるが、全員が求めているレベルには到達していない
- 健康アプリを制作しようとした学生グループがあり、企業の API を利用させてもらえないか交渉したが実現せず、API のプロトタイプを学生が作れるように指導(提案)している

【学生の能力差の原因】

- 基礎学力
- 読解力
- 理系的な感覚というよりも理解力、読み解く力が不足している
- 優秀な学生は教員が求めていることを理解できるので作業を進めていけるが、理解できない学生はそこで止まってしまう
- 理解できない学生には、基本レベルの学習を繰り返すことしかないと思っている
- 意識の高い学生は企業連携やゲームアプリ制作などを進めている
- 学生のレベルは二極化しており、中間層はあまりいない
- TechFUL の大会は 1 年生～3 年生までが参加する

【就職先について】

- 3年生 19人のうち1名はIT業界以外に行きたいという学生がいたが、他の18名はIT企業を目指して動いている
- 下位層もIT企業就職を希望しており、開発は厳しくとも運用などの仕事に就く事が多い
- 技術よりも人間力・コミュニケーション能力で評価されている実態がある

【リモート授業の課題について】

- リアルとオンラインのハイブリッドで授業をやった際、目の前のリアルな学生が置き去りになってしまうという課題が出てきた
- オンラインのプログラミングの授業で、学生は教員の画面共有と開発環境の画面を半々にするかタスクバーで切り替えながらすることになるが、入学してすぐの学生は、その操作感にも慣れていないので切り替えが難しい

【授業内容】

- コーディングのステップ数はあまり意識していない
- コーディングにはネイティブなJava、PHPを使っている
- 基本は1から作っていく授業をやっている(他者のコードを読み取って改変することが開発企業からは求められている)
- 企業からのヒアリング内容で、データベースを先にやってプログラミングという流れが理解しやすいという話に納得している(現在は並行して進めている)
- オブジェクト指向でクラス設計から入った方が無駄なくきれいなのが出来るが、卒業制作においてもそこまでは指示できていない
- それよりは、アルゴリズムやプログラムを組む力に時間をかけたい(クラス設計などは設計者が考えることなので、プログラマとしては組む力のほうが求められている)

【プログラム教育について】

- フレームワークを使っただけの授業はやっていない
- 授業ではフルスクラッチで開発する経験を繰り返して経験値を上げる(実務で繰り返しやっているとパターンが見えてくることもある)
- Javaの授業も満遍なくやる教員もいるし、必要なところだけをやる教員もいる

【テスト設計について】

- テストまでやりたいがやれていない
- 卒業制作も含め、現場では形にすることで手一杯な状態
- 卒業制作のグループ開発では、トップクラスのグループはカチッと設計もしてやれているが、それ以外のグループではそこまでできていない

【コーディングの基礎や規約はあるか？】

- コーディング基準は決めていない。紹介して勧めているコーディング規約もあるが、ルール化まではできていない
- 他人のコードを読むことでわかりにくさを実感するとコーディング規約の重要性がわかる
- 下位層に限ってその点(コーディング規約)を雑に扱うので(手を抜くので)、余計わかりにくく複雑なプログラムになる
- 学生に対していろいろやらせすぎているが、実務では使わないので、科目や内容はもっと絞ってもよいと思っている

【プログラミングの演習授業で困っている点】

- 開発現場を離れて10年以上経ち、新しい技術も出てきている中で、自分自身が経験のないことを教えている。自分自身のブラッシュアップに悩んでいる
- 自分自身が経験のない新しいことを教えなければならないことに対する悩み(面白さを伝えられるか?など)がある

【学生の成果物の確認】

- 課題提出は Google クラウドの機能を使ってやっている
- プログラムについては教員のパソコンに取り込んで稼働状態を確認している
- プログラムの成果物は、提出後の動作確認が教員にとって大変である
- 教員によっては授業で学んだこと(メモ)を提出させている

日本電子専門学校

1. ヒアリング日時

令和3年12月10日(金) 13:30 ~ 14:45

2. 訪問者

富山情報ビジネス専門学校 教務部 学科長 山田 太

富山情報ビジネス専門学校 情報ビジネス学科 清水 大樹

専門学校中央情報大学校 事業開発室 プロジェクトリーダー 小澤 慎太郎

日本電子専門学校 エンジニア教育部長 ケータイ・アプリケーション科 科長 大川 晃一

3. 対応者

日本電子専門学校 情報システム開発科 科長 柳橋宏樹

4. ヒアリング場所

日本電子専門学校1号館 9階 来賓室

5. 学校情報

(1) 基本情報

学校法人名	日本電子専門学校
所在地	〒169-8522 東京都新宿区百人町 1-25-4
Web サイト	https://www.jec.ac.jp/school-outline/message/
設立	1951年
募集定員数	昼間部 1,370名 夜間部 140名
教育目標	一人ひとりに、未来への約束。 日本電子専門学校 人と仕事を学びで結ぶ、MUSTな存在へ。 私たちは約束します。あなたが想い描く未来を実現する、力になることを。 あらゆる分野で大きな変化が進むいま、思い通りに生きるために必要なのは、誰にも負けない技術力と、自らの手で未来を切りひらく熱い志です。そんな技術と心を持った人材を輩出するため、私たちは学生の夢や希望に応え、社会と結ぶ「MUSTな存在」を目指します。 一人ひとりの未来を育むことで、これからの世界を支える、なくてはならない存在になる。それが、日本電子専門学校の使命なのです。

設置学科	昼間部 コンピュータグラフィックス科、CG 映像制作科、コンピュータグラフィックス研究科、ゲーム制作研究科、ゲーム企画科、ゲーム制作科、アニメーション科、アニメーション研究科、グラフィックデザイン科、AI システム科、Web デザイン科、ケータイ・アプリケーション科、情報ビジネスライセンス科、情報処理科、情報システム開発科、高度情報処理科、ネットワークセキュリティ科、高度電気工学科、電気工学科、電気工事技術科、電子応用工学科
	夜間部 ネットワークセキュリティ科、電気工学科、電気工事士科、情報処理科

(2)沿革

1951年(昭和26年)	社団法人ラジオ技術協会設立 日本ラジオ技術学校設立
1953年(昭和28年)	日本高等テレビ技術学校に改称
1961年(昭和36年)	日本電子専門学校に改称、創立10周年
1963年(昭和38年)	学校法人電子学園として認可 電子計算機コース新設(コンピュータ教育に着手)
1966年(昭和41年)	郵政省認定校となる(第2級無線技術士:電子工学科)
1967年(昭和42年)	通産省認定校となる(電気工事科) 郵政省より無線実験局の免許がおりる、電子計算機科新設
1968年(昭和43年)	本館完成(地上10階、地下2階)
1969年(昭和44年)	情報処理科新設、情報工学部の強化のため新たに電子計算機2台増設、2号館完成
1971年(昭和46年)	通産省認定校となる(第二種電気主任技術者:電気工学科) 創立20周年
1972年(昭和47年)	郵政省認定校となる(第2級無線通信士:電子工学科)
1973年(昭和48年)	放送工学部強化のため本格的録音スタジオ完成 情報工学部に HITAC8150 と HITAC-10 を導入
1974年(昭和49年)	放送芸術科新設(クリエイター・芸術分野教育に着手) 3号館完成
1975年(昭和50年)	4号館完成
1976年(昭和51年)	専修学校制度が発足し専門学校(工業専門課程)として認可 社会人対象の技術研修「実用講座課」開設 東京外語専門学校(姉妹校)設立
1977年(昭和52年)	電子情報処理科新設、5号館完成
1978年(昭和53年)	専門学校初の本格的オンラインシステム導入

1979年(昭和54年)	音響芸術科新設 日本電気と共同開発したマイコン教育システム(MES-II)をワシントンで開催されたコンピュータ国際会議で発表、注目を集める マスコミデザイン科でCGの授業開始
1980年(昭和55年)	レコーディングスタジオ完成 テレビスタジオ完成 6号館・8号館完成
1981年(昭和56年)	アニメーションスタジオ完成 本館にコンピュータ・センター完成 教育用大型コンピュータ HITAC M-160H 導入 創立30周年イベント「学校展」開催
1982年(昭和57年)	CG・CAD・自動制御用に YHP-1000 導入 教育ソフト開発部発足 10号館・11号館完成(実験センター)
1983年(昭和58年)	コンピュータグラフィックス科新設、12号館・13号館完成 録音・ビデオ編集室設置、スタジオ24完成 YIS・CG用システム33台導入、AED-5・CG用システム増設 ドラフタ製図105台導入
1984年(昭和59年)	那須研修所「オオシマフォーラム」完成 ニューメディア・サテライト・スタジオ設置 就職センターに電子ファイルシステム導入
1985年(昭和60年)	9号館「オオシマ・メディア・センター」完成 (120インチ2画面スクリーン、300人収容ホール) 第2回CG国際シンポジウムを本学園で開催 14号館・15号館完成、教育総合研究所設立
1986年(昭和61年)	人工知能セミナー開催(人工知能理論研究の世界的権威であるマサチューセッツ工科大学(MIT)のマーヴィン・ミンスキー教授など世界の第一人者7人の講師を招聘) 17号館完成、アルゴリズム教育システム開発 カーエレクトロニクス実験室設置、21号館完成
1987年(昭和62年)	情報システム開発科・人工知能科新設、AI展(東京流通センター)に教育機関として唯一出展参加、海外からの視察団相次ぐ PALビジネス専門学校(姉妹校)設立 留学生担当セクション設置
1988年(昭和63年)	最新ラップトップ型パソコン120台導入 CGの日米作品交流会「ATRANTA-SHINJUKU CO-

	GRAPH' 88」を米国アトランタ芸術大学でフジテレビの協力により開催 マーヴィン・ミンスキー教授が再び来校し「ニューロコンピュータについて」のフォーラム開催
1989年(平成元年)	吉祥寺インテリジェントスクール完成 シニア情報処理科(現・高度情報処理科)新設 CGワークステーションセンター完成 文部省職業教育高度化開発指定校となる、19号館完成
1990年(平成2年)	電子情報処理科にてゲーム制作の授業がスタート ハンディ型パソコン PC-9801(PC ノート)導入
1991年(平成3年)	通産省(情報処理関連)来校見学 英国教育科学省視察員来校見学 日米教育委員会(フルブライト)来校見学 創立40周年
1992年(平成4年)	コンピュータミュージック科新設
1993年(平成5年)	マルチメディア'93 出展、マルチメディアスタジオ完成
1994年(平成6年)	コンピュータアミューズメント科(現・ゲーム制作科)新設
1995年(平成7年)	「専門士」の称号を付与 インターネット本格運用開始 コンピュータネットワーク科・CG映像科新設 マックワールドエキスポ'95 出展 デジタルコンテンツフェスティバル'95 出展
1996年(平成8年)	オーストラリア「ワールド・ソーラー・チャレンジ」3千km 完走(世界24位)、東京ゲームショウ初出展(以降毎年出展)
1997年(平成9年)	コンピュータデザイン科(現・グラフィックデザイン科、Webデザイン科)新設、20号館完成
1998年(平成10年)	コンピュータアミューズメント研究科(現・ゲーム制作研究科)新設、コンピュータグラフィックス研究科新設、MIDI ワールド'98 出展
1999年(平成11年)	CGアニメ科(現・アニメーション科)新設
2000年(平成12年)	翌2001年に創立50周年を迎えるイベント実施(JR大久保駅壁画プロジェクト)
2001年(平成13年)	創立50周年記念イベント「学校展」を新宿NSビルで開催
2002年(平成14年)	日立グループ「地球にいいことポスター」で特別賞受賞 コニカ「コニカQP デジタルフォトコンテスト」で特別賞受賞 第11回全国専門学校ロボット競技会で優秀ハードウェア賞受賞

2003年(平成15年)	新館7号館完成(地上10階、地下2階)
2004年(平成16年)	ネットワークセキュリティ科新設 公開講座「ITセミナー」「クリエイターセミナー」、高等学校教員対象IT講習会開催(以降毎年開催)
2005年(平成17年)	アニメーション研究科新設 新館4号館完成
2006年(平成18年)	高度コンピュータグラフィックス科新設(現:コンピュータグラフィックス研究科) ネットワークセキュリティ科(夜間部)新設 世界最高峰のCGコンペティション「SIGGRAPH 2006」アート部門入選作品の制作に参加
2007年(平成19年)	ゲーム企画科新設 新たなチャレンジと決意が込められたシンボルマークを発表
2008年(平成20年)	第三者評価制度で教育環境、教育成果、学生支援など47項目すべての基準をクリア 第46回技能五輪全国大会で、銀・銅メダル獲得 第17回全国専門学校ロボット競技会 二足歩行ロボット競技部門で優勝 ハイジア巨大クリスマスバナーコンテストでグランプリ受賞
2009年(平成21年)	第5回公共広告CM学生賞で優秀賞受賞 若年者ものづくり競技大会で、金・銀メダル・敢闘賞を獲得 第47回技能五輪全国大会で、金・銀・銅メダル・敢闘賞独占 「東京ゲームショウ」の日本電子ブースに、過去最高1万人超の来場 世界のCG関係者が集う「SIGGRAPH」「SIGGRAPH ASIA2009」に、学生作品を出展 ハイジア巨大クリスマスバナーコンテストで、優秀賞・ハイジア賞・社長賞受賞 第3回福岡ゲームコンテストで、福岡市長賞受賞
2010年(平成22年)	ケータイ・アプリケーション科新設 第48回技能五輪全国大会で、金・銅メダル、敢闘賞を獲得 ソフトバンクモバイル認定 「ソフトバンクソリューションプロバイダー(SSP)」として学校法人初の授業を展開
2011年(平成23年)	新本館完成(地上9階、地下2階) 第49回技能五輪全国大会で、金・銀・銅メダル独占

	<p>第 41 回技能五輪国際大会で敢闘賞獲得</p> <p>第 41 回技能五輪国際大会</p> <p>日独交流 150 周年「ドイツ・日本の学生による ふろしきデザインコンテスト」最優秀賞受賞</p> <p>第 5 回福岡ゲームコンテスト「ゲームポスター部門」で優秀賞受賞</p> <p>高等教育機関初の「太陽光発電システム施工技術カリキュラム」を産学連携で共同開発</p> <p>創立 60 周年</p>
2012 年(平成 24 年)	<p>情報ビジネスライセンス科新設</p> <p>新宿クリエイターズフェスタ 2012 で優秀賞を獲得</p> <p>日本語学校の教職員が選ぶ留学生に勧めたい進学先専門学校部門賞に選出</p> <p>NEC 映像ソリューションフェア 2012 に出展</p> <p>高等教育機関として日本で初めて LPI-Japan OSS-DB アカデミック認定校に決定</p> <p>東京ドームでスポーツフェスティバルを開催</p>
2013 年(平成 25 年)	<p>U-20 プログラミング・コンテスト(主催 経済産業省)「経済産業省商務情報政策局長賞」受賞</p> <p>日本留学 AWARDS2013 専門学校部門賞受賞(2 年連続)</p> <p>私立専門学校等評価研究機構「第三者評価制度」全 47 項目の基準をすべてクリア(前回評価に続き 2 回連続)</p> <p>グラフィックデザイン科の産学連携「まねっこカメラ」が App Store からリリース。教育カテゴリーの「人気作品」に選ばれる</p> <p>第 42 回技能五輪国際大会(ドイツ開催)で敢闘賞獲得</p>
2014 年(平成 26 年)	<p>文部科学大臣認定 職業実践専門課程 認定スタート</p> <p>若年者ものづくり競技大会に Web デザイン科、グラフィックデザイン科、電気工事技術科が出場</p> <p>各職種で受賞</p> <p>第2回 ICT ネットワークトラブルシューティングコンテスト(ドワンゴ杯)で「優秀賞」受賞</p> <p>NetRiders CCNA Japan 2014 Competition で優勝、日本代表となる</p> <p>第 52 回技能五輪全国大会で、金・銀・銅・敢闘賞を制覇</p> <p>国際大会への出場が決定</p> <p>日本留学 AWARD2014 専門学校部門賞受賞(3 年連続)</p>

2015年(平成27年)	<p>高度電気工学科新設</p> <p>第43回技能五輪国際大会(ブラジル)で敢闘賞獲得</p> <p>第10回若年者ものづくり競技大会</p> <p>グラフィックデザイン職種金賞/厚生労働大臣賞</p> <p>ウェブデザイン職種 銀賞・銅賞</p> <p>日本留学 AWARD2015</p> <p>専門学校部門賞受賞(4年連続)</p> <p>技能五輪全国大会銅賞受賞</p> <p>東京ゲームショウ 20年連続出展</p> <p>TOKYO DESIGN WEEK2015 出展</p>
2016年(平成28年)	<p>若年者ものづくり競技大会に Web デザイン科、グラフィックデザイン科、ネットワークセキュリティ科が出場 各職種で受賞</p> <p>第44回技能五輪国際大会(アブダビ)へグラフィックデザイン職種の日本代表として出場が決定</p> <p>東京国際プロジェクションマッピングアワードで審査員特別賞</p> <p>ACA 世界学生大学生大会に日本代表として出馬</p> <p>Device 2 Cloud コンテストで準優勝</p> <p>日本留学 AWARD2016 専門学校部門賞受賞(5年連続、殿堂入り)</p>
2017年(平成29年)	<p>第44回技能五輪国際大会(アラブ首長国連邦)に日本代表として出場(4期連続で日本代表選手を輩出)</p> <p>東京ゲームショウ 22年連続出展</p> <p>Device 2 Cloud コンテストで優勝</p> <p>NetRiders Competitions(世界大会)出場</p> <p>2018 AIシステム科新設</p>
2018年(平成30年)	AIシステム科新設



6. ヒアリング要約



【学科情報】

- 情報システム開発科ほか 160 名の学生数(2 年制で 1 学年あたり 80 名)

- 基本情報技術者試験対策の他、Web、プログラミング、データベース、オブジェクト指向などを学ぶ
- 就職の職種はプログラマ、システムエンジニアで、それを旨とする学習内容となっている
- 実習については、学校(実習室)では Windows デスクトップを使っているが、オンライン受講する学生は OS(Windows、mac)もスペック、パソコンの新旧もまちまちという状況

【習得に力を入れているコンピュータ言語】

- Java、CASL、SQL、HTML
- CASL…応用情報技術者試験対策としてやっている
- Java、C…基本情報技術者試験対策としてやっている

【コロナ後の時間割や実習授業について】

- 学科の方針・裁量に任されている部分もあるが、実習は登校、座学については zoom を使ったオンラインに切り替えた
- 対面での実習は密を避けるためにクラスの半数の人数(40人→20人)で実施、現在でも継続中
- 教室も教員も2倍必要になるが、その分オンライン授業を2クラス同時実施するなどして調整を図っている
- 実習室が2倍になるので入らなかった科目もあり、アルゴリズムのjavaが本来座学2コマ、実習3コマなのが、オンライン座学3コマ、対面実習2コマに代わっており、実習時間が足りていない
- データベース設計(SQL)も前期は対面でやったが、後期は授業が入らず(実習室が足りず)オンラインで行った

【オンライン実習の取り組み】

- SQL のデータベース設計はオンラインで Web サービスを活用して実施した(Web サービス上での演習課題実施、課題提出は Google クラウド)
- 前期は対面で行ったが、後期は教室がなかったためオンラインで行った
- オンライン講義で文法解説の後に、サンプルを動かす実習を自宅 PC でやってもらった
- 学生側の環境がまちまちなので Web サービスを活用しなければならなかった
- 実習用の Web サイトに課題提出用のページを設けている
- サンプルのコードをコピーしてオンラインの開発環境(DB Fiddle)で動かしてみる実習
- DB FiddleはWebアドレスが実行ごとに生成できるので、アドレスをチャットで送ってもらくと教員と同じ画面を共有できる(エラーの際に教員が実行確認できる)
- 人数は1クラスが限界、2クラスは質問回答が追い付かない

- 昨年度はSQL Fiddleも使っていたが動きが重くて使えないことがあった

【オンライン授業でよかった点】

- 学生が実際に動かして確認できたところ
- 座学講座に比べて手を動かせるので積極的な学生が多くなった
- フォローのしやすさ(DB Fiddle を使うことで、URL をチャットで送ってもらって教員が確認できるので、フォローしやすくなった)

【オンライン授業の今後の改善点】

- 自前の Web サービスではないのでサービス停止のリスクがある
- 課題のチェックが厳しい(チェックやフォローが課題)
- 実習課題の量は対面の 5 分の1程度になる(対面授業の中では 1 コマで 10 個くらいの課題を与えていたが、オンラインでは 1 コマに 1,2 個)
- 学生の速度差はリアル以上に顕著に出る

【オンライン授業の実施状況について】

- SQLとJava(3コマ中の1コマのみ)をオンライン実習で実施している。Webコンパイラを使用している。
- 他の情報系学科の状況
 - JavaとCASLの実習を登校とオンラインのハイブリッドで実施している。クラスを2分割してフォローが必要な学生を登校させている。サーバ環境の必要なものについては登校で対応している。
 - JavaとHTMLを実施しているが、Javaはオンライン授業の補足としてJavaのサンプルを動かす程度。学校のノートPCを貸与している。ブレイクアウトルームでサブの先生が画面共有して個別に対応している。
 - JavaとSwiftをオンラインで行っている。ブレイクアウトルームで画面共有してリモートでフォローしている。

【Google Classroom の活用】

- 週2コマの授業で Classroom を 1 ページつくる
- 実習用 Web サイト、講義資料、課題提出用 Google フォームがセットされている
- 事前準備はかなり大変
- コロナ前は全部ホワイトボードへの板書だった
- 課題の量は減らしている(5分の1から10分の1)
- 課題は早い学生は5分くらいで終わるが、遅い学生は30分くらいかかる。その差が如実に表れるようになった。早い学生にはステップアップ課題を用意している

- ステップアップ課題が足りないという問題も起きている
- Google フォームでの課題提出は、選択式なら自動採点ができるが、SQL 文などは模範解答を出して自己チェックとして

【質問について】

- 学生はチャットで教員に質問する
- 先生 1~2 名で対応している
- 講義の時は流している学生も実習課題となると質問してくれる
- 1 回の授業で 10 人くらいが質問してくる
- 5 人~10 人くらいは質問しなくても自分でできる層
- あとの学生はバックで友達同士のオンラインがつながっていて、相談しながらやっているようである

【オンライン科目の選定について】

- オンラインにするのは基本的には講義科目が中心
- メインの言語が Java 以外の他学科では Python をやっているが、Python は主要科目なので対面授業でやっている
- Java はオンラインでやりやすい
- J Doodle は DB Fiddle と同じように実行ごとに URL が生成される。
- あくまでも外部の Web サービスなので、保証があって動かせるサイトがあるとよい

【検定取得のために学習に力を入れている言語】

- 基本情報の対策では表計算
- 他の情報分野の学科では、CASL、Java、C 言語
- 資格対策の言語は、対面の座学実習での実施が多い

【ペアプログラミングについて】

- ペアプログラミングはやっていない
- 卒業制作はグループで開発になるが、チームの中で分担を決めてやっている
- 1 台のモニターで、ペアでプログラムを組むことはない
- 授業の中で少しだけ試すことは過去にはあった

【ノーコード開発】

- ノーコード開発はやっていない
- デザイン系のクリエイターがアプリを制作する際はあるかも知れないが、今のところない

【プログラミング経験なしの入学者】

- 統計を取っているわけではないが、7～8割が未経験だとみている
- 体験入学でのヒアリングで10人中1～2人、8割くらいが未経験
- 授業のスタートは初学者に合わせている
- 経験者や早く終わった人向けにはステップアップ課題で調整している
- 初学者と経験者は入学後すぐに追いつくので、差はなくなる

【今後の学科の展望】

- 論理的思考が身につけていけばどんな言語でも対応できる。全体的な教育はそういう方向に向いている
- 開発目標は日々進化している。クラウドや開発環境・実行環境などは意識している
- クラウド環境は、今はあまり使っていない。今後は取り入れていく必要あり
- 一部で、組み込みの Arduino の Web 環境やレンタルサーバを実験的に使ってみたり、PHP で作ったプログラムを Web サーバ側で動かすのを、クラウドを使ってやってみたりしている。
- Microsoft Azure は、今は使っていない
- フルでクラウドの開発環境で勉強するのはこのあとやっていきたい
- AWS は他の学科で取り組んでいる
- クラウド開発環境は外部要因が絡むことになるので実施が難しい

【スキル評価の基準など】

- スキル評価の指標などは特にない
- 将来的にはあると良いかも知れない

高知情報ビジネス&フード専門学校

1. ヒアリング日時

令和3年12月13日(月) 13:10 ~ 13:50

2. 訪問者

富山情報ビジネス専門学校 教務部 学科長 山田 太

専門学校中央情報大学校 事業開発室 プロジェクトリーダー 小澤 慎太郎

日本電子専門学校 エンジニア教育部長 ケータイ・アプリケーション科 科長 大川 晃一

3. 対応者

高知情報ビジネス&フード専門学校 副校長 中川 隆

高知情報ビジネス&フード専門学校 教務部 濱渦 昇

4. ヒアリング場所

高知情報ビジネス&フード専門学校

5. 学校情報

(1) 基本情報

学校法人名	高知情報ビジネス&フード専門学校
所在地	〒780-0056 高知県高知市北本町 1-12-6
Web サイト	https://www.ryoma.ac.jp/kcc/
設立	1986年(昭和61年)4月
募集定員数	280名
教育目標	高度な職業教育を通して、専門知識と人間性豊かな、地域・国家国際社会に貢献する人材を育成する。
設置学科	システム工学科、情報システム学科、ゲームクリエイター学科、会計ビジネス学科、公務員学科、セールスマネジメント・ブライダル学科、製菓製パン学科、調理経営学科・調理師学科

(2) 沿革

1985年(昭和60年)6月	「高知情報ビジネス専門学校」開校のための事務局を設立
1986年(昭和61年)1月	高知県知事より「高知情報ビジネス専門学校」の設立に当たり許可を受ける。設置者 佐竹茂市 校舎(1号館、2

	号館)は高知市北本町1丁目12-6
1986年(昭和61年)4月	高知情報ビジネス専門学校が開校 初代校長 寺尾好男 就任、情報システム科、OA ビジネス科、情報処理科を開設
1987年(昭和62年)2月	高知情報ビジネス専門学校第二代校長 佐竹茂市 就任
1987年(昭和62年)3月	高知情報ビジネス専門学校本館完成(高知市北本町1丁目12-6)
1988年(昭和63年)3月	龍馬学園校友会発足 通産大臣より「情報化人材育成連携機関」に高知県で唯一委嘱される
1988年(昭和63年)4月	高知情報ビジネス専門学校第三代校長 浅野忠雄 就任 システム研究科、情報経理科を新設
1989年(平成元年)3月	高知県知事より「学校法人龍馬学園」としての認可を受ける
1989年(平成元年)4月	学校法人龍馬学園設立 高知情報ビジネス専門学校の設置者を「学校法人龍馬学園」に変更 OA ビジネス科をコース制(ビジネスコース、メディカルコース)に変更 情報経理科を知寄町校舎(高知市知寄町2丁目4-13)に移転
1990年(平成2年)3月	高知情報ビジネス専門学校3号館完成(高知市北本町1丁目12-6)
1990年(平成2年)4月	高知情報ビジネス専門学校にホテル観光科を新設 OA ビジネス科に短大併修コースを設置(併修先「産能短期大学」) 情報経理科をコース制(経理コース、経営コース)に変更
1990年(平成2年)9月	「高知情報ビジネス専門学校」5周年式典(新阪急ホテル)
1991年(平成3年)3月	高知情報ビジネス専門学校4号館完成(高知市北本町1丁目5-3)
1992年(平成4年)2月	国際デザインカレッジの本館完成(高知市旭町2-22)
1992年(平成4年)2月	高知県知事より「国際デザインカレッジ」の設立にあたり認可を受ける 国際デザインカレッジが開校 初代校長 佐竹茂市 就任

	グラフィック科、インテリアデザイン科、マンガ科を開設
1993年(平成5年)4月	高知情報ビジネス専門学校第四代校長 小谷鉄夫 就任 高知情報ビジネス専門学校に税理士科、経営情報科、情報メディカル科を新設、OA ビジネス科をビジネス秘書科に名称変更
1994年(平成6年)4月	国際デザインカレッジ第二代校長 竹村義典 就任 国際デザインカレッジに建築デザイン科を新設 オープンカレッジ(龍馬学園グループ)が社会人向け講座を開設
1995年(平成7年)4月	高知情報ビジネス専門学校に医業経営学科(大学併修、4年制)を新設 ※併修先「産能短期大学」
1996年(平成8年)3月	平成8年開校予定の「医業福祉専門学校」の介護福祉学科が介護福祉養成施設として厚生大臣指定を受ける
1996年(平成8年)4月	医療福祉専門学校が開校 初代校長 大黒寿雄 就任 ※医業経営科、福祉保育学科、※情報メディカル科、介護福祉学科を開設 ※は高知情報ビジネス専門学校より移設 高知情報ビジネス専門学校にCG マルチメディア科を新設 学科の大幅な改変と名称変更を行う。 (システム研究科⇒システム工学科、情報経理科⇒会計ビジネス科、経営情報科⇒短大併修経営情報科) 国際デザインカレッジにデザイン研究科(専攻課程)を設置 国際デザインカレッジ第三代校長 松井大州就任
1997年(平成9年)1月	国際デザインカレッジ2号館完成
1997年(平成9年)4月	国際デザインカレッジにオートメカニク科を新設 高知情報ビジネス専門学校のシステム研究科をシステム工学科に変更
1998年(平成10年)4月	高知情報ビジネス専門学校第五代校長 三谷松太郎 就任 「龍馬学園オープンカレッジ」を学園(生涯学習部)に移設
1999年(平成11年)1月	医業福祉専門学校第二代校長 佐竹茂市 就任
1999年(平成11年)4月	高知情報ビジネス専門学校にスキルアップ科を新設 高知情報ビジネス専門学校 税理士科を会計スペシャリスト科に名称変更

	<p>高知情報ビジネス専門学校 情報ビジネス科(会計コース、ビジネスコース、ショップコーディネーターコース)を新設</p> <p>高知情報ビジネス専門学校 情報システム科をコース制(プログラマコース、システムコース、CG コースに変更)</p> <p>医業福祉専門学校にトータルビューティ学科を新設</p>
2001年(平成13年)4月	<p>高知情報ビジネス専門学校第六代校長 山崎清郎 就任</p> <p>医業福祉専門学校第三代校長 三谷松太郎 就任</p> <p>高知情報ビジネス専門学校の情報ビジネス科にインターネットビジネスコースを設置</p> <p>高知情報ビジネス専門学校にCG クリエーター科を新設</p>
2002年(平成14年)4月	<p>国際デザインカレッジ第四代校長 佐竹新市 就任</p> <p>医業福祉専門学校第四代校長 中坂雄一 就任</p> <p>龍馬学園オープンカレッジがプロメトリック社のCBT試験会場として認定される</p>
2003年(平成15年)3月	<p>龍馬学園オープンカレッジ「上雅ビル」(高知市北本町1丁目13-3)に移転</p>
2003年(平成15年)4月	<p>「国際デザインカレッジ」を「国際デザイン・ビューティカレッジ」に校名変更</p> <p>国際デザイン・ビューティカレッジに美容科を新設</p> <p>国際デザイン・ビューティカレッジにトータルビューティ科を医業福祉専門学校より移設</p> <p>「医業福祉専門学校」を「龍馬ふくし専門学校」に校名変更</p>
2003年(平成15年)10月	<p>国際デザイン・ビューティカレッジに美容科通信課程を新設</p>
2004年(平成16年)4月	<p>高知情報ビジネス専門学校の情報ビジネス科のコース制(会計、インターネットビジネス、ショップコーディネーター)を廃止</p> <p>高知情報ビジネス専門学校に公務員科、会計ビジネス科、インターネットビジネス科、ショップマネジメント科を新設</p>
2004年(平成16年)8月	<p>国際デザイン・ビューティカレッジの本館増築工事完了</p>
2005年(平成17年)3月	<p>高知県園芸連跡地に龍馬学園別館完成</p>
2005年(平成17年)4月	<p>国際デザイン・ビューティカレッジにマンガ科をマンガ・アニメ科(3年制)に変更</p>

2006年(平成18年)1月	龍馬ふくし専門学校が四国厚生支局より看護師養成所として指定される 龍馬ふくし専門学校と国立ウーロンゴン大学(オーストラリア)との間で協定が結ばれる(研修生の受け入れに関する提携調印式が行われる)
2006年(平成18年)1月	龍馬ふくし専門学校「看護棟」校舎完成
2006年(平成18年)3月	龍馬学園オープンカレッジ「四国文教ビル」(高知市愛宕町1丁目10-6)に移転
2006年(平成18年)4月	高知情報ビジネス専門学校 第七代校長 佐竹新市 就任 国際デザイン・ビューティカレッジ 第五代校長 包国勝 就任 「龍馬ふくし専門学校」を「龍馬看護ふくし専門学校」に校名変更 龍馬看護ふくし専門学校に看護学科(大学併修、4年制)を新設(併修先「人間総合科学大学」)
2006年(平成18年)5月	学校法人龍馬学園 第二代理事長に佐竹新市就任 学校法人龍馬学園 初代学園長に佐竹茂市就任
2006年(平成18年)6月	創立20周年 記念講演会「かるぽーと」 祝賀会(ホテル日航高知旭ロイヤル)
2007年(平成19年)4月	高知情報ビジネス専門学校 第八代校長 井上佳彦 就任
2007年(平成19年)6月	龍馬学園創立20周年記念事業 1000人献血キャンペーン 1000人達成で高知県知事より表彰される
2007年(平成19年)10月	生涯学習健康センター「キラヴィ」を設置(旧こうち社会保険センター) 生涯学習健康センター 初代センター長 中西康廣 就任
2007年(平成19年)12月	高知情報ビジネス専門学校 製菓製パン科 校舎完成
2008年(平成20年)4月	高知情報ビジネス専門学校に製菓製パン科を新設 高知情報ビジネス専門学校のインターネットビジネス科をweb・ゲーム科に変更 高知情報ビジネス専門学校のショップマネジメント科をショップ・ブライダル科に変更 国際デザイン・ビューティカレッジにプロダクトデザイン科を新設 国際デザイン・ビューティカレッジ インテリアデザイン

	科・建築インテリア科の2学科を統合、インテリア建築科に変更
2009年(平成21年)4月	高知情報ビジネス専門学校 第九代校長 中西康廣 就任 生涯学習健康センター 第二代センター長 井上佳彦 就任 国際デザイン・ビューティカレッジのデザイン研究科を総合研究科に変更
2009年(平成21年)5月	龍馬馬学園オープンカレッジが生涯学習健康センター内に移転、名称も生涯学習健康センターに改名
2010年(平成22年)4月	高知情報ビジネス専門学校のCGクリエイター科をデジタルメディア科に変更 高知情報ビジネス専門学校のweb・ゲーム科をゲームクリエイター科に変更 国際デザイン・ビューティカレッジのトータルビューティ科にコース制導入(エステコース/メイク・ネイルコース) 国際デザイン・ビューティカレッジのマンガ・アニメ科(3年制)をマンガ科(2年制)に変更
2011年(平成23年)1月	生涯学習健康センター 第三代センター長 中西康弘廣 就任
2011年(平成23年)4月	国際デザイン・ビューティカレッジのインテリア建築科を住空間デザイン科に変更しコース制を導入(建築士・インテリアコーディネーターコース/インテリア雑貨コース) 国際デザイン・ビューティカレッジのオートメカニック科・プロダクトデザイン科を統合しカーテクノロジー科に変更しコース制を導入(オートメカニックコース/デジタルエンジニアコース)
2011年(平成23年)4月	龍馬学園創立 25 周年記念 3 校合同学園祭開催(高知ちばさんセンター)
2012年(平成24年)4月	国際デザイン・ビューティカレッジ 第六代校長 近藤邦夫 就任 龍馬看護心くし専門学校 第五代校長 岡田健 就任 文部科学大臣より 3 校 16 学科に対し、職業実践専門課程に認定した。(申請該当学科の全学科が認定された。) 高知情報ビジネス&フード専門学校 文部科学大臣が「職業実践専門課程」に認定

	<p>国際デザイン・ビューティカレッジ 文部科学大臣が「職業実践専門課程」に認定</p> <p>龍馬看護ふくし専門学校 文部科学大臣が「職業実践専門課程」に認定</p>
2013年(平成25年)4月	<p>国際デザイン・ビューティカレッジの住空間デザイン科を建築・インテリアデザイン科に変更</p> <p>国際デザイン・ビューティカレッジのカーテクノロジー科を自動車整備科に変更</p> <p>龍馬看護ふくし専門学校に子ども未来学科(2年制)新設</p>
2015年(平成27年)3月	生涯学習健康センター 第四代センター長 永橋正至 就任
2015年(平成27年)7月	調理系新校舎完成(旧1号館・2号館跡)
2016年(平成28年)4月	<p>国際デザイン・ビューティカレッジ 第七代校長 永橋正至 就任</p> <p>「高知情報ビジネス専門学校」を「高知情報ビジネス&フード専門学校」に校名変更</p> <p>高知情報ビジネス&フード専門学校に調理師学科(1年制)・調理経営学科(2年制)を新設</p>
2016年(平成28年)6月	創立30周年記念祝賀会(ザ クラウンパレス新阪急 高知)
2018年(平成30年)4月	龍馬看護ふくし専門学校 第六代校長 野町裕 就任
2020年(令和2年)4月	<p>生涯学習健康センター「キラヴィ」 第五代センター長 上岡直司 就任</p> <p>生涯学習健康センター「キラヴィ」に学童保育「龍馬学園キッズアカデミー(RKA)」を開設</p> <p>高知県知事より、「高知まんがBASE」の運営を受託</p>



6. ヒアリング要約



【習得に力を入れているプログラミング言語】

- 情報システム学科(2年制)では、C 言語のみをやっている・・・基本情報技術者試験を

C 言語で受けさせているため

- Visual Studio の C++開発環境を使用して C 言語の実習を行っている
- システム工学科(3年制)では、C 言語に加えて Java、Visual Basic(以下、VB)をやっている
- オブジェクト指向については、2 年制の学科では実践はしていないが、3年制学科では Java を通して実習を行っている
- 県内企業からの要望として、以前は COBOL をやってほしいというリクエストがあり、検定対策も COBOL でやっていた。(基本情報技術者試験の科目に COBOL があった頃)

【就職先の県内外比率】

- 県外:県内=7:3~8:2

【卒業制作について】

- 設計からテストまでの一連の工程は VB でやっている
- Web 系の言語(HTML、Java script)にはほとんど手をつけておらず、VB なら UI も作ることができ、言語習得にかかるコストを設計やテストに回すことができるため
- 開発演習はウォーターフォール型で進めている
- VB は半年間(50分授業で週 8 コマ)基礎をやってから実践に入る
- 到達点はテキストに載ってるものを作るレベル
- アルゴリズムで躓くことはほぼなく、ある程度のものは作れるようになる
- 3年制システム工学科では3年前期から卒業制作がはじまる
- 卒業制作は、要件定義は教員側でやる。インクリメンタルモデルで4期に分けたうちの、2期からスタートする想定(企業で1から作ることは少ないため)
- 外部設計は 2 年次にやる。1 期終了段階での画面数は 20 画面程度
- 具体的には、ERP システムの開発で、在庫管理機能の追加、定期発注・定量発注機能の追加を行う
- データベースはオラクル使用。データベースを見に行く部分は教員側でセッティングする。ベースのプログラムはあまり知らなくてもよく、処理順序を学ぶ事がメイン

【テスト設計について】

- シナリオベースの機能試験を行っている
- 単体テスト・結合テストなどのホワイトボックステストは時間がないのでやらない

【授業でのチーム開発について】

- 開発は 8 名1チームで1つのシステムを組む(今年は 2 チーム、過去最大で 3 チーム、

教室キャパシティが 24 名なので 3 チーム以上になることはない)

- チーム編成は、国家試験の取得状況とクラス内の関係性を考慮し、担任教員に確認しながら組み合わせる
- 人間性(資格はないがリーダーシップはある、など)を考慮し、能力的なばらつきはチームの中で解決する
- チームリーダーも作業を行うという意味で、教員がプロジェクトマネジャー(兼お客さん)の立場
- チーム開発では、プログラミングがうまくいっていないケースもあるが、コミュニケーションの問題の方が大きく、できない学生をどうやって巻き込んでいくかがポイントとなる
- できない学生には周りがサポートするよう教員が促している
- レベル別のチーム編成とした場合、できない方のグループが何もできずに終わってしまう懸念がある
- 「自分に与えられたことだけをやればいい」というタイプの学生が増えてきた
- 完成しないことはなく、最終的には期間内に納まる。下級生の前で発表する機会があり、それが最終納期
- チームが崩壊するケースは、最近はないが、昔はチーム内で喧嘩などもあった
- 卒業制作科目「システム構築」は、学校だけでやるよう指導している(家には持ち帰らない)。決められた時間内(前期週 11 コマ/後期週 13 コマ)で作業を終わらせることを課している
- コミュニケーションには Slack を使っている
- スケジュール管理はレッドマインを使っている
- オンライン授業ではコミュニケーション不足をどう解消するかが課題になった
- チーム作業が個人個人になって会話ができていない状況となった
- 授業時間中の教員はレビュー対応がメインとなっている
- 教員はレビューの際に、「こうしなさい」ではなく、なるべく提案を受け入れる立場を取っている
- 早く終わりそうなチームに対しては、追加作業(楽観排他の追加、パスワード保護など)を課すこともある
- 課題のレベル感としては、企業でやってきた内容と質的には近い。責任感の面では違いがあるが、内容的には現場に近いと考えている。ただし実用品質には至っていない

【フレームワークの活用について】

- フレームワークを授業で扱うことについては、ついて来れない学生が出てきてしまう懸念がある。Java のフレームワークで Spring があるが、Java 習得で精一杯となっている

- フレームワークで簡単に作れるという考え方もあるが、基礎をやったほうが良いと考えている

【オンライン授業について】

- 完全オンライン授業は、国家試験対策だけ行った
- コロナの緊急事態宣言で学校が開けられない期間は、課題を出した

富山情報ビジネス専門学校

1. ヒアリング日時

令和3年12月16日(木) 15:00 ~ 16:00

2. 訪問者

富山情報ビジネス専門学校 情報システム学科 清水 大樹

富山情報ビジネス専門学校 情報システム学科 新田 雅道

高知情報ビジネス&フード専門学校 副校長 中川 隆

3. 対応者

富山情報ビジネス専門学校 情報システム学科 講師 高井 波留美

4. ヒアリング場所

富山情報ビジネス専門学校 C館会議室

5. 学校情報

(1) 基本情報

学校法人名	富山情報ビジネス専門学校
所在地	〒939-0341 富山県射水市三ヶ576
Web サイト	https://www.bit.urayama.ac.jp/
設立	1966年(昭和41年)4月
募集定員数	220名
建学の精神	1 質実にして明朗な人格形成 1 専門的な学問とその応用を通して社会に貢献する人格形成
設置学科	情報システム学科 建築・デザイン学科 Webクリエイター学科 ホテル・ブライダル学科 医療事務学科 診療情報管理士研究科 公務員学科 インターナショナルビジネス学科 日本語学科

(2) 沿革

1966年(昭和41年)4月	中央予備校開校
1966年(昭和41年)12月	学校法人 浦山学園認可
1968年(昭和43年)4月	中央経済専門学校開校
1970年(昭和45年)4月	中央経済専門学校を富山経済専門学校に校名変更
1976年(昭和51年)5月	富山経済専門学校商業実務専門課程設置
1988年(昭和63年)7月	富山経済専門学校文化教養専門課程設置
2002年(平成14年)4月	富山経済専門学校を富山情報ビジネス専門学校に校名変更、工業専門課程、教育・社会福祉専門課程が設置される
2012年(平成24年)3月	中国瀋陽の国際交流協会との共同による日本語教育センター(「中山日本語中心」)を設立
2013年(平成25年)4月	厚生労働省「指定保育士養成施設」に指定される
2014年(平成26年)4月	高度情報システム学科、情報システム学科、デジタルメディア学科、インターネットビジネス学科、ホテル・ブライダル学科、医療事務学科の6学科が、職業実践専門課程として文部科学大臣より認定される。
2016年(平成28年)2月	幼児教育学科が職業実践専門課程として文部科学大臣より認定される
2016年(平成28年)12月	浦山学園 50周年記念式典
2018年(平成30年)4月	浦山学園 USP 竣工



6, ヒアリング要約

- 1年次の授業は Visual Basic 週 2 コマ×15 週=30 コマ、Java 週 2 コマ×15 週=30 コマ
- 1年次の授業は言語習得とアルゴリズムを中心にやっている(アルゴリズムが弱い学生は多い)



【学習に力を入れているプログラミング言語】

- Visual Basic(以下 VB)…プログラム教育の導入として、動くものができると楽しいので、If 文や繰り返し文などのプログラミングの基礎的なことを覚えるのに VB を使っている。
- Java…1 年後期から文法的に VB の復習のような形で入っていく。しかし、VB では学習していない Java メソッドやクラスが入ってくると難しいと学生は言っている
- VB の方が視覚的にプログラムを確認でき、遊び心もあり刺激になるので学生は楽しめているので導入には向いている
- その遊び心や楽しさを考慮しないのであれば、最初から Java でもいい
- 当初は VB だけを教育していた。最終的にゲームを制作させたりしていたが、今は VB の授業は半期だけなので、時間がなくそれはできない
- 言語をたくさん覚えるよりも一本の言語を極めた方がいいという意見と、たくさんの言語に触れた方がいいという意見があるが、用途が異なるものは学習した方がいいと思う。VB と Java は文法が似ているので同じグループと考えていいだろう。Web コンテンツ制作には Java script が必要になるので、別途学習する方がいいと思う

- 以前 VB と C 言語を同時にやっていた時期もあったが、文法的に混乱する学生もいた。VB と Java は一本化してもいいとも思う。Java script は別なのであった方がいい
- 2 年制学科では Java、PHP、Java script をやっているが、PHP、Java script は時間数が足りないと担当の先生から聞いている。
- レベルの高い学生たちは学習能力が高いので、PBL の授業では Java script を活用して制作しているグループもある

【コーディングの基準について】

- 基礎的なコーディング方法(記述・文法)については厳しく指導している
- VB と Java は記述法が異なるがアルゴリズムは一緒なので特に違和感なく Java に入ることができている
- 能力の高い学生には結果が出るだけでなく、できるだけすっきりしたプログラムを書くことを心がけるよう声掛けしている

【学生の能力評価】

- 本当の能力評価は、コーディングをさせてみるしかない
- プログラムには正解はないが、サンプルコードを配布し、違いを発見させるレビューはやるようにしている
- 能力が高く課題が早く終わる学生には、段階によって追加課題を出している

【リモート授業について】

- 昨年は2か月間リモートで授業を行なった
- プログラム演習の授業は対面になってからの再開を要望したが、1科目だけリモートで行った
- リモート授業では躓きのある学生のコードを見ることができない点がボトルネックになった
- Google Meet で画面共有すればコードの確認はできるが、全員に画面が共有されてしまうと他の学生にもそのコードが見えてしまう。そこで Google Classroom を 2 つ用意し、1つは全員のルームとして画面共有せず、もう1つは質問用のルームで、質問のある学生のみ画面共有しながら個別指導をしていた
- 対面時には通常ホワイトボードを使って授業していたが、リモートのためにパワーポイントでスライドを作成したので、対面授業になった今でも授業でそれを使っている
- 授業時間外にも Google Meet を使って学生の質問に答えていた
- 対面授業では授業中に学生同士で助け合いが起こりやすいが、リモートではそれが出来ない。しかし一部の学生同士が Meet のグループを活用して助け合いをしていた

【授業の進め方】

- プログラミング経験のない人にプログラムを嫌がられないよう基礎からなるべく丁寧にやっている
- 授業そのものは中間レベル層に合わせて授業を進めている
- 最初の導入時は「VB だったら、こうすると色が変わるよ」というようにゴール(プロトタイプ)を見せている。特に單元ごとの工夫はないが、ゴールや出来上がりの状態を最初に見せるようにしている。java でも、実行結果はプリント等に表示している
- 1 年次は VB と Java と両方やり、2 年次は Web 系の言語が中心になる。実際 VB はどこまでを教えればいいのかわからない
- PBL の授業で上手くいかないことがあった時に、その理屈がわからない学生は何を調べればいいのかをわからずに調べている。また、対処の指示待ちが多いので自分たちで調べる力もつけるべきだと思う
- 1 年の java については、1 つのプログラム課題は 30~40 ステップのものが多い。
- 凝ったものでも 60~70 ステップ程度
- サンプルコードとの照合はするように指示している

【検定対策授業】

- 基本情報技術者試験の午後問題は表計算を選んでいる。表計算の授業があるので Java では検定対策はやっていない
- 以前は CASL で受験していた。現在は表計算で受験しているが、Java は言語の授業もあるので Java で受験する学生がいてもいいのではないかととも思う

【これからどんなことを教育したいのか？】

- 企業のニーズがわからないと何をやったらいいのか実際にはわからない
- 企業から派遣されている講師から情報を得るのが一番良いと思う
- 実際にはアルゴリズムの弱い学生が多い

【テスト設計について】

- 2 年生の卒業制作(PBL)ではテストまでさせている
- 設計書の段階でテストデータを作成する必要があるが、時間がないのでやっていない
- テスト指針までは示していない

プログラマ育成に関する実態調査(学校調査編)総括

【開発実践力習得・検定取得のためのプログラム言語】

- すべての学校で Java は履修しており、C 言語を履修する学校も複数あった
- ほかに Visual Basic、Java script、HTML、PHP、SQL などを履修する学校があった
- 一部で CALS、Python をやっている
- 基本情報技術者試験の選択言語は C 言語、Java、表計算が選ばれている

【リモート教育の状況】

- コロナの影響でほとんどの学校ではリモート授業を経験しているが、演習をフルリモートでやっている学校は少なく、ハイブリッドか対面で実施している
- 多くの学校で Google Classroom、Google meet が用いられ、一部で zoom、Webex が使われている
- 一部の学校では学生の使うパソコンのスペックがまちまちで課題となった
- リモート授業では下位層の学生のフォローが課題となるケースが多い
- リモート教育では、リアル授業以上に学生の速度差・レベル差が堅調に出る傾向がある
- リモート授業においては、学生のコードを見ることができない点がボトルネックになっている
- オンライン授業ツールとして DB Fiddle を活用する学校があった
- 教員側としては準備に余計時間がかかるようになった
- Paiza や TechFUL、AtCoder を活用している学校があった

【ペアプログラミングの演習】

- ペアプログラミングを授業で取り入れている学校はなかったが、グループワークやソースコードをお互いに読むなどを取り入れている学校はあった

【学生チームによる演習】

- 卒業制作などの開発演習(EC サイト、電子受発注システムなど)はほとんどの学校がチームでやっている一方、2 年制課程ではチーム開発は授業では取り入れていない学校もあった
- チームは学校によって規模が異なり、4 名～8 名の構成であった
- チーム決定は教員が行うケースが多く、一部ではチームリーダーによるドラフト制にしている学校もあった

【ノーコードでのシステム開発】

- ノーコードでの開発を行っている学校はなかった

【検定取得とプログラミング教育の割合】

- 1 年前期を国家資格対策にあて、1 年後期からプログラミングスキルを習得する学校や、1 年前期からプログラミングスキルの習得と同時進行で検定対策をする学校があった

【言語やアルゴリズム以外で教育している能力】

- 読解力や基礎学力が能力差の原因となっているとする学校があった

【コーディング基準について】

- コーディング基準はない学校が多いが、記述方法や文法については指導している
- すべてのソースコードにコメントを書かせ、徐々にコメントを減らしていく方法を採用している学校があった

【テスト指針について】

- ほとんどの学校でテストまではしっかりと取り組めていない
- 単体テストや動作確認はするが、負荷テストのようなテストや設計段階におけるテストデータの作成までは実施できていない

【その他】

- フレームワークの授業はやっていない学校が多かった
- 下位層の学生の引き上げに課題を抱える学校・教員が多かった
- 学生の二極化は全ての学校で課題であった
- 上位層には追加課題で対応している学校が多かった
- リモート授業を導入することで中間層から上位層の伸び率が上がったという報告もあった
- 教員の課題として、現場から離れる年数が増えると現場感覚から遠のくことがあげられた
- 地方企業は技術よりも人物評価や資格取得を評価する傾向が強く、東京などの都会の企業は技術をより評価する傾向があると感じている学校があった

プログラマ育成に関する実態調査
(企業調査編)

プログラマ育成に関する実態調査(企業調査編)

調査期間 令和3年11月12日～令和3年12月18日

1. 株式会社オートメイション・テクノロジー
2. システム・アルファ株式会社
3. 北電情報システムサービス株式会社
4. 株式会社北陸ソフタス
5. 株式会社トスバックシステムズ
6. 株式会社高志インテック
7. 株式会社ウチダ人材開発センタ
8. 株式会社シンクスバンク
9. 株式会社エクスチェンジ ソリューションズ
10. パシフィックソフトウェア開発株式会社
11. 株式会社 ソフテック
12. 株式会社ソフト
13. 株式会社ブレインファクトリー

株式会社オートメイション・テクノロジー

1. ヒアリング日時

令和3年11月12日(金)15:00 ~ 16:15

2. 訪問者

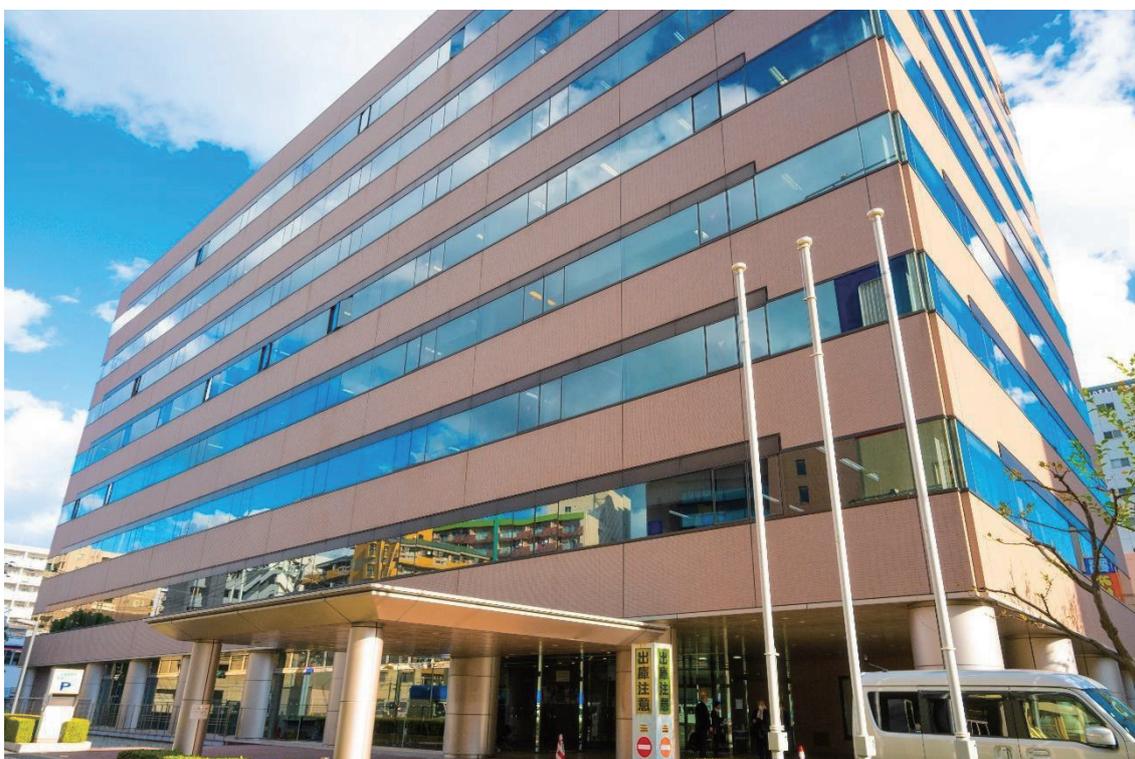
富山情報ビジネス専門学校 教務部 学科長 山田 太
麻生情報ビジネス専門学校 福岡校 校長代行 北原 聡
高知情報ビジネス&フード専門学校 副校長 中川 隆
専門学校穴吹コンピュータカレッジ 副校長 産学連携センター 戸倉 潤也

3. 対応者

株式会社オートメイション・テクノロジー 取締役管理部長 権藤 泰二
株式会社オートメイション・テクノロジー 管理部 課長 堀 純子
株式会社オートメイション・テクノロジー システムソリューション部長 右田 和寛
株式会社オートメイション・テクノロジー システムソリューション部 副部長 江田 道也

4. ヒアリング場所

株式会社オートメイション・テクノロジー



5. 企業情報

(1) 基本情報

企業名	株式会社オートメイション・テクノロジー
所在地	福岡県福岡市南区那の川1丁目24-1 九電工福岡支店ビル 5F
Web サイト	https://www.inf9.co.jp/
設立	平成3年12月18日
従業員数	160名(2021年4月現在)
企業理念	<p>人として技術を以って社会のために</p> <p>人として:企業は人なりの理念のもとに、一人ひとりの個性を生かし能力を高め、活力とうるおいのある企業風土をつくります。</p> <p>技術を以って:時代のニーズを先取りした技術の研究・開発に努め、つねに未来を考える企業としてお客さまの信頼と期待にこたえる事業を展開します。</p> <p>社会のために:広い視野に立った最先端 IT 企業として「人」・「環境」・「技術」の最適な調和をめざし豊かな人間環境の創造に力をつくします。</p>
事業内容	インフラ構築、システム開発、エネルギー監視、セキュリティマネジメント、Web サイト制作、データセンター事業

(2) 沿革

1991年(平成3年) 12月18日	株式会社九電工(出資比率60%、工場の一括受注を目標)とサンリツオートメイション株式会社(出資比率40%、九州への業務拡大)の合併会社として設立
1992年(平成4年)1月	福岡市南区那の川マンションの倉庫を仮事務所として、FA(ファクトリーオートメイション)事業開始
1992年(平成4年)3月	福岡市中央区ヘルスビジネスビルに移転
1996年(平成8年)10月	インターネットプロバイダー事業開始
1999年(平成11年)6月	旧九電工グループビルに移転
1996(平成8年)年6月	金融関連事業開始
2001年(平成13年)6月	株式会社九電工出資比率100%となる
2003年(平成15年)10月	ネットワーク事業開始
2006年(平成18年)4月	株式会社九電工情報管理部門の業務委託開始に伴い、九電工IT事業開始
2006年(平成18年)5月	九電工福岡支店ビルに移転

2008年(平成20年)7月	BEMS事業開始
2010年(平成22年)4月	データセンター事業開始
2013年(平成25年)4月	太陽光発電計測監視サービス開始
2018年(平成30年)2月	株式会社システックを子会社化
2018年(平成30年)7月	港湾カメラ監視システム施工開始
2019年(平成31年)1月	病院ネットワーク構築
2020年(令和2年)4月	鹿児島事業所・壱岐事業所開設
2020年(令和2年)4月	スマホアプリ開発開始

6. インタビュー要約



【プログラミング言語について】

- Java・PHP が主流であるが、顧客の指定があれば、その限りではない
- ごく一部、C#・ルビーを使用している
- 組込系はあまりやっていないので、C言語・C++はあまり使用していない
- 一つの言語を理解すれば応用が利くので、Java を突き詰めて習得すると、他の言語へ展開しやすい
- オブジェクト指向で開発しており、フレームワークを活用している
- フレームワークを使った開発経験があればよい

【プログラミングテストについて】

- 単体テスト・結合テストを実施しているが、本番環境で動かさないと分からないバグもあり、障害対応のためには、インフラ系の知識(OS上でどう動いているか)も必要である
- 単体テストはホワイトボックステストを実施している
- テストの結果を見て原因工程を分析することが重要である(仕様ミスなのかコーディングミスなのか?)
- プログラミングテスト専門会社に委託するケースもある

【採用基準について】

- 書類や面接ではプログラミングスキルはわからないので、対人関係スキルで判断する部分は大きい(人柄 8 割、プログラミングスキル 2 割)
- どういう言語を経験してきたかは聞く
- プログラミング経験が全くない人でも、コミュニケーション能力が高いため採用するケースがある
- 中途採用に関しては対人関係スキル・マネジメント能力を見る
- 技術のみで採用する場合、話の内容からよほど出来そうに見える場合である
- ゲーム業界と違い、プログラム作品を持ち込む人はいない

【専門学校でのプログラミング教育について】

- 遠隔授業で、企業から課題を出してもらって学生がそれに取り組むことは、採用評価の基準として面白い取り組みだと思う
- 作成するプログラムのレベル感は重要だと思う
- 学生チームでプログラム作成を行う場合、概ね一人がプログラミングして、あとは傍観者となるケースがあるが、企業の採用時のチームプログラムテストでも同様であり、できる者が一人居たら完成する
- プログラミングのプロセスを見たい

【業務スキルについて】

- 新規開発案件と改修案件では、改修案件の方が多く、顧客の業務理解が重要
- 新卒プログラマでも全く人と喋らない仕事はないので、成長する過程で会話力を向上してほしい

【新入社員教育・配属について】

- 入社後 3 ヶ月間は外部の研修専門会社(福岡ソフトウェアセンター)の「IT エンジニア育成コース」で基礎スキルを学ばせる

- 研修ではビジネススキルのほか、コンピュータサイエンス、開発ドキュメントの作成、データベース技術、ネットワーク技術、アルゴリズム、情報セキュリティ、データ管理、Java、HTML、CSS、オブジェクト指向プログラミングなどを学習する
- 未経験者は、アルゴリズムでつまづくケースがある
- 新入社員には、内定時に基本情報技術者試験の書籍を渡して勉強を促している
- ネットワーク系・インフラ系どちらの業務に配属するかは、本人の希望もあるが、適性で判断する

【OJTについて】

- 3ヶ月の研修後、仮配属となり、部門研修を実施する。その後正式配属となり、OJTがスタートする
- OJTはできるだけ年齢の近い先輩が指導役となるが、30歳代の先輩社員が指導役となることもある
- 例えばRPAの開発チームに配属となった場合、学校ではやっていないのでOJTで1から教育している
- システム開発部門のOJTは、指導役の先輩が抱えている案件となるケースが多い
- 新入社員にプログラムを作成してもらったケースもある(チェックは厳重に行う)
- 指導役の先輩がテスト案件を抱えていれば、テストをやってもらう
- 開発工程は1年を超えるものから3ヶ月くらいのものである
- 開発は単独で行う場合が主流であるがと、計測メーカーと協業で行うという取り組みもある

【採用時の資格保持について】

- 入社時に情報処理技術者試験を取得していることは、プログラミング技術を持っていることの証明とはならないが、頑張った経験としては評価している。また、基礎知識があるという点でも評価している
- 入社後に情報処理技術者を取得する場合、合格時の受験料は会社が負担する
- 会社の指定する資格(情報処理技術者試験を含む)を取得した場合、報奨金を支給している(一時金であり、継続ではない)

【採用側の視点】

- 大学生の場合、卒業論文のテーマは採用時に見る
- 学生時代に成功したことよりも、失敗したこと、苦労したことを言って欲しい
- 採用側からすると、内申書にはいいことしか書かれていないので、もっと個性が分かる部分を記載して欲しい
- 面接で聞くことになるので、内申書は正直な方がいい

【専門学校教育に対するリクエスト】

- チームで取り組むものをもっとテコ入れして欲しい
- HTTP のプロトコルや TCP/IP に関しても、業務上必要な知識であるため、基礎知識はあったほうがいい
- あらかじめバグを埋め込んだプログラムでトラブルシューティング的な授業があるといい
- 近年、ローコード開発が流行しているが、プログラムがどのような原理で動作しているのかを理解してもらいたいため、教育ではオンコーディングを学んで欲しい

システム・アルファ株式会社

1. ヒアリング日時

令和3年11月16日(火) 14:00 ~ 16:00

2. 訪問者

富山情報ビジネス専門学校 教務部 学科長 山田 太
専門学校中央情報大学校 事業開発室 プロジェクトリーダー 小澤 慎太郎

3. 対応者

システム・アルファ株式会社 SIビジネス部 取締役 統括部長 岡田 健吾

4. ヒアリング場所

システム・アルファ株式会社

5. 企業情報

(1) 基本情報

企業名	システム・アルファ株式会社
所在地	〒371-0847 群馬県前橋市大友町 2-23-5
Web サイト	https://www.system-alpha.co.jp/
設立	昭和56年9月3日
社員数	189名(令和3年4月現在)
経営理念	<ul style="list-style-type: none">● 人と人とのつながりを大切にする お取引先のご担当者様と当社の社員との間、あるいは来店されるお客様と当社のスタッフとの間はもちろんのこと、社員同士、スタッフ同士など、すべての場面で、わたしたちが創業の理念として共有し、一番大切にすべきことだと考えています。● すべてのお客様にプラスアルファのサービスを提供する システムの機能、サービスの品質、提案の内容など日頃の仕事をを行う上で、これを実践したいという思いからシステム・アルファという社名が誕生しました。● 事業を通じて地域社会の活性化に貢献する 基本は「各事業がお客様の事業活動や生活に役立つ」ということであり、それが地域の雇用や消費につながると考えています。そして、日常の行動や作業の積み重ねが、社会への貢献に結び付くと意識して、

	行動できる集団でありたいと思っています。
事業内容	<ul style="list-style-type: none"> ・システムインテグレーション 基幹系業務システム(販売管理、生産管理)導入コンサルティング、設計、開発、アフターサポートなどのトータルサービス ・インターネット関連 ASP、Web システム開発、ハウジング／ホスティングサービス、コンテンツ作成 など ・ネットワークエンジニアリング ネットワーク導入コンサルティング、設計、構築(LAN／WAN／VPN、電話交換機設備、仮想化基盤)など ・情報セキュリティ 個人情報／機密情報の保護及び漏洩防止ソリューション導入コンサルティング、構築 など ・ソフトウェアプロダクト受託開発 ・複合機販売 カラープリンター、総合文書管理、基幹連携ソリューション など ・IT 保守 インフラ、アプリケーションの保守・運用サービス ・ビジネスプロセスアウトソーシング 業務プロセスのコンサルティング、IT 技術の提供・指導、人材派遣、業務代行 ・法人向けモバイルネットワーク 導入コンサルティング、設計、開発、付帯サービス ・Android／iOS 対応システム 導入コンサルティング、設計、開発、コンテンツ作成、付帯サービス ・ドコモショップ運営 ドコモ携帯情報端末・各種商品の販売、契約、保守など各種サービス ・オフィスサプライ販売 リサイクルトナーカートリッジ、OA サプライ、印刷帳票、ASKUL ・ビジネス版メッセージャーサービス向け Bot 開発 direct、LINE WORKS

(2)沿革

1981年(昭和 56 年)	富士通コンピュータ・OA 機器ディーラーとして前橋市に設立される
----------------	----------------------------------

1984年(昭和59年)	現在の本社所在地に新社屋完成・移転
1985年(昭和60年)	電話交換機等の通信基本ソフト、各種ターミナル、POS等の基本ソフト開発を開始
1987年(昭和62年)	xerox複写機、PLUS製品の取扱開始
1999年(平成元年)	機構改革、事業部制の導入を図り、総合的なソリューションサービスを提供できる体制の整備を行う
1995(平成7年)	ドコモショップ前橋西店オープン
1998年(平成10年)	ネットワーク関連事業の強化 東京支社開設
2000年(平成12年)	サーバセンター開設
2003年(平成15年)	車両運行管理サービス「DoCoです・CarALPHA」をリリース
2006年(平成18年)	ドコモショップイオンモール高崎店オープン、個人情報の取扱いを適切に行う事業者が付与される「プライバシーマーク」制度の認定を受ける
2007年(平成19年)	ドコモショップけやきウォーク前橋店オープン
2008年(平成20年)	サーバセンター(SIビジネス部 e-SERVICE!、サーバセンターおよび関連部署)にてISMS認証取得
2013年(平成25年)	システムインテグレーションサービス業務、ソフトウェアプロダクト受託開発業務、並びにそれらの保守サポート業務においてISO9001認証取得
2017年(平成29年)	ビジネス版LINE「LINE WORKS」パートナー基本契約締結および取扱開始



6. インタビュー要約

インタビューに先立ち、社内の意見をアンケート集約してくださいました。本文のランキング部分は社内アンケート結果を示すものであり、それ以外の部分は岡田統括部長からのヒアリング内容となります。

なお、アンケート回答者は次の属性の 30 名でした。

	5 年未満	5 年以上	10 年以上	20 年以上	30 年以上
一般職	8 名	5 名			2 名
リーダ		1 名	6 名	4 名	1 名
マネージャ			1 名		2 名



【習得に力を入れているプログラミング言語】

- SQL
- C#、Visual Basic
- Java script
- Python

- 主力パッケージ商品に C# を使っているため、C# には力を入れている
- 地元ユーザーがターゲットで、販売管理システムなどの基幹システムを開発している
- Python は人気がある。高校生が授業で習得し使っているのでベンチマークしている
- Java は標準的に使っている
- Java script や PHP は Web 系の開発に使っている
- MATLAB は研究開発でシステムを構築したときに使った

【ペアプログラミング経験の評価について】

- 人材育成目的でペアプログラミングは(自部署では)導入している
- 他部署ではやっているようでやっていない(レビューのレベルである)
- 教育の観点で導入しているため、リスクヘッジにはなっているが開発コスト面で課題
- 現在、在宅ワーク導入で出社率は 50% であり、若手社員のコミュニケーション不足が課題となっている。ペアプログラミングにより、わからない点が放置されるリスクは減少した

【リモート環境下で開発物完成経験の評価について】

- タスク管理がきちんとできているのであればリモートは有効だと考えている
- 出来る人はどこでも出来るので、リモートかそうでないかはあまり関係ない
- JOB 型の業務であれば、受注した業務を分担して複数名でやるため在宅でも問題ない
- お客様から常に問い合わせがあり、JOB が定型化できない業務は在宅だと非効率
- 学生の場合、テーマに基づいて自分でスケジュールを決めて実行する場合には有効だと考えている

【リモートワークの課題】

- 教育(特に新人教育)、コミュニケーション不足
- 情報セキュリティ
- 業務効率や生産性の低下
- 開発プラットフォームには GitHub を使っている

【ノーコード開発の評価について】

- ノーコードでもフルコードでも顧客価値を創造することは変わらない
- 顧客企業はフルコードで作る人間を求めている。特に大手 IT 企業は内製ではフルコードでシステムを作らないことを宣言しており、ものを考えられる人や、人を使って仕事を任せられる人を求めている
- フルコード開発の習得には5年くらいかかる。また作れば作るほどバグが入る可能性が高まる。それよりも数をこなす方が効果的

【検定取得実績について】

- 検定取得実績は通常業務での実績とは区別している
- オラクルライセンスを持っていても実務で効果を出せない者もいる
- 資格取得には一時金を支払っている。会社としてはPRになるので資格取得を奨励している。(資格保有者数によってパートナー契約のランクが変わるなど企業評価が上がる)
- 専門学校生には資格取得してほしい。持っている人を評価する
- 国家資格とベンダー系資格では、国家資格に重きを置いている

【プログラマが1ヶ月でコーディングする規模】

- 1ヶ月の仕事が設計・製造・試験に分かれて3分の1ずつだとしたら、C#の新規プログラムで2キロを目安にしている
- この質問により、指標が現場に行き渡っていないことに気づいた

【プログラマの能力評価について】

- コードの半分以上はエラー系であり、そこが想定できているかどうか能力的にチェックすべきところである
- 入社後はプログラマ→SE→マネジャーというステップになるが、採用試験で見るのは「人」。最優先は文化に馴染めるかどうか。採用試験で小論文は書かせるけれどプログラミングテストはやらない(Web系採用ではポートフォリオが有効だが、プログラマはポートフォリオを持っていないので、人物評価になる)

【言語やアルゴリズム以外で求める能力】

- コミュニケーション能力は絶対に持っていた方がいい
- 仲間意識を持って一緒に仕事ができるかどうか最優先
- プログラミングを教えるよりも、コミュニケーション、ディベート、プレゼンテーションを磨き上げた方がいい
- プログラムと関係のないプロジェクトも動いている(公立学校の先生はインフラ・ネットワークのトラブルに対して手を打てないので、学校にサテライトオフィスを設けて常駐させる計画がある。トラブル対応はするがプログラムは作らない)

【プログラマに求められるもの】

- 学習意欲
- 読解力
- 伝える力
 - その他・・・プログラミングを楽しむ気持ち、プログラミングスキル、コミュニケーションスキル、ロジカル思考能力、モチベーション管理能力

【入社後のプログラム教育】

- 文系学生も採用しているので、社内で教育している
- 一部外部研修機関にも出している(テスト手法など)
- 約3ヶ月間教育(C#演習など)をして現場に入り、OJT を経て製品開発に携わる
- 就職前のアルバイト期間にも製品に携わらせる。ただし本物の製品は触らせない。疑似環境で触らせることはしている
- 過大な負荷が係った場合などのトラブルのリスクは伝えないと分からない(トラブルが起こって始めてわかることもあるので、社員でも理解していない人もいる)
- データベース系のトラブル(クラウドへの移行時等)は多いが、システムサポートを専門にしているコンサルティング会社に入ってもらうこともある。わからない社員が徹夜でやるよりも、高額だがコストを支払って解決してもらった方が早い分野もある

【プログラミング教育で今後強化して欲しい領域】

- スクラッチ開発
- アルゴリズム
- PG レビュースキル
- コミュニケーションスキル
- チームでのプログラムの作成
- クラウドサーバーの展開・管理(Amazon Web Service 等)
- AI 領域
- Web システム、クラウドネイティブ
- ロジカルシンキング
- UX への理解(利用者への体験に即したアプリケーション作成)

【プログラミングスキルの評価方法】

- プログラミングスキルに限ると計測できていない

【プログラミング規約】

- メーカーと基本契約を結んでいれば規約があるのでそれに従う
- プロジェクトごとに規約があるのでそれに従う
- ノーコード開発の場合、コーディング規約が意味を持たないことも出てくるので、そういったところは整理されていない

【テスト設計の指針】

- ISO9001 の関係でテスト計画の作成等、テスト基準はある
- 本質は PDCA を回すこと

- テスト技術よりもテストシナリオを書けることが大事
- 大手 IT 企業はテスト自動化ツールを使っている

北電情報システムサービス株式会社

1. ヒアリング日時

令和3年11月24日(水) 16:00 ~ 17:00

2. 訪問者

富山情報ビジネス専門学校 情報システム学科 清水 大樹

富山情報ビジネス専門学校 情報システム学科 山本 晋平

3. 対応者

北電情報システムサービス株式会社 マネジメント部 専任部長 室谷 千秋

北電情報システムサービス株式会社 システム開発部 開発第3グループ

ITソリューション第1チーム 副課長 齋藤 靖彦

北電情報システムサービス株式会社 システム開発部 開発第3グループ

ITソリューション第2チーム 副課長 野田 利絵

北電情報システムサービス株式会社 システム開発部 開発第3グループ

ITソリューション第2チーム 塩苅 敏治

4. ヒアリング場所

北電情報システムサービス株式会社

5. 企業情報

(1) 基本情報

企業名	北電情報システムサービス株式会社
所在地	〒930-0004 富山県富山市桜橋通り3番1号(富山電気ビル2F)
Webサイト	https://www.hiss.co.jp/company/
設立	1987年4月1日
従業員数	341名(2021年4月1日現在)
経営理念	HISS は知的中枢を担う、総合情報サービスを目指しています。
事業内容	システムの構築と保守・運用、及び機器販売、 データセンターサービス(クラウド、ハウジング)、 インターネットサービス、セキュリティサービス、その他
主な事業実績	電気事業システム、工事業システム、通信事業システム、製造業システム、その他

(2)沿革

1987年(昭和62年)4月	北電情報システムサービス株式会社設立
1989年(平成元年)7月	富士通代理店契約締結
1990年(平成2年)4月	一般第二種電気通信事業(FITVAN) 営業開始
1995年(平成7年)8月	石川支店開設(現:石川ソリューションセンター)
1996年(平成8年)7月	福井支店開設(現:福井ソリューションセンター)
1996年(平成8年)8月	インターネットプロバイダー事業開始(FITWeb)
2000年(平成12年)2月	プライバシーマーク認定取得(富山県内企業初)
2001年(平成13年)4月	データセンター事業開始
2001年(平成13年)4月	ISO9001 認証取得
2002年(平成14年)4月	東京ソリューションセンター開設
2003年(平成15年)3月	Oracle パートナー契約締結
2004年(平成16年)5月	ISO/IEC27001 認証取得
2004年(平成16年)9月	SAP サービスパートナー契約締結
2005年(平成17年)5月	安全安心マーク認証取得
2007年(平成19年)2月	「SAP AWARD OF EXCELLENCE」を受賞
2011年(平成23年)6月	IDC 事業拡大
2014年(平成26年)7月	FIT-iDC データセンター増床
2015年(平成27年)6月	建設業許可取得
2018年(平成30年)6月	ISO/IEC27017(クラウドサービスに関する情報セキュリティ管理策のガイドライン) 認証取得
2018年(平成30年)7月	労働者派遣事業許可取得



6. インタビュー要約



【習得に力を入れているプログラミング言語】

- Java・・・ IT ソリューションチームにおいて、新入社員教育ならびに実案件で利用してい

る。

- C#・・・ITソリューションチームにおいて、実案件で主として利用している。
- Java script・・・Webシステムの開発において利用している。
- ABAP・・・ERPチームにおいて、主として利用している（業務用件で必要）。
- Python・・・AIチームにおいて、主として利用している（業務用件で必要）。
- 企業の基幹系システムの開発が多く、仕様が決まったものに対して要件定義から入るウォーターフォール型の開発手法を主として利用している。
- アジャイル型で実施する案件は少ない（ウォーターフォール型開発の一部に取り入れるケースはある）。

【ペアプログラミング学習経験の評価】

以下、すべて配属後の現場の評価です

- 現状はOJTとして実施している程度のため、評価にはつながっていない。
- 設計書通りに正しくコーディングすることは正確なプログラミングができるという観点から、評価につながると考える。
- プログラミング経験(何を作ってきたか)は意識しない。
- 各部署配属後に個人のレベルに見合った教育を実施するので、未経験者であっても問題ない。
- 大学卒/専門学校卒は意識しない(入社時点での技術差は意識しない)。

【リモート環境下での開発経験の評価】

- 経験が0よりはあった方が良いが、必須ではない。
- コロナ禍でリモートワークを余儀なくされていた時は、コミュニケーションを取るのが大変だった。
- リモート環境下でのプロジェクト活動経験は役に立つと考える。
- プロジェクトを可視化するツールとして、TimeTracker（デンソークリエイト）、Lightning Review(デンソークリエイト)を活用している。
- Web会議では主に zoom、Microsoft Teams、Webex(シスコシステムズ)を利用している。

【ノーコード開発の評価】

1. ノーコード開発はツールを使った製造手段の一つであり、この経験自体の評価は難しいと考える。また、ノーコードの経験のみの場合も評価は難しい。
2. ツールによっては、ツールのみでは実現できない仕様に対してコードを記述すると、ツールから読み込めなくなるケースがある。利用する上では、考慮が必要。

3. ノーコード開発は IT 業界よりも例えば製造業の IT 部門など社内システム部門に浸透している ような印象を受ける

【検定取得実績の評価】

1. 国家資格・ベンダー資格共に、入社・配属には有利に働くと思う。
2. 年度毎に計画する個人目標においても、資格取得を目標とするケースがある。
3. 目標は評価のために定量的であることが条件で、資格取得以外だと開発リーダー経験や障害対応、プレゼンスキル、主担当案件の件数や割合を目標とするケースがある。
4. 評価ならびに入札案件の参加条件として必要となるケースがあるかもしれないが、実務上はほとんど意識することがない。

【プログラマが 1 ヶ月のコーディングする規模】

- コーディング量については通常評価していないが、見積の際に利用する弊社の FP 見積標準を参考とすると、1 人月での製造規模(COBOL ステップ)はおよそ 1K ステップ。

【プログラマの能力の評価基準】

- プログラマとしてではないが、プログラミング能力の評価はスキルマップを基準としている。
- 一般・・・言語特性を理解し、プログラムを作成ができる。
- 中堅・・・言語特性を理解し、正確なプログラムを作成できる。
- 上級・・・言語特性を理解し、正確かつ効率的なプログラムを作成できる。
- 上位・・・プログラミングの指導ができる。
- スキルマップを元に、年度ごとに個人の能力開発テーマを決め、年度末に評価している。

【プログラム言語、アルゴリズム以外に必要な能力】

- コミュニケーション能力(報連相ができること)
- 協調性
- 最後まであきらめないこと
- 設計書を読む力、理解する力
- 自己学習能力

【プログラム経験なしの者を自社で教育しているか】

- 4 月～5 月、新入社員は全員一律で入社時教育を受講する(Java、開発手法、アルゴリズム、プログラム言語ごとの技法。20 年前は COBOL でアルゴリズムを覚えた)。
- 6 月～8 月は現場で OJT、教育内容を先輩が考えて個々に合わせた教育を実施。
- 9 月以降は実案件に参画し、案件を利用して教育を実施。ペアプログラミングで先輩が

補佐する場合もある。

- 案件は、汎用機から最新版のパブリッククラウド活用まで幅広い。
- 学生時代の経験の差による社内教育の差はない。

【今後のプログラミング教育で強化して欲しい領域】

- Web セキュリティの知識・技術は意識して勉強して欲しい。
- コードを書くだけでなく、詳細設計、プログラム設計、コーディング、単体テスト仕様書作成、単体テストまでをセットで習得して欲しい。
- プログラミング時には参考となるソースコードを事前に提供することが多い。よって、1 からコーディングする能力よりも、ソースコードを読んで理解して改変できるスキルの方が役立つと考える。保守等の仕様変更時にも有用である。

【テスト設計の指針】

- 社内にテスト設計の指針はある。顧客ごとに存在する場合もある。
- テスト計画、単体テスト、結合テストおよびシステムテストについて、記載の粒度や実施時の観点を記載した資料がある。
- 初めての案件はテスターからという事もある。真摯に対応してくれる人材は重宝する。
- 不適合製品を出してしまうと、お客様の業務へ影響を与える可能性があるとともに信頼の失墜にもつながるため、テストは非常に重要と考える

株式会社北陸ソフトアス

1. ヒアリング日時

令和3年11月26日(金) 10:00 ~ 11:00

2. 訪問者

富山情報ビジネス専門学校 情報システム学科 清水 大樹

3. 対応者

株式会社北陸ソフトアス システム開発部 岩井 瑞樹

4. ヒアリング場所

富山情報ビジネス専門学校(オンラインにて実施)

5. 企業情報

(1) 基本情報

企業名	株式会社北陸ソフトアス
所在地	〒933-0911 富山県高岡市あわら町 8-25
Web サイト	https://hokuriku-softas.co.jp/company/
設立	2008年8月29日
社員数	42人(2021年10月現在)
経営目標	経営目標 お客様を成功へ導くために 1、社員の豊かな生活 心身及び、経済的に社員の充実した生活基盤となりうる企業を目指します。 2、社会的信頼の獲得 北陸ソフトアスの社会的信頼度を向上させることで、所属する社員の個人的信頼度を高めることを目指します。 3、社会貢献の実現 社員個々人が社会貢献を実感できる企業を目指します。
事業内容	インフラ構築・運用サービス システム開発サービス ビジネスソリューションサービス

(2)沿革

2008年(平成20年)8月	設立
2008年(平成20年)10月	IT事業を開始
2018年(平成30年)1月	資本金4000万円に増資
2018年(平成30年)4月	事業目的変更に伴い組織を変更
2019年(平成31年)3月	資本金7000万円に増資
2020年(令和2年)5月	プライバシーマーク取得
2020年(令和2年)6月	労働者派遣事業許可取得

6.インタビュー要約

zoomで実施



【習得に力を入れている言語】

- Java } Webアプリケーションの開発 7割
- Ruby } (例:リフォーム会社の見積請求+人材管理システム)
- C#・・・3割
- システム開発部の教育期間は人によって異なるが、社内で約2ヶ月間行う(ビジネスマネー・電話対応・LINUXサーバ構築・Webアプリケーション研修制作1ヶ月)
- インフラソリューション部では、客先でのLAN構築やサーバシステム構築の業務を行っている

【学校でのペアプログラミング経験は評価につながるか？】

- ある程度は評価につながる
- 実務ではコミュニケーション能力を求められるが、ペアプログラミング経験でその能力がついていれば評価できる
- ペアプログラミングでの成果物よりも、ペアでのコミュニケーション・情報共有を重要視している

【リモート環境下での開発経験は評価につながるか？】

- 評価につながる
- 実務でのリモートワークは現在も進行中(会社のデスクは一部フリーアドレス)
- 現状リモートワークを基本として、必要に応じて出社している
- コーディングで詰まった際には、上司に質問するために出社することもある
- アフターコロナの環境でもリモートワークは継続していく見込み

【ノーコードでのシステム開発経験の評価】

- ノーコード開発は、開発経験としてはあまり重視していない
- 上流工程からプログラミングまで全工程ができる人材育成を目指しているので、ノーコード開発で上流工程だけができて評価にはつながらない

【検定取得実績は通常業務での実績と区別するか】

- 人事評価の対象にはなるが、資格検定取得によって実務で重要業務を任されるかと言うと、それは別である
- 基本情報技術者試験、オラクルマスター(ブロンズから)などが評価の対象となる
- 資格取得は、会社が経費支援している
- (会社の目標管理制度の)個人目標として、資格取得を設定している

【プログラマが1ヶ月にコーディングする規模】

- 数千ステップ程度(なかには数万ステップを1ヶ月でコーディングするメンバーもいる)
- 案件や経験年数によって異なり、ステップ数自体を評価基準にはしていない

【プログラマの能力の評価基準・評価方法】

- 評価は年2回(上半期・下半期)
- 資格や知識ベースでの評価プラスOJTでの評価を行っている(ソースレビュー等)
- 評価基準(約200項目)はあるが、OJTベースでの評価が加味される
- 「データベース設計をどこまでやったことがあるか?」「開発環境構築を行ったかどうか?」「ユニットテスト・結合テストの実績」「成果物の保守計画の作成」などの項目で自己

評価を行い、その後上司からの他者評価があり、最後に上司との面談を行う

【プログラマに言語・アルゴリズム以外で求める能力】

- プレゼンテーション能力(ソースレビュー・他者への意図の説明)
- 授業科目「技術発表演習」はよかった、もう少ししっかりやっておくべきだった
- 論理的に物事を説明する能力は必須
- 頭の中でロジカルに思考できていたとしても、それをアウトプットできることが重要

【プログラミング経験なしの者の自社教育】

- Web アプリケーション開発の社内研修を行った後、現場での経験を積んでいく
- 同期入社 4 名のうち 3 名(短大卒)はプログラミング経験なしだった(一部学校の授業で PHP 言語程度をやった人はいる)
- プログラミング経験だけでなくコミュニケーションや人柄も重視される

【求める教育カリキュラム】

- 開発環境構築(Ruby の環境開発構築など)や上流工程の学びが少なかった
- プログラミングに限らず上流工程も含めて満遍なく学べるカリキュラムであれば理想
- テスト設計・実施は在学中に全くやっていないので、テストの部分はやったほうがいい
- 他者レビューも授業ではあまりやっていなかったが必要
- コーディングに力が入っているが、システム開発全体の力を伸ばすことが求められる

【プログラミングスキルの評価制度】

- 自己評価・他者評価のシートがあり、それがベースになっている
- 社内でプログラミングスキルによる階層分けはしていない
- TechFUL のような指標・制度は取り入れていない(今後導入の可能性はあるかも知れないが現段階ではない)
- TechFUL が社内で話題に上がったことはない

【テスト設計の指針】

- テストは上流工程を除いて最も時間をかける部分
- テストをしっかりとやらないと納品できないので極めて重要である
- テスト設計の指針は存在している。テスト項目を作成してテスト仕様書をレビューしてもらい、テスト実施し、テスト後にもテスト結果をレビューしてもらう

【専門学校への要望事項】

- プログラミング技術以外の部分(学生の時にしか学べない部分)を強化する方がよい

- プレゼンテーションの授業はよかった(社会人になってからプレゼン能力を学ぶ機会はない)
- 資格は学生の時にとっておいたほうがいい(社会人になると学生と比べて圧倒的に時間がなくなる)

株式会社トスバックシステムズ

1. ヒアリング日時

令和3年12月2日(木)13:00 ~

2. 訪問者

富山情報ビジネス専門学校 教務部 学科長 山田 太
専門学校穴吹コンピュータカレッジ 副校長 産学連携センター 戸倉 潤也
専門学校穴吹コンピュータカレッジ 教務部教務課 課長 塩田 和正
専門学校中央情報大学校 事業開発室 プロジェクトリーダー 小澤 慎太郎
日本電子専門学校 エンジニア教育部長 ケータイ・アプリケーション科 科長 大川 晃一

3. 対応者

株式会社トスバックシステムズ ソリューション事業統括部 ソリューション技術部
部長 湯谷 政春
株式会社トスバックシステムズ 管理部 部長 真鍋 匡司
株式会社トスバックシステムズ 管理部 業務グループ 主任 小林 裕子

4. ヒアリング場所

株式会社トスバックシステムズ

5. 企業情報

(1) 基本情報

企業名	株式会社トスバックシステムズ
所在地	〒760-0066 香川県高松市福岡町4丁目5番1号
Web サイト	http://www.tosbac-systems.co.jp/profile.html
設立	昭和56年4月13日
社員数	112人(令和3年4月現在)
内容	印刷業ソリューション「ひだりうちわシリーズ」 承認&グループウェアソリューション 市場ソリューション 建設ソリューション 運輸ソリューション 漁協ソリューション 映像提供ソリューション 流通ソリューション

	生産管理ソリューション プラットフォームソリューション 浄化槽ソリューション その他ソリューション
--	--

(2)沿革

1981年(昭和56年)4月	株式会社トスバックシステムズ設立
1986年(昭和61年)10月	高知営業所開設
1988年(昭和63年)9月	東京営業所開設
1991年(平成3年)4月	販売管理パッケージ「じょうでき」販売開始
1992年(平成4年)10月	松山営業所開設
1995年(平成7年)10月	事業企画推進部発足、システム保守サービス開始
1996年(平成8年)4月	東京支店開設(東京営業所昇格)
1997年(平成9年)4月	印刷システム事業部発足(東京支店内) 印刷業総合管理システム「ひだりうちわ C/S」販売開始
1998年(平成10年)4月	印刷業総合管理システム「SUPER」販売開始
2000年(平成12年)4月	eBusiness 事業グループ発足
2000年(平成12年)6月	印刷業総合管理システム「ひだりうちわ 2000」販売開始
2002年(平成14年)2月	姫路営業所開設
2003年(平成15年)4月	印刷業総合管理システム「新ひだりうちわ SUPER」販売開始 販売管理パッケージ「じょうでき 5」販売開始
2005年(平成17年)2月	PAGE2005 に「ひだりうちわシリーズ」を出展(以降、毎年出展中)
2005年(平成17年)2月	「市場総合システム」を販売開始
2006年(平成18年)4月	3事業部体制発足(ソリューション事業部、印刷システム事業部、東京支店)
2006年(平成18年)7月	印刷システム事業部/関西事業所 大阪市淀川区へ事務所移転
2006年(平成18年)9月	プライバシーマークの認証取得
2008年(平成20年)4月	建設システム「じょうでき PLUS」販売開始 アパレル業トータル管理システム販売開始
2009年(平成21年)9月	印刷システム事業部/大日本スクリーン製造株式会社と技術提携
2010年(平成22年)9月	印刷システム事業部/西日本事業所 広島市中区八丁堀

	へ事務所移転
2011年(平成23年)12月	東京支店において情報セキュリティマネジメントシステムの認証取得
2012年(平成24年)4月	プラットフォーム事業部を新設、4事業部体制スタート
2012年(平成24年)11月	ソリューション事業部／松山営業所 松山市天山へ事務所移転
2013年(平成25年)1月	印刷業総合管理システム「ひだりうちわ PriNOS」販売開始
2014年(平成26年)6月	ソリューション事業部／福岡営業所・印刷システム事業部／九州事業所 開設
2015年(平成27年)12月	プラットフォーム事業部・管理部において情報セキュリティマネジメントシステムの認証取得
2016年(平成28年)2月	ソリューション事業部／姫路営業所 姫路市東延未へ事務所移転
2016年(平成28年)5月	東京支店・印刷システム事業部／関東事業所 東京都港区へ事業所移転
2020年(令和2年)2月	本社、ソリューション事業部、プラットフォーム事業部 高松市福岡町の自社ビルへ移転



6, ヒアリング要約



【主に使っている言語】

- VB.net…最もポピュラーな言語だと思っている (Windows で動くシステムが多い)
- C 言語、DELPHI…一部で使っている
- COBOL…かつては使っていたが 2000 年ごろにオープン言語に切り替えた
- PHP、Java…Web 系の部署で使っている
- VB.net は学校であまりやっていないが、独学で学んで来る人はいる
- 文系の大学生はゼロに等しい。理工系の大学生でプログラム経験のある人または専門学校卒が大半を占めている
- 現場サイドでは理工系の学生を採用したいという思いは強い
- ものづくりの能力が優先される傾向にある
- 経営層の一部は文系採用でマネジメント職を育てる意向もあるが、現在はエンジニアがマネジメント職に昇格している

【学校でのプログラミング経験の評価】

- 学校で教えることと実務で組むプログラミングレベルは異なる
- 学校での教育では足りていないのが現状
- 業務システムはデータが絡むので、データベースの知識はもっとあった方がいい

- 入社後一定期間(約6ヶ月)は社内で教育する(プログラム言語や帳票ツールの勉強)
- 部署内の採用人数は1~2名なので、OJTで先輩が1名ついて教育をし、上司に報告する(会社全体では5~6名を毎年採用している、状況により転勤もある)
- 東京支店では、その年のプロジェクトに準じて必要な言語を外部研修(3ヵ月程)で習得してもらう。
- データベース、プログラム言語を学んだ後、サンプルプログラムを何本か作らせる
- その後は実務に入って製品を触っていく
- パッケージをカスタマイズするパターンとオーダーで作るパターンがある
- システムとして新しいものを開発するケースもあるが、部品は既存のものを使う

【プログラマの評価基準】

- 半年に1回個別面談があるので、そこで評価し目標を立てている
- 新入社員の評価基準は、Cは簡単な知識がある、Bはプログラムに触れる、Aは問題ない
- 早い人は4月に入社し、8月くらいから実務で開発に入る人材も稀にいる

【学校で学んだ知識が役に立つか】

- プログラミング経験は評価しているがVB.netの経験は特に評価していない
- ロジックを組み立てる力(論理的に考える力)を重要視している
- データの取り扱い方に対するシビアさは実務の方が圧倒的に高い
- 顧客のデータは顧客企業の資産であり、壊したり改変したりしたら賠償問題にもなりかねないので、その取り扱いのシビアさは学校では学べない

【プログラマが1ヶ月にコーディングする規模】

- 個人差とプログラムの難易度にも関係するが、帳票であれば仕様書に基づいて1日1本仕上げるくらい(単体テストまで終わっている)のスピードは求めている。
- 100ステップ+画面の組み合わせで1本。画面プログラムの流用はあり

【コーディング基準はあるか?】

- 共通ルーチンはある(共通部品を使える部分は共通部品を使う)
- 社内にコーディング規約はある(パッケージソフトにだけ適用する規約や、それ以外のものに対する規約など何パターンもある)

【開発手法について】

- ローコード開発・ノーコード開発はない
- 信頼のある部品の組み合わせでプログラムを使っているが、元になるプログラムがあって、顧客に合わせてそれをカスタマイズして使っている

- 社内のキャリアパスは、プログラマからシステムエンジニア、マネジャーの順で上がっていく
- なるべく誰がプログラムを組んでも同じようなものが組めるように教育している

【資格について】

- 採用するときに判断基準にしている(基本情報技術者試験が判断材料になっている)
- 基本情報を持っていなくても採用はするが、持っている方が評価は高い
- 困難に立ち向かった経験として評価している
- 大学生はほぼ資格は取っていない。入社後に挑戦することになる
- ベンダー資格については、オラクル(ブロンズ)を評価していた時期もあるが、今はあまり重視していない(困難に立ち向かった経験としては評価している)
- 資格に対する報奨金や手当はない。開発に必要と認めたものに関しては会社が経費を支弁する
- 官公庁の運用サポートは資格要件があるので、基本情報は取らせることが多い
- 簿記会計の資格は評価している

【リモート環境下での人材育成について】

- 東京は1年半ほど在宅でリモートワークだった
- 東京配属は今年いなかったなので、リモートでの教育実績はない
- 営業はリアルで同行訪問等が必要だと思っている

【テスト設計の基準について】

- 社内に単体テスト仕様書がある(全てその仕様書でやっているわけではない)
- 結合テスト、システムテストにも基準はある
- 結合テストはシステムそれぞれなので、テストデータを作って流してみる。それでも現地の本番環境には対応できない漏れがある

【採用時の評価基準】

- 大学生と専門学校生の違いは2歳の年の差。大学生は余裕感がある
- 枠は決めていないが、どうしても大学生から採用される傾向にある
- 専門学校生だった場合は、資格を取っているかどうかは見る
- コミュニケーション能力があるかないかは見る
- 採用時の技術差は業務で埋められていくが、コミュニケーションはないとダメ(お客さんの前に出ることができない)
- 本社はリアル面接、東京は今回初めてオンライン採用面接を行なった(不安要素はある)
- 福岡では来社して別室でテレビ会議面接を行なった

- インターンシップは1day、グループワークが主体

【遠隔プログラム教育に対する企業参加の可能性】

- 業務主体なので、教育課題の開発に人を割けるほど余裕がないのが現状

【リモートワークのメリット・デメリット】

- 社内でリモートワークのメリット・デメリットについて社内アンケートを行なった
- メリット1位:時間の制約・場所の制約からの解放、2位:快適な作業環境、3位:家事/育児との両立
- 女性エンジニアは3名
- デメリット1位:個人資源を利用することの弊害(PC・回線速度・セキュリティ)、2位:コミュニケーション・状況把握の難しさ、3位:リモートで使用できない機能がある(帳票ツール・遠隔操作ソフト)
- プログラムの製造だけに関してはリモートでも問題ない。コミュニケーションについては課題がある(指示の理解や質問など)

【専門学校に求めること】

- 学校の授業でやることをしっかりと身につけてほしい
- 基礎がしっかりしていれば応用が効く
- 専門学校生には「これしかできない」という視野にならないでほしい
- システム開発の中ではSEの仕事が一番大変だが、会社としてはSEになってほしいのでコミュニケーション能力は重要
- 地場の企業が顧客の中心なので、顧客からの要求がプレッシャーになるケースはある。そうしたプレッシャーに耐えうる人材を求めている

株式会社高志インテック

1. ヒアリング日時

令和3年12月8日(水) 16:00 ~ 17:00

2. 訪問者

富山情報ビジネス専門学校 情報システム学科 新田 雅道

富山情報ビジネス専門学校 情報システム学科 山本 晋平

3. 対応者

株式会社高志インテック システム事業本部 ビジネスソリューション部 副部長 杉野 敬臣

4. ヒアリング場所

株式会社高志インテック

5. 企業情報

(1) 基本情報

企業名	株式会社高志インテック
所在地	〒930-0804 富山県富山市下新町5番23号 インテック富山第三ビル
Web サイト	https://www.koushi-intec.co.jp/
設立	2005年7月
従業員数	557名(2021年6月1日現在)
経営理念	OUR PHILOSOPHY(TIS インテックグループ基本理念) OUR PHILOSOPHY、それは TIS インテックグループの価値観です。そこにはグループの経営、企業活動、構成員において、TIS インテックグループが大切にしている考えやあり方が幅広く明確化されています。TIS インテックグループの全ての営みはこの OUR PHILOSOPHY を軸に行われます。 OUR PHILOSOPHY を以下にご紹介します。 Mission ムーバーとして、未来の景色に鮮やかな彩りを ミッションは、TIS インテックグループが果たすべき社会的役割であり、TIS インテックグループの存在意義です。 ここに掲げた「ムーバー」とは、世の中を新しい世界へと動かしていく

モノやコト、システムを生み出す人のことです。

つまりTIS インテックグループおよびTIS インテックグループ構成員のことです。

TIS インテックグループは デジタル技術を駆使したムーバーとして、未来の まだ見ぬ 景色 の中 に 、社会を魅了する斬新な可能性や選択肢の提供によって 鮮やかな彩りをつける 存在でありたいと考えています。

Style オネストというスタイル

企業行動のもっともベースとなるあり方は「オネスト」—正直、誠実であることと考えています。

なぜなら、オネストは信頼のための不可欠な要件であり、ステークホルダーとの価値交換の大前提となるものだからです。

TIS インテックグループでは、オネストの他「オープン」「パイオニアリング」などのスタイルを大切にしています。

これらは TIS インテックグループ経営の意思決定から企業活動全体にまで貫こうとする私たちの信念でもあります。

Policy 人のためにあること

TIS インテックグループがもっとも大切にしている経営政策は「人のためにあること」です。

なぜなら企業は人の集まりであり、企業を取り巻くステークホルダーもまた人だからです。

人を忘れた経営からは良質な企業活動は決して生まれません。

他にも、良き社会の一員として社会規範や社会の期待のさらに上を歩むことを徹底する「社会を超えること」や、合目的的経営を目指す「目的に向かうこと」などを主たる経営政策としています。

Membership 良き社会のメンバー

メンバーシップとは、TISインテックグループ構成員の行動指針です。そこでは「良き社会のメンバーとして」、および「意思と意見を表すメンバーとして」「進化を担うメンバーとして」の振る舞いがあげられています。

たとえば「良き社会のメンバーとして」は、TIS インテックグループ構成員である以前に良き社会の一員であることを求めたものです。

また、他の2つの指針においては人間らしく自分の存在をアピール

	<p>し、個性・能力を存分に発揮することを願っているものです。</p> <p>企業とは 幸せ追求の社会システムとして 企業の目的についてはさまざまな考え方があり、どのように定めるかによって企業経営は大きく変わります。 TIS インテックグループでは「企業は社会の公器」との考え方をベースに、企業は ステークホルダーと「価値交換 を通して社会の期待や人びとの幸せに貢献 する幸せ追求の社会システム 」と定義しています。 このような、企業はステークホルダー、ひいては社会のために存在するものであるという考えが TIS インテックグループの企業活動の根底に流れています。</p> <p>成長とは 成長 - ミッションに向かって TIS インテックグループでは「ミッションの実現性の向上」と、ステークホルダーとの「価値交換性の向上」の 2 つを成長と定義しています。 「ミッションの実現性の向上」とは、TIS インテックグループが掲げるミッションの実現度が高まること。それはすなわち TIS インテックグループの社会における存在意義が大きくなることを意味します。 また「価値交換性の向上」とは、TIS インテックグループとステークホルダーとが交換する価値の質・量の増強、言い換えれば価値交換における互いの満足度が大きくなることです。 TIS インテックグループはこれら 2 つの本質的な成長を目指しています。</p>
事業内容	<ul style="list-style-type: none"> ・システムソリューション ・システムマネジメント ・リース ・ヒューマ(人材派遣)

(2)沿革

1979年(昭和54年)6月	株式会社北陸インテック技研(本社富山市)設立
2000年(平成12年)1月	株式会社日本インテック技研・株式会社西日本インテック技研と合併
2000年(平成12年)4月	株式会社インテック ソリューション パワーに商号変更
2005年(平成17年)1月	グループ再編により、株式会社インテックの完全子会社になる

2005年(平成17年)7月	株式会社インテック ソリューション パワー(北陸本部)分社化により株式会社高志インテック設立
2007年(平成19年)4月	純粋持株会社インテックホールディングスの完全子会社になる
2007年(平成19年)7月	株式会社アイ・ユー・ケイより同社北陸本部の営業権を譲受
2007年(平成19年)11月	株式会社インテックリースよりリース事業を譲受
2008年(平成20年)4月	株式会社インテックホールディングスと TIS 株式会社が共同持株会社 IT ホールディングス株式会社を設立
2009年(平成21年)10月	株式会社インテックが株式会社インテックホールディングスのすべての事業を承継
2011年(平成23年)1月	株式会社高志インテックと株式会社ヒューマが合併
2016年(平成28年)7月	IT ホールディングス(株)が TIS(株)を吸収合併し、社名を TIS(株)に変更 グループ名を「IT ホールディングスグループ」から「TIS インテックグループ」に改称
2019年(平成31年)4月	株式会社アイ・ユー・ケイにシステムインテグレーション事業を譲渡



6, ヒアリング要約



【習得に力を入れているプログラミング言語】

- 開発案件や企業によって使用している言語は異なる
- その中でも Java は中心的であり、社内で java の勉強会もやっている
- Windows の業務系システムは VB.net、C #が多い(運用系は少し違う)
- 過去のシステムでは COBOL などもある

【学校でのペアプログラミング経験の評価】

- ペアプログラミングの効果は認めているが、それだけで判断・評価することは難しい

【リモート環境下での開発経験の評価】

- 学校でのリモート環境下での開発経験は評価につながる
- 成果物の評価は品質・納期になるので、リモートであってもアウトプットが出せれば評価につながる
- 社内でもリモートワークは継続しているが、隣の席にいないとダメだということはなく、対面との差はあまりない(生産性にはあまり影響しない)
- 学校でリモートのコミュニケーション経験はあった方がいい
- リモートなりのコミュニケーションの取り方があるので、慣れていているという点で評価できる

【ノーコード開発経験の評価】

- 現時点でノーコード開発をそのまま評価するのは難しい
- ノーコード開発にはプログラムスキル以外の部分が必要で、要件定義の学習といった面では経験として評価できる

【検定取得実績について】

- 目標管理で資格取得を目指すなど評価につながる
- 資格取得は一定期間、毎月の手当に反映される(一時金はない)
- 役職者の要件として資格取得はない

【プログラマが1ヶ月でコーディングする規模】

- 言語にもよるが、600~700ライン(0.6~0.7キロ)
- 書くだけの仕事であれば1000行くらいになるが、pythonやJava scriptだとコード量は少なくなるので行数は減る
- 見積段階での予定工数とそのくらいなので参考にしている
- 積算時のコストにもそのくらいを反映している

【プログラマの能力評価基準】

- 目標管理制度を導入しており、業績目標と自分がどうなりたいのかを上期・下期に上長が面談をして評価している
- プログラマとしての評価は、アウトプットの品質が最も評価を左右する要素
- 設計が上手い下手というのは数値化できていないが、そうした部分も(実務や面談等で)見ながら評価している

【言語やアルゴリズム以外で求めること】

- コミュニケーション能力…コンピュータ相手だからといって人と喋らなくてもいいわけではない。仕事なのでお客様とのコミュニケーションは必要
- コミュニケーション能力によって適材適所は考えている

【自社でのプログラミング教育】

- 自社でプログラム教育をおこなっている
- 新入社員対象ではないが、Javaの勉強会などもしている(講師に来てもらい教材を使ってやっている)
- 新入社員には半年間の教育期間があり、アサイン(担当)する予定のプロジェクトに必要なプログラム言語を教えている

- スキルトランスファーで運用から開発に移る場合も社内で教育している

【教育カリキュラム】

- Java、VB.net、C#については社内に教材がある
- 基本はカリキュラムを進めて、その後にプロジェクトに入ってOJTをやる(ペアプログラミングの手法を用いることもある)
- 学校形式(講義形式)でやることもあるが、新入社員が2~3人の場合は、1日のうち何時間かを教育に充て、先輩が時間を取って教えている

【プログラム教育で強化してほしい部分】

- デザインパターンを教えてほしい(オブジェクト指向の理解、クラスの設定、シングルトンパターンなど)
- 基本設計を詳細設計に落とし込むときに、オブジェクト指向のデザインパターンが頭に入っていないと会話が噛み合わない事がある
- 挨拶やマナー、コミュニケーション能力は強化してほしい

【プログラミングスキルの評価基準】

- 目標管理制度で、業績と本人のスキルを合わせて評価している
- 得意な言語などは上長が把握しており、評価している

【コーディング基準】

- プロジェクトに依存するが、コード基準はある
- 細かいプロジェクトだと最初に「このプログラム規約を使う」というのを決める
- ノーコードツールは使っていない

【テスト設計の指針】

- プロジェクトごとに作ってテストをやっている
- キロラインあたりのバグ数などの指針はある
- コーディング終了後のユニットテスト、単体・結合テストなどは社内基準がある。品質を高めるために、テスト結果を見て分析・改善している

株式会社ウチダ人材開発センタ

1. ヒアリング日時

令和3年12月10日(金) 11:00 ~ 12:00

2. 訪問者

富山情報ビジネス専門学校 教務部 学科長 山田 太

富山情報ビジネス専門学校 情報ビジネス学科 清水 大樹

専門学校中央情報大学校 事業開発室 プロジェクトリーダー 小澤 慎太郎

日本電子専門学校 エンジニア教育部長 ケータイ・アプリケーション科 科長 大川 晃一

3. 対応者

株式会社ウチダ人材開発センタ 営業本部ラーニングソリューション部部长 山川 宏樹

株式会社ウチダ人材開発センタ 営業本部ラーニング営業部販売促進課課長 上田 あゆ美

4. ヒアリング場所

ウチダ人材開発センタ

5. 企業情報

(1) 基本情報

企業名	ウチダ人材開発センタ
所在地	〒130-0015 東京都墨田区横網1丁目6番1号 国際ファッションセンタービル 7F
Web サイト	https://www.uhd.co.jp/
設立	1996年(平成8年)4月12日
従業員数	
企業理念	
事業内容	・一般労働者派遣事業 ・有料職業紹介事業 ・ヒューマンスキル教育、IT 技術者教育 ・eラーニング事業 ・アウトソーシング受託

(2) 沿革

1988年(昭和63年)12月	人材派遣業を目的として(株)アプロスを設立
-----------------	-----------------------

1995年(平成7年)1月	(株)内田洋行の人事部人材開発センターを受け入れ、研修事業を開始
1996年(平成8年)1月	大阪府中央区の内田洋行本町ビルに大阪支店を設立
1996年(平成8年)4月	(株)ウチダ人材開発センターを設立し、(株)アプロスの研修事業を引き継ぐ
1996年(平成8年)7月	人材派遣業務を開始する
2002年(平成14年)3月	銀座オフィスを開設
2004年(平成16年)2月	銀座オフィスに本社を移転
2007年(平成19年)10月	プライバシーマーク認定取得
2012年(平成24年)6月	ISO29990 認証取得 ※現在はISO29993
2014年(平成26年)9月	両国オフィスに本社を移転
2021年(令和3年)8月	えるぼし(認定段階3)認定取





6. ヒアリング要約



同社は顧客企業にIT教育サービスを提供している。

【習得に力を入れているプログラミング言語】

- Java…消極的理由(潰しが効く)で受講するケースもある。新入社員教育では、アサインす

るプロジェクトの個別最適化させた言語を習得させてしまうと、同期で共通言語がなくなってしまうので、汎用的な Java を習得させる企業もある

- VB.net…親会社の内田洋行はじめパッケージソフトを VB で開発されている企業が少なくない。そのような企業が弊社を選択してくださっている
- C 言語…意外と研修ニーズは今でも多い。オブジェクト指向の学習をする前に、メモリやポインタなどを理解していた方がよい、と考えている企業が少なくない。プログラムの基礎を習得させる目的でやる。新入社員研修では Java を受講する企業のうち3~4割は C 言語も受講している
- Python…最近増えてきた。大手企業が多い

【同社の研修サービス体系】

- 公開型新入社員研修…厚生労働省の助成金対象講座。カリキュラム・時間が決まって公開しており、希望する企業が申込み参加する。30 日間で時間は 9 時~17 時 30 分。文系学卒者が Web アプリケーションを作れるようになるレベル(開発環境は Tomcat を使用)。詳細仕様書を基にプログラムを作れるレベルを目指している。
- 公開型研修 30 日間のアジェンダは、基本文法→オブジェクト指向→データベース連携→Web アプリケーションの4ブロック
- オブジェクト指向は最小限でいいという考え方もある。オブジェクト指向が理解されていなくてもプログラムは組める。オブジェクト指向は体感覚で覚えていくしかない
- プライベート研修…顧客企業の要望に合わせてカリキュラムをカスタマイズする。個別にカスタマイズするので、プログラム教育だけでなく営業研修やコミュニケーション研修などが加わることもある。プログラム開発職種ではなく、開発会社に発注する側の人材(プロジェクトマネジメント、システム開発委託先との交渉、納品物の検収を担当するスタッフ)を養成するコースを提供したこともある。
- 大手企業ほど、配属先で即戦力となることよりも長期的な人材育成を視野に入れて研修を組むケースが多い。新入社員に求めるスキルが変わってくることと、受講者が多くなるとコストが嵩むので費用対効果を鑑みて設計ができるプライベート研修となることが多い
- 大手企業の一部は研修を発注する際にリバースオークションをかけてコストダウンを図っている企業もある(書類審査である程度絞り込んだ後にリバースオークションで値下げ交渉をしてくる)。

【オンライン研修のツール】

- 基本は Zoom、ブレイクアウトルームも活用する。お客様によっては WebEx、Teams のケースもある
- チャットや情報共有は Slack
- データの提供や課題提出は Google ドライブ

【オンライン研修の活用状況】

- 公開型研修でもリモート対応はしており、集合研修と両方を用意している
- 2020 年はリモートのみだった。2021 年もプログラミング研修は全てリモートだった（緊急事態宣言対応として、集合研修もオンラインで実施した）
- 復習用に動画も用意しているが、基本はオンタイムのライブ研修が中心
- 厚生労働省の助成金対象講座となっているので、9 時から 17 時 30 分で出席確認もする
- リアル講座を求める声は根強い。日本企業の新入社員教育の場合、会社に馴染ませる目的もあるのではないかと。

【リモートは対面と成果が変わったか？】

- プログラミングスキルはオンラインの方がより早い段階で目標に到達できている
- ソースコードを間近で詳しく見られるというアドバンテージがある。
- 受講者同士でも画面レビューできるのでスキルアップに繋がりがやすい
- 確認テストの点数も上がっている
- レベル差が大きくなってきている傾向も感じられる（受講者の層が変わっただけなのかもしれない）
- 理解の遅い人が気楽に聞きにくくなったのかも知れない（講師への質問が、リアル対面と比べて“目立つ”）。
- アルゴリズムを自分で0から考えなければならない時に、顕著にレベル差が現れる
- アルゴリズムの考え方が見当もつかない人が増えたように見受けられる。一方で、言語仕様が複雑な単元でも、パターン応用して作成できる部分では現れない
- グループでプログラムを作る研修は、集合研修で行いたいとの要望はある

【グループ学習の課題】

- スキルレベルを混合にしたグループ編成では、遅い人が技術の高い早い人に頼ってしまい、結果的に遅い人が達成感を得られず自信を育てられないケースがある
- レベル別のグループにした場合には、どのグループも達成感を得て自信を持ってもらやすい
- 講師と受講生の信頼関係を大切に、受講生に頑張ってもらえるよう努めている

【教育成果に対する顧客企業の評価】

- プライベート研修の場合
 - 成果に対する考え方は顧客企業（ニーズ）によって異なる
 - 費用対効果は当然だが、目的達成のためにカリキュラムに合意できるかが焦点。到達

度の設定は企業によって様々。事業投資としてリターンがあるものとして教育を位置付けているのか、新入社員時期の「体験」の場としてとらえているかによっても異なる

- 成果を求める企業においては、テクニカルスキルだけでなくパーソナルスキルも含めて当人たちのレベルや適性を尋ねられることもある
- 教育プロセスの合意形成までもできていることが理想だが、それほど容易ではない
- 教育効果としてのスキルの判断基準を提示している企業はあまりないが、1年以内にオラクルマスターJava ブロンズを取得することを指標にしている企業もある
- 新入社員教育における技術的な基準は、開発企業の場合配属時にプログラミングができることであるが、発注側企業の場合はコミュニケーションスキルが求められることもある

- 公開型研修

- クラス設計や演習問題の実績、プログラム行数などを顧客企業に報告するが、相対評価(順位)を聞かれることもある

- 共通

- 新規開発案件は少なく、既存システムの改修の方が多いのが実務の実態。従って、教材も同様であるべきか、議論を続けている
- 新規開発型学習の場合、市販品の教材を利用できる。一方、改修型の教材は市販にはないため、教材開発の必要が生じる
- 動画も撮影している。復習に利用できる、また体調不良等による欠席時のフォローアップに使える

【フレームワークの活用】

- 30日間の公開型講座の後にオプションとして4日間のフレームワーク研修をやっている。通常カリキュラムには取り入れていない
- フレームワーク研修ではSpringを使っている
- フレームワークは各社でカスタマイズをしており企業によって異なるので研修では環境を決めにくい
- 実業務に近いコース設計をすることもできるか検討の価値はある。パターン化させた課題をフレームワーク前提で構築するのを初級に据えて、アルゴリズムをゼロから考えるスクラッチ開発を上級として設定する考え方もある

【テスト設計】

- テストもカリキュラムには含まれている
- テストの作法は教えるが、グループ演習に於いて丁寧にできているのは一部の進捗の早

いグループのみであることが否めない。メイキングが中心でテストは最後になるため、多くのグループでは「テスト工程の体験」になりがちで、本格的なテスト設計にまで至っていない。

- レビュー工程も含めているが、講師が実務を意識してレビューすると全く通らないレベルである
- コーディング規約を含めた演習課題を提示している。また、研修カリキュラムの中にソースコードの可読性も含んでいる。

【今後の課題】

- パートナー講師(外部講師)の受講者評価の均一化が課題。ビューポイントは提示しているが、運用時に違ってしまう

株式会社シンクスバンク

1. ヒアリング日時

令和3年12月10日(金) 14:30 ~ 15:30

2. 訪問者

富山情報ビジネス専門学校 教務部 学科長 山田 太

富山情報ビジネス専門学校 情報ビジネス学科 清水 大樹

専門学校中央情報大学校 事業開発室 プロジェクトリーダー 小澤 慎太郎

日本電子専門学校 エンジニア教育部長 ケータイ・アプリケーション科 科長 大川 晃一

3. 対応者

株式会社シンクスバンク 教育事業部教育研修課 課長 久保 雅彦

株式会社シンクスバンク 教育事業部スクール課 プログラムセクションリーダー 杉山 大樹

4. ヒアリング場所

株式会社シンクスバンク KEN スクール新宿校

5. 企業情報

(1) 基本情報

企業名	株式会社シンクスバンク
所在地	〒151-0053 東京都渋谷区代々木 2-9-2 久保ビル 4F
Web サイト	https://www.thinkethbank.co.jp/about/outline/
設立	平成 24 年 7 月
社員数	114 名(契約社員含む) ※2021年12月時点
企業理念	教育により人材の継続的な向上をはかり、 日本の発展に寄与する
事業内容	IT スクール運営、企業研修、システム開発

(2) 沿革

1988年(昭和63年)10月	KEN スクール 開校
1998年(平成10年)12月	システム開発部門スタート
2007年(平成19年)12月	技術研修センター開設
2008年(平成20年)1月	受託開発センター開設
2010年(平成22年)2月	KEN スクール 梅田校・京都校開校

2012年(平成24年)7月	株式会社ケンソフトから株式会社シンクスバンクにパソコンスクール事業、法人教育事業の2つを事業譲渡。
2012年(平成24年)7月	株式会社シンクスバンクとしてITスクール事業を継続。継続校舎は「新宿本校」「梅田校」「池袋校」「銀座校」「京都校」「北千住校」「横浜校」の全7校舎。
2012年(平成24年)10月	セキュリティ技術者の育成の促進として、株式会社サイバーディフェンス研究所との提携及び基本合意を発表。
2014年(平成26年)4月	株式会社アウトソーシング(東京証券取引所 市場第一部指定)のグループ企業となる
2016年(平成28年)1月	KENスクール 名古屋校開校
2016年(平成28年)10月	KENスクール 福岡校開
2016年(平成28年)10月	KENスクール 札幌校開校
2018年(平成30年)7月	宇佐美順也社長就任
2020年(平成28年)	株式会社KENテクノロジー設立
2021年(令和3年)	KENスクール online 開校





6. ヒアリング要約



【事業概要】

同社は、東証一部上場である株式会社アウトソーシングのグループ(以下、OS グループ)のうち、国内技術系アウトソーシング事業に属する教育・研修事業者であり、全国8拠点に展開されている「KEN スクール」を中心に、個人および法人に向けて IT エンジニア・デジ

タル人材などの技術者育成に向けた教育・研修サービスを展開している

- 「KEN スクール」の受講スタイルは、同社校舎で受講する個別指導の「通学型」、オンラインでの個別指導の「eラーニング型」、KENスクールの研修室で行う「オープン型(集合型)」、法人向けのカスタマイズ研修がメインとなる「講師派遣型」などがあり、利用者やクライアント企業の要望に対応すべく多様な受講スタイルを提供している
- 同社は、OS グループで展開する国内技術者派遣業を通じて、各産業の最新技術やトレンド情報をいち早く収集し、30年以上に亘り培ったノウハウを活かすことで、他社にはない研修プログラム・教育カリキュラムを策定・サービス化している
- OS グループが掲げる「労働格差をなくし、生き甲斐が持てる職場を創出することで、世界の人々の人生を豊かにする。」の経営理念のもと、同社は就業機会と質の高い教育を提供することを目指している
- 直近では、コロナ禍で業績不振や経営難によって深刻な雇用問題が発生した産業の方々を対象に、当社の教育・研修サービスを提供することで、社会や市場から求められているIT・デジタルスキルの習得・リスキルを支援することを発表した

【対面とオンラインの学習効果の違い】

- プログラミング研修ではコードレビューを行い、通学型の場合は直接指導できた細かなコードのミスに関してフォローを行っている
- コードレビューについてはブレイクアウトルームを使って画面共有しながら行うこともある
- プログラミング課題の提出方法はソースコードをドライブに提出
- レビューのみの場合もあるが、コンパイルして動作確認する場合もある
- 現在、オンライン授業は欠席者のために録画した授業を提供していないが、今後は提供することも検討している

【今後のオンラインの活用について】

- グループ企業(グループ規模は2万人)のうち年間約12,000人が受講するので、通学型とオンラインを併用して提供している
- オンライン授業では、自由に質問の時間を設けたり、確認テストを必ず実施したりしている
- オンライン授業のみで提供するコース(講座)も開発中である

【オンラインの問題点】

- 事前に推奨環境を満たしているか確認してもらっているが、受講者のインターネット回線の不具合でつながりにくかったり、音声が届かなくなったりすることがある

【フレームワークの学習について】

- Java の開発では様々なフレームワークが使われるが、KEN スクールでは Spring

Framework を学習する講座を提供している

- Python の Web フレームワークである Django を学習する講座を開発中である

【コーディング規約について】

- システム開発演習を行う講座では、コーディング規約の目的や必要性について教えている

【テスト設計について】

- システム開発演習を行う講座では、コーディング規約の目的や必要性について教えている
- ウォーターフォールモデルをもとに開発演習を行う講座では、コーディングした後、テスト仕様書を使ってテストを実施している

【オブジェクト指向について】

- 取引先企業からオブジェクト指向の概念を理解させて欲しいというリクエストを頂くことはある
- システム開発演習ではオブジェクト指向の理解度が高い受講者が良い成果物を作成できている
- 良い設計を行う為にはオブジェクト指向の学習が必要である
- まずはオブジェクト指向を意識せず、自由にコーディングしてプログラムが動くことの面白みを体験してから、次にオブジェクト指向をもとに適切な設計ができるように教育している

株式会社エクステンジ ソリューションズ

1. ヒアリング日時

令和3年12月10日(金) 17:00 ~ 18:00

2. 訪問者

富山情報ビジネス専門学校 教務部 学科長 山田 太

富山情報ビジネス専門学校 情報ビジネス学科 清水 大樹

専門学校中央情報大学校 事業開発室 プロジェクトリーダー 小澤 慎太郎

日本電子専門学校 エンジニア教育部長 ケータイ・アプリケーション科 科長 大川 晃一

3. 対応者

株式会社エクステンジ ソリューションズ 取締役 稲葉 太郎

4. ヒアリング場所

株式会社エクステンジ ソリューションズ

5. 企業情報

基本情報

企業名	株式会社エクステンジ ソリューションズ
所在地	〒108-0073 東京都港区三田 3-9-7 三田三好ビル 4階
Web サイト	https://s.exchange.jp/
設立	2011年9月21日
従業員数	32名(2021年12月現在)
事業内容	Web アプリケーション開発、保守 ネイティブアプリケーション開発 システムインフラの構築、保守



6. ヒアリング要約



【事業概要】

- エクスチェンジグループ全体では140名

- Web アプリケーション開発・ネイティブアプリケーション開発のエクステンジソリューションズと、デザイン会社エクステンジクリエイティブがある
- 福岡、沖縄に事業拠点がある

【リモート環境の活用状況】

- フルリモートで業務を行ったところ、新卒新入社員の帰属意識の問題や、本当に社会の役に立っているかと言った不安が発生した
- 新人は12月から週3日出社してもいいということにしたところ、みんな会社に来たがる
- 飲み連れて行ってほしいという要望もある

【力を入れているプログラミング言語】

- PHP、Java、Java script、C#・・・お客さんからの要望が多い
- PHP、Java は特に多い
- 会社では PHP、Java を教えている
- 新入社員は3か月間リアルで研修を受ける
- 受託開発が業務の中心なので、1年かけて1人前になってほしい
- 1人前は、1人月の仕事を1人月でできるようになること
- 社会経験を積んでいる人のほうが習得は早く、3か月くらいで1人前になることもある

【コーディング規約について】

- PHP は PSR-12 (PHP の一般的なコーディング規約) に沿ってやっていくように指導している
- 新人研修は、2～3日のビジネスマナー研修のあとは外部研修機関で約3か月のリアル研修を受けるが、そこで PHP もやる

【リモート環境下での開発経験の評価について】

- 積極的なプラス評価にはならない
- 今後はフル出社もないがフルリモートもないので、その経験が判断ポイントになることはない

【ペアプログラミング経験の評価について】

- 評価している。一人前になるためには通らなければならないと考えている
- 人に見られているところでコードを書くとか、人がコードを書くところを見られるのは勉強になるし、一人前になるまでは隣に座らせて教えていくのが効果的
- 教育手法の一つとしてペアプログラミングの有効性を認めている

【採用について】

- 採用については、プログラマ適性検査でふるいにかける
- 性格やポテンシャルは面接で判断する
- 情報系学科の卒業生は、その時点でシステム・PC と飽きずに向き合うことができているということなので、最低限の適性をクリアしていると考えている
- 資格取得に対する評価は、全く同じ条件だったとしても、資格以外の部分で判断する
- ポテンシャルの判断は感覚値。失敗例・成功例があって経験則で採用している
- プログラミング経験のない人も多少いるが、経験者の方が多い
- 6～7割はプログラミング経験者

【新入社員教育後のキャリアパス】

- 3か月の外部研修後、OJT に入る(戦力に入れない OJT)
- その後、データベース、ネットワーク、Linux などに進んでいく
- 受託開発中心で、新規開発、改修、運用とあるが、外部研修は全員同じ研修を受ける
- (OJT では)先輩がおこなれているところ・謝っているところをもっと見せたい
- 離職率は低いので採用は成功していると考えている
- 1人月の仕事が1人月でできるのが一人前の指標だが、その中身は上流工程ができるようになること。インフラやネットワークなどどこを目指すのかは本人の意欲次第
- プロジェクトマネジャーやプロジェクトリーダーを目指すところ
- SE は外部から調達できる。(SE や)プログラマは過程でしかない
- 受託開発中心なので開発手法はウォーターフォール型でやっている。アジャイル開発は契約形態が異なる(大きなプロジェクトだとウォーターフォール型を進めていくのが顧客も安心する)

【プログラマが1か月でコーディングする規模】

- 測れない
- コード量だけでなく、要件定義の不備や不足を埋める仕事もある

【プログラマの能力の評価基準】

- 組織人としてどう上に上がっていくかの基準はある
- テクニカルスキルとヒューマンスキルを上司による他者評価で人事考課している
- 社内の評価基準に資格取得は入っていない
- 引退するまでプログラマと言うキャリアパスは設定していない。SE やプロジェクトマネジャーを目指すことになる
- チームが基本単位なので、プログラムだけ出来てもダメ

【専門学校教育でもっとやってほしいこと】

- ネットワークの知識をもっと増やしてほしい
- データベースの知識を強化してほしい
- 頭が柔らかいうちにいろいろな知識を吸収しておいてほしい
- フレームワークは入社してからやればいいことだが、イメージは持っておいてほしい
- 遊んでほしい=人間力を高めてほしい

【資格に対する評価】

- 基本情報技術者試験は知識を吸収したという点で評価はしており、どこかでつながっていくと思っている
- 資格取得によって手当や一時金はないが、取得にかかる経費は会社が支援している

【テスト設計指針】

- 社内にテストガイドライン(目的・内容・視点)がある
- テストコードを埋め込むなどしっかりとやると期間とコストが見合わないので、問題は起こるという前提でテスト方針を決める

【情報系専門学校卒業生に求めること】

- 最低限、エンジニアが知っている言葉を知っている
- プログラムは書けなくてもいい
- 即戦力とは考えていない。やったことがないから、即戦力にはなり得ない
- 知識として、必要最小限以上のことを知っているとおっ！と思う
- 社会人として一般的なことを知っていてほしい
- 本をどれだけ読んできたか
- アルバイトなどの社会経験
- ものを知っているということ・・・いつか使い物になる

パシフィックソフトウェア開発株式会社

1. ヒアリング日時

令和3年12月13日(月) 14:00 ~ 15:00

2. 訪問者

富山情報ビジネス専門学校 教務部 学科長 山田 太

高知情報ビジネス&フード専門学校 副校長 中川 隆

専門学校中央情報大学校 事業開発室 プロジェクトリーダー 小澤 慎太郎

日本電子専門学校 エンジニア教育部長 ケータイ・アプリケーション科 科長 大川 晃一

3. 対応者

パシフィックソフトウェア開発株式会社 技術本部 第2技術部開発1課 副課長 吉本 堅悟

パシフィックソフトウェア開発株式会社 管理部係長 リクルート担当 福永 龍雄

4. ヒアリング場所

高知情報ビジネス&フード専門学校 教室より zoom 会議にてヒアリング

5. 企業情報

(1) 基本情報

企業名	パシフィックソフトウェア開発株式会社
所在地	〒780-0945 高知市本宮町 105-22
Web サイト	https://pacificsoftware.co.jp/about/
設立	昭和47年9月22日
社員数	77名(2021年4月現在)
事業内容	ソフトウェア開発(物流、電子決済、レーダー)／ 自社製品(AI 自動給餌システム「餌ロボ」、海洋土木高度情報技術サービス「Sea Vision」)の開発、販売／ システム開発(IoT・AI・医療)

(2) 沿革

1972年(昭和47年)9月	資本金 50 万円で高知市にパシフィックソフトウェア開発(株)を設立、ミニコン、マイコンのシステムサポート業務を開始する
1973年(昭和48年)4月	マイクロコンピュータの販売を開始する
1979年(昭和54年)7月	大阪営業所(現 大阪開発センター)を大阪市北区に開設

1983年(昭和58年)1月	組織改組 本社・システム技術センター[高知] システム開発センター[大阪] 同時にシステム開発センターを大阪市西区に移転
1990年(平成2年)8月	マルチウインドウ・テキストエディタ「ZED」を販売開始
1990年(平成2年)12月	資本金2,000万円にて子会社(株)プラネットを設立
1991年(平成3年)4月	子会社(株)プラネットが本格的に営業を開始する
1995年(平成7年)8月	中小企業創造活動促進法の高知県初認定企業となる
1997年(平成9年)3月	超音波測深装置「SeaVision 500」を販売開始
1998年(平成10年)10月	データ駆動型メディアプロセッサに関する研究・開発に着手
1999年(平成11年)1月	PSDグループ新社屋をフロンティアこうち団地内に新築・移転
1999年(平成11年)9月	林業用、素材生産評価シミュレーション「LogHouse」を販売開始
2002年(平成14年)1月	新型超音波測深装置「SeaVision 501」を販売開始
2006年(平成18年)4月	子会社(株)プラネットを吸収合併
2006年(平成18年)10月	釣場情報サイト「fishon 四国」オープン
2007年(平成19年)12月	(財)社会経済生産性本部 サービス産業生産性協議会より「ハイ・サービス日本300選」第1回受賞企業に選定される
2008年(平成20年)10月	釣場情報サイト「フィッシュオン四国」がドコモの公式サイトとなる
2012年(平成24年)10月	浚渫施工管理システム「SeaVision Navigator」を販売開始
2016年(平成28年)6月	新型超音波測深装置「SeaVision SV-1000」を受注開始
2017年(平成29年)6月	機器情報モニター「Virtual Bridge Monitor」を販売開始
2019年(平成31年)4月	海面養殖業向け自動給餌システム「餌ロボ」を受注開始
2019年(令和元年)5月	高知本社増築
2021年(令和3年)3月	釣場情報サイト「フィッシュオン四国」サービス終了

6. ヒアリング要約



【学習に力を入れているプログラミング言語】

- それぞれのシステム開発案件で適切な言語を選んでいる
- C 言語系(C 言語、C+、C#)、Java 系(Java、Java script、Android Java)が多い
- 制御系は C 言語が多く、Web、クラウド系は Java が多いが、お客さんによって分かれる
- 開発エンジニアは大まかには分かっているが、一人で複数言語を扱うエンジニアもいる

【入社時のプログラミング教育について】

- 入社時に習得させる言語は C 言語か Java で年によって異なるが、最近は C# から入ることが多い
- 1 か月間の新入社員教育の後に配属先での OJT を行う
- 教育は設計や開発の進め方重視で、アルゴリズムや文法をやるカリキュラムとは分けて、1 つの課題を出して設計からテストまで一通り制作するプログラムがある。社内の技術者が講師を務める
- 一人で任せられるまでになるのは概ね3年かかる。一人前とは、仕様書を渡したらプログラミング設計込みで自分で出来るようになるレベルを想定している
- 最初のうちは配属先での先輩の指導のもと、小さな部分を任せ、密にレビューを行う
- 入社時に求めるのは、設計書を渡したら一人でプログラムが組み上げられるレベル(プログラムに近いレベルの設計書)
- フローチャートをコードに置き換えるところから始める

【ノーコード/ローコード開発について】

- 実務ではノーコード開発/ローコード開発も取り入れている
- 新入社員にはアルゴリズムを考えられるようになることを求めている
- 1から作れるようになってから、ローコードで作れるようにする

【フレームワークの活用について】

- フレームワークは使っている
- 実際の開発はフレームワークでやることになるが、中身がわからないと使えないので、1年間のカリキュラムで入社1～3年の若手社員中心に勉強会をやっている

【入社後のキャリアパスについて】

- 入社時はプログラマからスタートで、システムエンジニア、プロジェクトリーダー、プロジェクトマネジャーに進んでいく
- 管理面で特化していく道と、技術面で特化していく道の2つに分かれる
- その人の特性に合わせて選択することができる

【社内のレベル基準】

- 相対的な評価に関しては、評価表があり、社内評価している
- 資格は評価の対象になる。情報処理技術者試験は推奨している
- その後は業務に合わせて専門的な資格を取っていくことになる
- 学生が基本情報技術者を持っていれば評価の対象となる。資格で出題される内容を理解しているという判断材料にしている
- 資格で勉強する内容は、社会に出て実践力を高める基礎となる

【入社時の技術の評価】

- 学生がどんな技術を持っているかは、あまり重視していない。人柄の方を重視している
- 適性試験で論理的思考ができるかどうかを見る
- 課題に対してどういう視点で作り、同課題解決したのか、そのプロセス・考え方は評価の対象になる。技術力は入社してから何とかなるが、視点や課題解決力はなかなか身につかない

【専門学校卒業生に求めること】

- 資格取得
- 問題解決の考え方を重点的にやってほしい
- 技術は企業に入ってからでも間に合うので、下地の部分をしっかりやってほしい
- プログラム作成は将来的には自動化されるので、考え方の部分を身に付けてほしい
- 「こんなものが欲しい」と言われたときに、0から自分で考える力を求めている

- 採用したいのは、学習に対する姿勢ができている学生。時代とともに技術は進化し、毎年新しいことを学ぶ必要が出てくる。その際に、根本となる考え方を理解しているかどうかは重要

【社内にコーディング規約はあるか？】

- コーディング規約はある
- 顧客からコーディング規約を指定してくる場合もある

【テスト設計の指針】

- テスト設計の指針を用意しているわけではないが、新入社員教育や現場のOJTでテストの考え方を教えている(明文化できていないわけではない)
- テストの時間は足りないので、手作業を減らしてテストを自動化できるよう心掛けている

【コミュニケーションツール】

- コミュニケーションには GitHub や Slack を使っている部署もある
- いろいろなツールを使えることはアドバンテージになる

【採用時の基準について】

- 高校生、専門学校生、大学生は平等に見ている
- 文系でも論理的思考ができる人はいるので、適性試験で判断している
- ソフトウェア開発を仕事としてやっていく覚悟があるかどうかは重要視しているが、大学生が一就職先企業として入社してくるイメージなのに対し、専門学校生は興味があってやってきている人が多い。高校生の中にもプログラミングに興味があり、ハングリリーさもある人もいる。
- 10 日間のインターンシップでは、実際の仕事と同じように小さなソフトウェアを設計からテストまでやってもらう。成功体験をしてほしい

【エンジニアの成果やノルマについて】

- ノルマはないが、チーム全体で納期を達成し納期までにものを仕上げることは求めている
- 生産性の指標としては、1日600ステップ、C 言語だと1日200ステップ程度
- ステップ数よりも、商品をどれだけ考えられるかが重要

【リモート環境下での開発について】

- リモート環境下での開発は、コロナで一時的にはやったが、今はほとんどやっていない
- 顧客データ等の持ち出し禁止もあるが、コーディングするだけならベテランになら任せても良い

【ペアプログラミング経験について】

- ペアプログラミングは教育目的でやっている
- インターンシップもペアでやっている
- ペア同士のアドバイスやレビューをどうやって進めているかを見ている

株式会社ソフテック

1. ヒアリング日時

令和3年11月12日(金) 16:00 ~ 17:00

2. 訪問者

富山情報ビジネス専門学校 教務部 学科長 山田 太

高知情報ビジネス&フード専門学校 副校長 中川 隆

専門学校中央情報大学校 事業開発室 プロジェクトリーダー 小澤 慎太郎

日本電子専門学校 エンジニア教育部長 ケータイ・アプリケーション科 科長 大川 晃一

3. 対応者

株式会社ソフテック 執行役員 ソリューション開発事業部 事業部長 樋口 章広

株式会社ソフテック ソリューション開発部ソリューション開発1課 課長 竹村 崇史

株式会社ソフテック ソリューション開発部ソリューション開発2課 課長 増田 勝也

4. ヒアリング場所

株式会社ソフテック

5. 企業情報

(1) 基本情報

企業名	株式会社ソフテック
所在地	〒781-8001 高知県南国市蛸が丘1丁目4番
Web サイト	https://www.softec-inc.co.jp/company/
設立	昭和48年(1973年)10月
従業員数	246名(正社員210名、嘱託・契約社員6名、パート・アルバイト30名)
企業理念	■ソフテックは世の情報化に関わり、全員参加による新しい価値ある商品、サービスを創造、提供し、世の人々から信頼され誇りある企業集団になることを目指します。 ■ソフテックは当社にかかわる全ての人々を大切に、人類共通のかけがえのない財産である地球環境を守るために、積極的に関わります。
事業内容	システム機器販売、システム受託開発、ソフト商品開発、システム要員派遣、情報処理サービス、アウトソーシング受託、IDC(インターネットデータセンター)運用

(2)沿革

1973年(昭和48年)10月	高知市棧橋に株式会社高知ソフテックを設立
1973年(昭和48年)12月	FACOM230-25による計算センター業を開始
1974年(昭和49年)9月	FCA(富士通系情報処理サービス業グループ)入会
1976年(昭和51年)4月	オフィスコンピュータの販売を開始 金融機関への要員派遣とソフト受託開発を手掛ける
1980年(昭和55年)7月	FACOM M-140によるオンラインサービスを開始
1985年(昭和60年)8月	高知市土居町に社屋を移転
1985年(昭和60年)10月	東京都に事業所を開設
1987年(昭和62年)4月	株式会社クイック情報サービスを設立
1991年(平成3年)1月	香川県高松市に事業所を開設
1993年(平成5年)10月	社名を株式会社ソフテックに変更
1993年(平成5年)11月	電子ブック検索ソフト「Book Finder」発売開始
1994年(平成6年)4月	自動車部品販売会社向けアウトソーシングシステム稼働 業務ソフト「販売部長」発売開始
1995年(平成7年)12月	画像管理ソフト「Visual Shot」発売開始
1996年(平成8年)10月	最新鋭大型汎用機 GS8200 に入れ替える
1997年(平成9年)4月	「Visual Shot」が第11回高知県地場産業大賞を受賞 業務ソフト「経理部長」発売開始
1997年(平成9年)5月	自動車販売会社向けアウトソーシングシステム稼働
1998年(平成10年)4月	業務ソフト「給与部長」発売開始
1998年(平成10年)11月	スーパーマーケット向けアウトソーシングシステム稼働
1999年(平成11年)4月	パッケージソフト「Visual Shot」を米国にて発売開始
1999年(平成11年)12月	大型汎用機 GS8300 を追加導入
2000年(平成12年)6月	県内流通業界向けVANサービスの運用開始
2000年(平成12年)12月	品質マネジメントシステム「ISO9001」を認証取得
2001年(平成13年)6月	小売量販店向け物流EDIシステムの運用開始
2001年(平成13年)7月	医療業向けオーダーリングシステム初納入
2002年(平成14年)6月	医療業向け電子カルテシステム初納入
2002年(平成14年)7月	画像CDアルバム作成ソフト「デジカメからCDアルバム」 発売開始
2004年(平成16年)1月	高知富士通テクノポートに南国オフィスを開設 汎用機からオープンシステムへ移行
2004年(平成16年)7月	個人情報保護マネジメントシステム「プライバシーマーク」 を取得

2004年(平成16年)11月	環境活動評価プログラム「エコアクション21」を取得
2006年(平成18年)6月	情報セキュリティマネジメントシステム「ISMS」(現ISO/IEC 27001)を取得
2008年(平成20年)7月	職場風土改革促進事業への取組み開始
2008年(平成20年)10月	「高知県次世代育成支援企業」に認証される
2009年(平成21年)6月	緊急地震速報システム「KURUZEYO」サービス開始
2009年(平成21年)9月	かんたんグループウェア「MyWeb Portal Office」発売開始
2010年(平成22年)8月	FAXOCR eエントリーセンター高知を開設
2011年(平成23年)9月	全国小売量販店向けMDシステム「Trinitailer」(現「RetailFit」)発売開始
2015年(平成27年)4月	愛媛県松山市に事業所を開設
2017年(平成29年)10月	グループ企業「株式会社クイック情報サービス」吸収合併



6. ヒアリング要約



【習得に力を入れているプログラミング言語】

- Java・・・立ち上げ当初から Java でやっていた
- C#・・・Java 親和性が高いことから採用している
- マルチプラットフォーム、マルチデバイス、ブラウザブラウザであることが採用理由
- お客様によっては低予算でAccessを使うケースもある
- プラットフォームはアジュール、AWSで開発している
- JavaとC#の比率は8:2～7:3くらいで、圧倒的にJavaが多い
- 高知本社では今後 C#が増えていく可能性はある。東京支社では当分は Java を使う

【採用について】

- 新入社員は6割が非情報系学科卒業生、4割が情報系学科に結果的になっている
- 情報系学科だけではなく、我々の視野を広げるため、垣根を省いています。情報系/非情報系をフラット(公正)にしたうえで、学力・能力を評価している
- 採用試験では学力と能力を見ている
- 学力は情報系の知識であり、能力は人となりやしっかりと主張すること、求めた質問に対する的確に答えられること
- 能力の比較で専門学校生が落ちることがある。4年制大学卒、高専卒と比較して能力面で専門学校生が落ちることがある
- 学力で光るものがあれば能力が低くても採用することはある。スペシャリストになる可能

性がある。成功事例もあるが結果的に会社を辞めた

- 尖ったものがあつたら採用する
- 採用試験(適性試験)で論理的思考などを測っているが、コロナの問題で対面の筆記試験ができなくなり、過去に蓄積したデータと比較することができなくなった
- 全員本社採用で 5~6 名を採用し、東京に 2~3 人配属する
- 東京採用がいまぐいかなかったので本社採用としている

【新入社員研修について】

- 新入社員には半年間の技術研修と半年間のOJTを行う
- 4月は座学で、富士通の教育会社から講師を招いて行う。インフラの解体などもやる
- その後3か月はJavaを、どうやって動くのかの原理原則から入って、インフラ、ネットワーク、プロジェクト開発とやっていく
- 将来的にJavaはC#になる可能性もある
- 高知県情報産業協会の研修ではC#を選択する企業が増えており、Javaとどちらを選ぶか悩んでいるが、仕事に直接つながるのはJavaなので会社としてはJavaを選びたい
- 半年間の技術研修を5か月間に短縮して少し早くOJTに入る方向である
- OJTでは実際のプログラムを組む。新入社員1名に対して先輩1名がつき、触れる範囲を最小限に限定して、新入社員に責任は取らせない
- 半年間の研修終了後、東京・高松も配属される

【資格取得について】

- 資格そのものが採用に有利になることはない。基本情報技術者ではまずない。応用情報技術者を持っていたらスペシャリストの可能性もあるため少し揺らぐ
- どちらかと言えばベンダー系資格(Java、Cisco)の方が光るものがあると判断している
- 入社後も国家資格取得は推奨しており、同業他社と比べて高額な報奨金を出している
- 採用時点ではあまり重視しないが、入社後のモノサシとしては有用だと考えており、特に東京支社ではその傾向が強い

【学校でのプログラム経験について】

- 業務に近い開発を学校でやっていたら経験としては評価になる
- 中途採用の時には Paiza で学習していることが評価につながる
- ネットワーク系で TCP/IP の基礎などの試験があつたらよい

【フレームワークを使った開発】

- C#は ASP.net、MVC のフレームワークを使っている
- Java はプロジェクトごとにお客さんに合わせたフレームワークを選んでいる

- 総合開発ツールとしてエクリップスを使っている
- 専門学校生がフレームワークを使った経験があれば、それは評価につながる
- コミュニケーションツールも含めて学生時代に経験があれば評価につながる

【テスト基準・品質基準について】

- 社内にテスト基準・品質基準はある
- 品質基準に関しては高い基準を持っている
- 品質保証ができる100%の完成品を納品する
- アジャイル開発的なこともやっているが、事故が起こっても業務が停止しないようにしている。ワークアラウンドで回避策は担保している
- 事故発生時には原因の追究はもちろん行うが、ワークアラウンドとして業務再開し、根本原因を特定して再発防止策を講じる
- 富士通でもアジャイル開発部隊があり、お客様との契約を見直し成功している事例が多数ある
- 品質基準を設けて品質向上に向けた取り組みをしているが、異常系を定する能力・考える能力がないと品質は上がっていかない

【コーディング規約】

- 全社的なコーディング規約はないがプロジェクトごとにある
- 属するプロジェクトによってコーディング規約が異なる

【プログラマの評価基準】

- キャリアパスは、プログラマ、システムエンジニア、プロジェクトリーダー、プロジェクトマネージャー等がある。また管理職という道だけでなく、スペシャリストとしてやっていく道もある
- 評価基準は相対評価となり、あまり数値化されていないが、できる人のところには間違いなく仕事が集まる傾向にある
- 営業向きの人材を採用しても3年間でプログラムをさせる
- インフラ系で採用してもプログラムを学ばせる。キャリアチェンジの際に幅が広がるため

【リモート勤務について】

- コロナが落ち着いてからは本社ではリモート勤務は1割未満(本人の選択制)。高松支社ではゼロ。東京支社では9割がリモートワーク(週1回はリアルで出勤する日がある)
- 人によってはリモートだと非効率になるケースがある
- コミュニケーションの取りやすさなどで出勤を選ぶ傾向にある
- 環境としては全員がリモートワークにできるセキュリティを考慮した環境を用意している

- リモート勤務を導入してから、東京支社に送り込んだ人材で退職の相談が少なくなった

【専門学校への質問】

- デジタル教育が普及しているが、脅威か？
→小中学校でプログラミングを教えていることが脅威である

株式会社ソフト

1. ヒアリング日時

令和3年12月17日(金) 10:00 ~ 11:00

2. 訪問者

富山情報ビジネス専門学校 情報システム学科 清水 大樹

富山情報ビジネス専門学校 情報システム学科 ファティン アミラ

3. 対応者

株式会社ソフト 取締役 総務部 部長 助田 篤郎

株式会社ソフト 総務部 部次長 吉田 いづみ

4. ヒアリング場所

株式会社ソフト

5. 企業情報

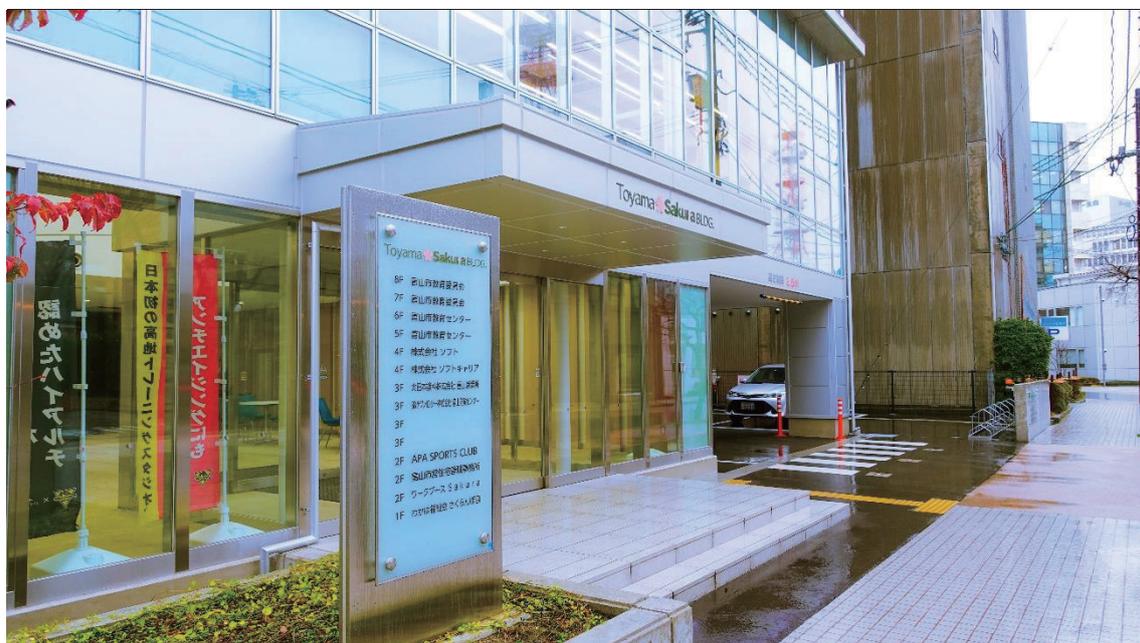
(1) 基本情報

企業名	株式会社ソフト
所在地	〒930-0005 富山県富山市新桜町6番15号 Toyama Sakuraビル4階
Web サイト	https://www.soft-co.jp/html/company/outline.html
設立	昭和58年6月1日
社員数	90名(令和元年5月現在)
経営理念	株式会社 Soft は幅広いIoTサービスを提供し、地域社会とお客様とともに、進捗向上にお役に立つ企業を目指します。 Solution Office Factory Technology ・意欲と熱意と責任感を持ち、未知の分野に果敢に挑戦する人財育成に努めます。 ・お客様の発展にお役に立つ想像力/技術力向上に努めます。 ・企業倫理を順守し高い志を持って企業活動に努め、適正な利益を確保し、株主の期待にお応えします。 ・社員が誇りと自信と喜びを感じる社風を築き、会社の持続的発展に努めます。
事業内容	コンピュータシステムの設計及びシステム開発 コンピュータシステムに関するコンサルティング

	コンピュータシステム関連機器及びパッケージ販売 人材派遣
--	---------------------------------

(2)沿革

1983年(昭和58年)6月	資本金500万円にて(株)ソフト創立
1986年(昭和61年)6月	東京支社開設
1987年(昭和62年)6月	資本金1,400万円に増資
1995年(平成7年)8月	資本金1,900万円に増資
2003年(平成15年)8月	資本金4,000万円に増資
2004年(平成16年)4月	本社移転
2006年(平成18年)6月	キャリアセンター(人材派遣事業)設立
2006年(平成18年)9月	「プライバシーマーク使用許諾事業者」認定取得
2007年(平成19年)9月	経済産業功労「職能力開発に対する知事表彰」受賞
2007年(平成19年)12月	東京支社を現住所に移転
2010年(平成22年)2月	名古屋支店開設
2014年(平成26年)6月	名古屋支店を現住所に移転
2021年(令和3年)5月	本社を現住所に移転



6, ヒアリング要約



【習得に力を入れているプログラミング言語】

- Java・・・メイン
- C#・・・在庫管理や会計管理を含めた業務システムでは、ニーズがネットワーク系に変わってきているので、VB.net に代わり、C#の顧客からのニーズが高まってきている
- パッケージソフトを作るのに C#に転換した。システム的な拡張性を考えると VB.net よりも C#のほうが優位

【新入社員研修について】

- 新入社員研修は集合研修に新入社員 5～7 名を出している
- 期間は4月から6月の3か月間で、名刺の受け渡しなどビジネスマナーを含め、IT の基礎およびプログラミング言語を習得する
- Java か C#は選択制になっているが、入社前の研修等で適性を見極めてどちらを選択するかを決めている
- 適性は、最終的には人同士で話さないとわからないが、業務系または制御系の顧客ニーズがあり、業務系の場合はJava、機械制御が入る場合はC#になる

【ペアプログラミングの経験の評価】

- 当社ではナビゲーターとドライバー(先輩+後輩)で、ペアで動くという形にはなっていないので、ペアプログラミングはやっていない

- 上級 SE が責任を持ってチームを見ていく形を取っている

【リモート業務について】

- コロナの影響でやむを得ずテレワークを導入したが、その評価にまでは至っていない
- 今でもテレワークをやっている少数のメンバーもいるが、大半は元に戻った
- やむを得ず一度はテレワークに移行したが、コミュニケーションの取り方等で課題はあるので、今後は必要に応じてやっていく
- 結果としてテレワーク対応ができることは必要な要件にはなってきている
- テレワークの際のコミュニケーションツールは、Microsoft Teams を使っている
- 顧客とのミーティングは相手からの要望によって zoom を使うこともある

【ノーコード開発の評価】

- ノーコード開発は一人で最初から最後までできてしまうというメリットがある
- 問題点もあるが、使い方がわかっていると早いので、やったことがあるという経験は実績・経験としては有利だと思う
- 業務的に、顧客要求を理解して既存のものをあてはめるとするのは、1ステップにはなる
- 頻発はしていないが、顧客からの要望があればノーコード開発もやる
- 採用時にノーコード開発の経験はそれほど評価にはならない
- 例えば製造系の会社が自前でシステムを作る場合などにノーコード開発は有利だと考えている
- (開発元企業が)10年後にどうなるかわからない

【プログラマの能力評価基準】

- プログラマとしてのスキルは計測していない。相対評価で人的評価がメインになる
- 経験のあることばかりをやっていれば生産性は上がるし、経験のない難しい業務を担当していれば生産性は落ちる。それだけでは測れない
- 言語やアルゴリズム以外に必要な能力は、設計者と意図を合わせるためのコミュニケーション能力、意志や感情を伝える能力
- 上長としては情報を吸い上げる能力も求められる(ヒアリング能力)
- チームでやるときは、「言ったつもり」「聞いたつもり」をなくし、協調性を発揮することがとても大事

【採用時のスキル評価】

- 今年採用した7人のうち3人はプログラミング経験が浅い。そのうち1人は文系出身者
- 技術的な経験は評価しているが、結果的に経験者と未経験者は半々くらいになっている
- 主に県内の大学生、専門学校生、能力開発大学校から採用している

- 研修内容は経験者と未経験者で変えてはいない
- 10月から内定者研修として必要最低限のことを理解する IT 基礎の e-ラーニングを開始する、経験・未経験・文理関係なく同じ内容をやる。E-ラーニングのベースになるのは基本情報技術者

【資格の評価】

- 基本情報技術者などの資格取得に対しては一時金(報奨金)を出している
- 昇進のための必要条件として基本情報技術者試験は課している
- エンジニアとして一定ラインを超える時に基本情報技術者レベルのことは求める。概ね 7～8年目にそのライン(主任クラスへの昇格)が来るが、少しずつ早くなっている
- 入社時に基本情報技術者を持っているのは一つのアドバンテージになる

【専門学校卒業生に求めること】

- 「好きでやっているか？」ということは常に問うてほしい
- ITの世界はおもしろいと思ってくれるように指導してほしい
- 「好きでやっている」という人を増やしてほしい
- 「こんなはずじゃなかった」と思ったら続かないので、好きでやっていること、面白いと思ってくれることは大切にしている
- 興味や思いがあればミスマッチは起こりにくい
- 採用方法は(時代に合わせて)変えているが、採用基準は変えていない
- 採用必要人数に満たないときはプレッシャーもあるので、採用基準が動かないこともないが、最低ラインは変えていない

【採用後に求めること】

- 集中力や結果の出し方、テストの仕方など実地のシステム開発をわかってほしい
- 開発はウォーターフォール型が中心。アジャイル開発をやっているチームもあるが、ユーザー要件が動くので事前に工数はわからないリスクがある
- テストが大変だということをわかってほしい。システムは動いたから OK ではない。地道なテスト工程があることを理解してほしい
- 強い思いを持ってプログラムを作る人になってほしい

【テスト設計の基準】

- テストの基準は、顧客が工程の中でどこに主眼を置いているかによって異なってくる
- テストの粒度はプログラムごとにより変動してくるので、テスト設計の指針は案件ごとに決めている

【コーディング規約について】

- コーディング規約はプロジェクト・顧客によって異なる
- 顧客にソースコードを提出することはまずない。あってもオブジェクト納品

株式会社ブレインファクトリー

1. ヒアリング日時

令和3年12月17日(金) 16:00 ~ 17:00

2. 訪問者

富山情報ビジネス専門学校 情報システム学科 清水 大樹

富山情報ビジネス専門学校 情報システム学科 ファティン アミラ

3. 対応者

株式会社ブレインファクトリー 代表取締役 中田 美千子

4. ヒアリング場所

株式会社ブレインファクトリー

5. 企業情報

企業名	株式会社ブレインファクトリー
所在地	〒939-2376 富山県富山市八尾町福島 194-1
Web サイト	https://www.brain-factory.jp/about.html
設立	2007年7月
社員数	25名(男性22名・女性3名)2020年6月現在
経営理念	・当社は人を尊重します。 ・当社は常に挑戦します。 ・当社は丁寧なモノづくりをします。 ・当社は社会や地域に貢献します。
事業内容	・システム開発 ・IT技術者派遣 ・システムサポート ・ITコンサルティング



6. ヒアリング要約



【習得に力を入れているプログラミング言語】

- C 言語・・・コードで理論的にプログラムを作っていくことを学んでもらう
- C#・・・ブラウザで動く Web ベースのものは実案件では C#が多い
- 最初は Excel VBA から入ってもらう

- Java もやっているが、単独言語だけのプログラムはなくなってきている。他の言語 + Java というように、今は一つの言語で完結することなく Java+PHP など組み合わせで作っていく
- Java を深く学んでいけば C 言語をやったことがなくても大丈夫だと考えている

【新入社員研修について】

- 外部研修で主に Java で製品を作り上げる 2 ヶ月間の研修に出している
- その後、社内 Off-JT でデータベースなどをやる。外部研修後、4~5 ヶ月は社内 Off-JT (後半は OJT)を行う

【ペアプログラミングの評価】

- メイン(プログラミング)・サブ(チェック)の両方ができたほうがよく、役割を交代しながらペアプログラミングをすることは評価できる
- 自分好みのプログラミングに突っ走ることがないようにペアで進めていくことには意味がある

【リモートワークについて】

- 現在は 3 分の 1 くらいが在宅で仕事をしている。本当は集まってやりたいが、会社のスペースだと密になるのでテレワークにしている
- 常駐プログラマは常駐先から在宅と言われ、会社で仕事をしている人もおり、在宅と同じくらいの割合
- 会社としてはリモートでも OK だが自分の意思で会社に来たいという人が多い
- リモート開発の評価は、プログラムを作るという面では成果物が変わらなければいいが、相手から聞き出す能力がないとコミュニケーションの面で難しいと感じている
- リモートワークには、機密書類のやり取りもあるので Microsoft Teams を使っている
- ゲストを交えてのミーティングには zoom も使っている

【ノーコード開発の評価】

- ノーコード開発は、正常パターンを作っていく上では役に立つが、システム開発においてはエラー時の対応が大事なので、解析できない点は評価できない
- 案件としてノーコードで開発した経験は当社ではない
- ノーコードとして準備されているものが、どこまでの選択肢があるのかによって評価する
- 社員には新しいものを作ると言うことを覚えて欲しい

【専門学校教育で取り入れてほしいこと】

- 障害を見つけて修正するテスト教育

- どうしたらいいのかを自分で調べて修正し、解決する力があれば即戦力になる

【コーディング規約】

- コードの書き方については、「変数はこのように書く」など社内で統一して決めている

【プログラミングスキルの評価】

- 仕事でスケジュールをいかに効率よくこなしていくかは大切だが、案件によって難易度が異なるので同じように評価することは出来ない

【テスト設計の指針】

- 入力チェック項目(桁、閾値、重複データなど)については受け継がれてきたものがある
- 明文化されてはいないが、メモ書きで「こういうパターンは閾値テストをやる」などが定着している
- 作った人以外の人がテストをするというルールになっている
- 作りながらの単体テストは作った本人がテストし、完成時のテストは別の人がやる

【プログラマが1ヶ月でコーディングする規模】

- チームでの協同作業のため、個々に出すのは難しい
- 以前はステップ数や画面枚数を基準にしていたが、今は、ステップ数での規模の算出をしない

【プログラマの能力の評価基準】

- 工数見積がある中で、見積時の工数の中で、いかに効率的に短時間で修められるか、納期を守れているかどうかを評価基準にしている
- 成果物のコードの美しさも評価している
- テストの緻密さも評価基準になる
- そのほかに個々の能力評価もある

【新入社員に求められるプログラム能力】

- プログラムの完成まで一人でできることは期待していない。プログラミング経験があれば十分
- 自分で調べる力があるとか、人に聞ける力があるかどうかは重要
- プログラミング経験のない人を多く入れている。プログラミング経験のない人のほうが多く、7割くらい、経験者は3割
- 経験者と未経験者には同じ研修を受けさせている
- プログラムを趣味で作っている人は結構いる

- 面接でもプログラムを触り慣れている雰囲気伝わってくる

【資格取得の評価について】

- 情報処理技術者試験(特にセキュリティ関係)のほか、マイクロソフトの MOUS、オラクル(データベース)、簿記などを評価している
- 簿記などの会計系は業務システムには絶対に役に立つ
- 資格手当がつくのは情報処理技術者試験(IT パスポート以外)、オラクルデータベース、簿記、(MOUS は資格手当の対象になっていない)
- 今後は AWS などクラウド系の資格を取らせたい
- 案件としては、クラウドは AWS よりも Microsoft 系が多い。アジュールを使った案件が入ってきている
- 資格の評価と開発実績の評価は別

【言語やアルゴリズム以外でプログラマ・SEに必要な能力】

- 自分で調べる能力、どうやって調べるか?周りの人に聞くなどが出来る能力
- 不具合からバグを見つけられる能力
- 提案能力を伸ばして欲しい。仕様に対して設計変更の提案が出来るなど、要求仕様通りに作るだけでなく改善提案できる能力
- もっと早く便利にできる提案ができるようになってほしい
- 提案能力を鍛えるために、Off-JT の時に社内で使うシステム開発について「どういう風にしたらわかりやすいかを考えてみて」と投げかけをしている。仕様は軽い感じで文章で渡して考えさせている
- 仕事の楽しさを感じてほしい。労働として会社に来ている人と、楽しい仕事として来ている人は働き方が全然違う。前者は定時で仕事を終え、後者は詰まっているところを解決したいという意欲がある
- 成功体験があると好きになりやすい
- プログラミングでも論理的に美しい、画面が使いやすい、デザイン的に見た目重視など、人によって得意としているところが違うので、適材適所でやってもらっている

プログラマ育成に関する実態調査(企業調査編)の総括

【プログラミング言語について】

- Javaはすべての調査対象企業で主要言語として使っており、そのほかVB.netや、Web系ではJava script、PHP、C#などがよく使われている
- 一部の企業においてはC言語も使われている
- Pythonはベンチマークしている企業はあるが、実用としてはまだ少数である
- ほとんどの企業では顧客からの要望があればどの言語でも扱うという姿勢である
- 新入社員教育ではJavaが多く採用されている

【学校でのペアプログラミング経験の評価】

- 新入社員教育など育成目的でペアプログラミングを導入している企業は複数あったが、ペアプログラミング経験そのものをプラス評価するということはない

【リモート環境下での開発経験の評価】

- 実務においては、入社してもリモートでも成果物の納期や品質に問題がなければよしとする企業が複数あり、リモート環境下での開発経験を積極的にプラス評価はしていない
- 学生の経験として、スケジュール管理を自分で決めてやるという点を有効だとする企業があった
- リモート環境下で使われるコミュニケーションツールの利用経験を評価する企業があった
- 在宅ワークに関する企業の評価は各企業で異なり、積極的に推進する企業もあれば、コミュニケーション上の問題からあまり評価していない企業まで様々である

【ノーコード開発の評価】

- ほとんどの企業であまり評価していないが、一部の企業では目的が達成されるのであればスクラッチ開発にはこだわらず、ノーコード・ローコード開発を導入すべきだという考え方もあった
- ノーコードはエンドユーザー側の開発者ツールであると考える企業があった
- ノーコード開発を評価しない理由としては、トラブル対応(バグの修正など)やシステム構築環境の継続性を問題視している

【検定取得の実績】

- 国家資格取得やベンダー資格取得を評価する企業が多かったが、仕事ができるかどうかとは別に考えられている
- 手当や報奨金を出している企業も多く、人事評価制度や昇進条件と結びつけている企業

もあった

- 簿記検定取得を評価している企業も複数あった

【プログラムの1ヶ月のコーディング量】

- 業務の種類や難易度によって変わるので測れないとする企業がほとんどで、数値化するとしたら0.6キロから1キロと答える企業もあった

【プログラムの能力評価基準・方法】

- 多くの企業で人事評価制度を導入しており、上司による面談等を通して一定期間の業務実績等を評価している
- 人事評価制度を導入している企業では、評価基準が明文化されており、チェックリストや採点表を用いて上長が評価する仕組みになっている
- テクニカルスキルだけでなく、ヒューマンスキルの部分も評価の対象となっている

【プログラムに言語やアルゴリズム以外で求める能力】

- ほとんどの企業でコミュニケーション能力が求められている
- テクニカルスキル以上にコミュニケーション能力を評価する企業がほとんどであった
- 自分で調べる能力や、不具合やバグを見つけられる能力をあげる企業もあった

【自社でのプログラミング教育・教育カリキュラム】

- 多くのシステム開発会社では、新入社員を入社後一定期間(概ね2ヶ月～半年)、外部のプログラミング研修機関に委ね、プログラム教育を行っている
- 外部の研修機関での研修を終えると自社での教育に入る
- 自社での教育はOJTで先輩の指導のもとプログラム開発の一部に携わる形で行われる
- 研修教育期間は最長1年であったが、ほとんどの企業では半年程度でOJTとして業務に携わる
- 社員同士で勉強会を定期開催している企業もあった

【プログラミング教育で強化して欲しい領域】

- 実務と学校教育では異なるため、あまり学校教育に過度に期待することはないが、一部には開発経験そのものを評価する企業があった
- テストは全ての企業が重要視しており、テスト教育を強化して欲しいとする企業があった

【プログラミングスキルの評価制度】

- 実務における評価は、プログラミングスキルだけではないので、スキルそのものを評価するということはない

【コーディング基準を持っているか】

- プロジェクトごとにコーディングの基準を決めている企業がほとんどであった

【テスト設計の指針を持っているか】

- ほとんどの企業ではテストに関する指針を持っており、指針として明文化されている企業もあれば、メモで受け継がれているなど文化として定着している企業もあった

【その他】

- 学校でのプログラミング経験は一定の評価はあるものの、採用に関してはプログラム経験がない者であってもコミュニケーション能力やその他の部分が評価され採用に結びつくことが多い
- 資格も同様に、一定の評価はしているがそれ以上に人物評価が優先されることが多い
- キャリアパスとしては、多くの企業でプログラマ、システムエンジニア、プロジェクトリーダー、プロジェクトマネジャーの順に昇進していくが、スペシャリストとしてプログラマに留まる道を用意している企業もあった

遠隔教育プラットフォーム

ポータル画面

教員用ポータル画面

 aspire-dev ▾

[ホーム](#) [メッセージ](#) [ワークスペース情報](#)

遠隔教育プラットフォーム

 GitHub Classroom

[Go to classroom](#)

授業チャンネル

lesson1

[Go to channel](#)

 [成績シート閲覧](#)

lesson2

[Go to channel](#)

 [成績シート閲覧](#)

課題チャンネル

kadai1

[Go to channel](#)

lesson1_kadai1

[Go to channel](#)

kadai1

[Go to channel](#)

lesson1_kadai1

[Go to channel](#)

 aspire-dev ▾

[ホーム](#) [メッセージ](#) [ワークスペース情報](#)

遠隔教育プラットフォーム

 GitHub Classroom

[Go to classroom](#)

授業チャンネル

jopsデモ_フレームワーク

[Go to channel](#)

lesson1

[Go to channel](#)

lesson2

[Go to channel](#)

課題チャンネル

lesson1_kadai1

[Go to channel](#)

 [課題回答ページ](#)

lesson1_kadai2

[Go to channel](#)

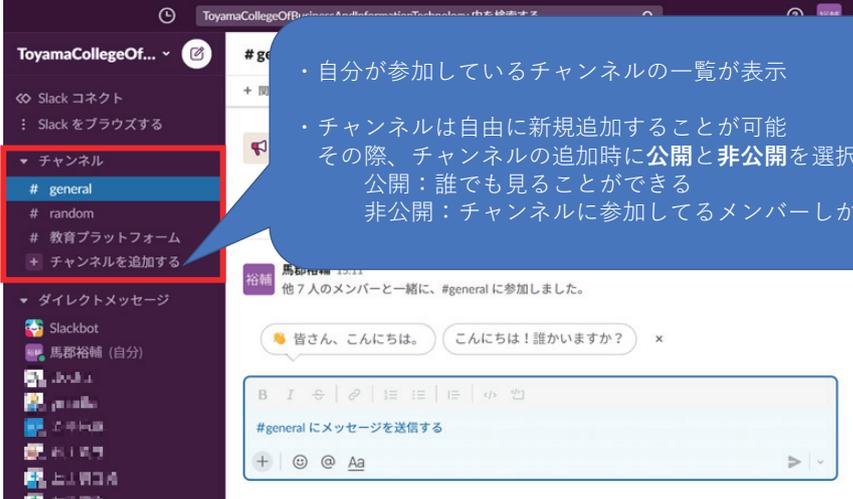
 [課題回答ページ](#)

lesson1_kadai3

[Go to channel](#)

 [課題回答ページ](#)

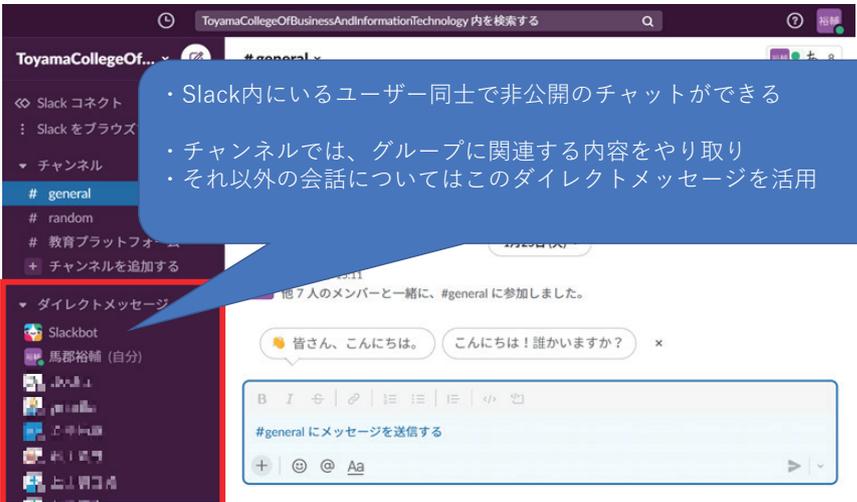
チャンネル一覧



The screenshot shows the Slack interface for a workspace named 'ToyamaCollegeOfBusinessAndInformationTechnology'. On the left sidebar, the 'チャンネル' (Channels) section is highlighted with a red box, showing a list of channels: '# general', '# random', and '# 教育プラットフォーム'. Below this is a '+ チャンネルを追加する' (Add channel) button. The main area shows the '# general' channel with a message from '馬郡裕輔' (Yusaku Umakuni) that says '他7人のメンバーと一緒に、#generalに参加しました。' (Joined #general with 7 other members). Below the message is a text input field with a placeholder '#general にメッセージを送信する' (Send message to #general) and a send button.

- 自分が参加しているチャンネルの一覧が表示
- チャンネルは自由に新規追加することが可能
その際、チャンネルの追加時に公開と非公開を選択できる
公開：誰でも見ることができる
非公開：チャンネルに参加しているメンバーしか内容を見ることができない

ダイレクトメッセージ



The screenshot shows the Slack interface for the same workspace. On the left sidebar, the 'ダイレクトメッセージ' (Direct Messages) section is highlighted with a red box, showing a list of direct messages: 'Slackbot', '馬郡裕輔 (自分)', and several other users. The main area shows the '# general' channel with the same message from '馬郡裕輔' as in the previous screenshot. Below the message is a text input field with a placeholder '#general にメッセージを送信する' (Send message to #general) and a send button.

- Slack内にいるユーザー同士で非公開のチャットができる
- チャンネルでは、グループに関連する内容をやり取り
- それ以外の会話についてはこのダイレクトメッセージを活用

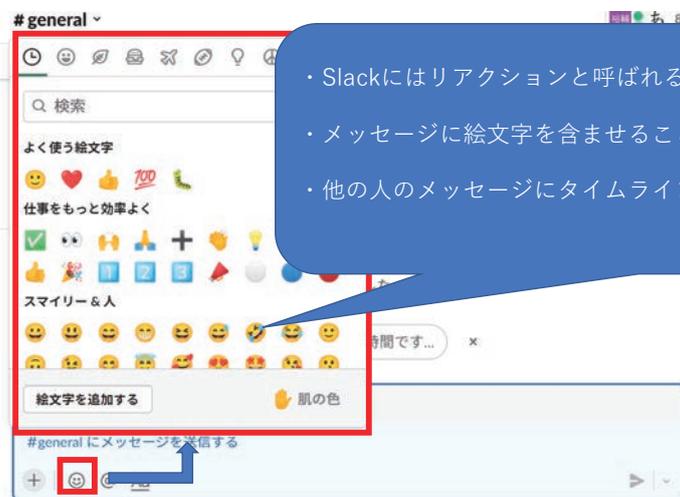
タイムライン



メッセージの入力

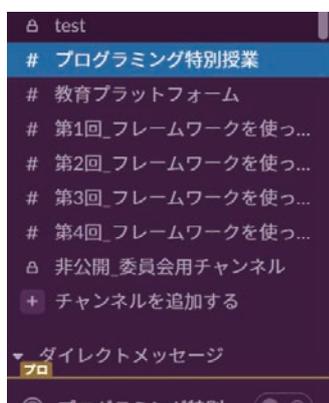


リアクション（絵文字）



- ・ Slackにはリアクションと呼ばれる、絵文字の使用が可能
- ・ メッセージに絵文字を含ませることも可能
- ・ 他の人のメッセージにタイムラインからリアクションを付けることも可能

チャットボット



@help-bot 質問内容
とメッセージを送るとチャットbotから回答

チャットbotからの回答に答える形で求める回答を導き出していく

教員用ポータル画面の表示

The screenshot shows a Slack interface for the workspace 'aspire-dev'. The left sidebar lists various channels, with the 'aspire' channel selected. The main content area displays the '遠隔教育プラットフォーム' (Remote Education Platform) page. This page includes a 'GitHub Classroom' link, a list of '授業チャンネル' (Lesson Channels) with 'lesson1' and 'lesson2' (each with a '成績シート閲覧' link and a 'Go to channel' button), and a list of '課題チャンネル' (Assignment Channels) with 'kadal1' and 'lesson1_kadal1' (each with a 'Go to channel' button). A blue callout box points to the main content area.

ポータル画面を表示した教員が担当する授業チャンネル、課題チャンネルの一覧と遷移ボタン、成績シートへのリンクを表示

学生用ポータル画面の表示

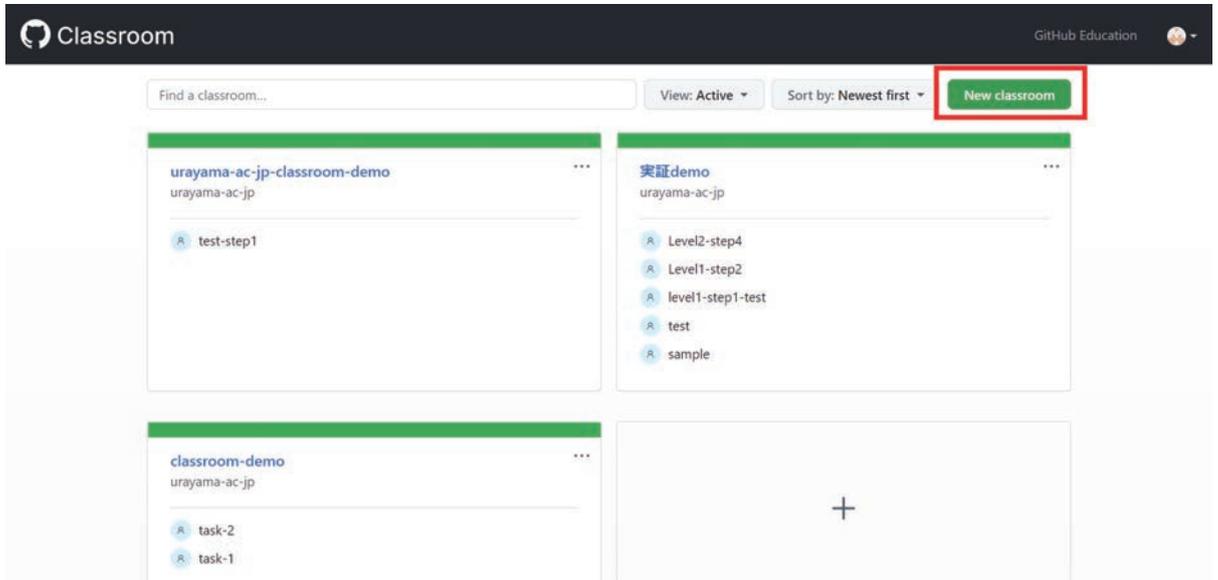
The screenshot shows a Slack interface for the workspace 'aspire-dev'. The left sidebar lists various channels, with the 'aspire' channel selected. The main content area displays the '遠隔教育プラットフォーム' (Remote Education Platform) page. This page includes a 'GitHub Classroom' link, a list of '授業チャンネル' (Lesson Channels) with 'jobsデモ_フレームワーク', 'lesson1', and 'lesson2' (each with a 'Go to channel' button), and a list of '課題チャンネル' (Assignment Channels) with 'lesson1_kadal1', 'lesson1_kadal2', 'lesson1_kadal3', and '課題回答ページ' (each with a 'Go to channel' button). A blue callout box points to the main content area.

ポータル画面を表示した学生が割り当てられている授業チャンネル、課題チャンネルの一覧と遷移ボタン、配布された課題のURLリンクを表示

課題の作成

Github Classroomの操作

- 先生(TA)・学生・クラス・課題を設定
 - 授業用のクラスを作成



- 所属する組織（浦山学園）を選択



First, select an organization for your new classroom.



- 授業名を設定

New classroom

Name your classroom.

Classroom name

Using your course name and section can help students identify your classroom.

Create classroom

- 必要であれば、他の先生やTAの招待も可能

urayama-ac-jp-20220204-demo

Success! Next, add collaborators to your classroom.

Invite TAs and admins [Skip this for now](#)

- 1 First, invite TAs and admins to your GitHub organization.**

Organizations are managed on github.com, not classroom.github.com. In order to grant admin access, you must first add them as owners to your GitHub organization, `urayama-ac-jp`.

[Invite TAs and admins](#) →
- 2 Then, ask them to join this classroom URL.**

In addition to inviting them to the organization, you also need to send them the classroom invitation URL above to join your classroom. Once they sign in to GitHub Classroom using the invitation link, they will automatically be added to this classroom as admins.

Continue

© 2022 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)  [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

- 授業に紐づける学生を名寄せ（今回はGoogleClassroomから連携）

urayama-ac-jp-20220204-demo

Last step! Add students to your roster.

Connect to a learning management system

[Skip this for now](#)

Connecting GitHub Classroom to your institution's learning management system will allow you to automatically import your roster. [Learn more.](#)



Google Classroom



Canvas



Moodle



Sakai



Other LMS

Or, create your roster by listing students

Enter your list of students, one per line.

Student identifier

[Upload a CSV or text file](#)

[Continue](#)

- 連携された情報を登録

Classroom GitHub Education

Classrooms / urayama-ac-jp-20220204-demo

urayama-ac-jp-20220204-demo

urayama-ac-jp

Assignments 0 **Students 0** TAs and Admins 1 Settings

Google Classroom integration was successfully configured. ×

Students

Let's set up your roster

The student roster allows you to easily track student progress on your dashboard.

Import students from your institution

It looks like you are connected to Google Classroom. GitHub Classroom can automatically import your roster for you!

[Import from Google Classroom](#)

Or, create your roster by listing students

Enter your list of students, one per line.

Student identifier

[Upload a CSV or text file](#)

[Create roster](#)

© 2022 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

- クラスに名寄せされた学生の一覧を確認

The screenshot shows the GitHub Classroom interface for a class named 'urayama-ac-jp-20220204-demo'. The top navigation bar includes the Classroom logo and 'GitHub Education'. Below the class name, there are tabs for 'Assignments 0', 'Students 23', 'TAs and Admins 1', and 'Settings'. The 'Students' tab is active, displaying a 'Classroom roster' section. This section has buttons for 'Sync from Google Classroom' and 'Download'. It shows a list of 'All students 23' and 'Unlinked GitHub accounts 0'. Three 'Unlinked user' entries are visible, each with a 'Link to GitHub account' button and a trash icon. A 'Delete this roster' warning is shown below the roster, stating that deleting the roster will delete all students but not assignments, repos, or submissions. The footer contains copyright information and various links like 'Terms', 'Privacy', 'Security', 'Status', 'Help', 'Contact GitHub', 'Pricing', 'API', 'Training', 'Blog', and 'About'.

- 課題の登録を行っていく(課題はクラスを作った後で作成することも可能)

The screenshot shows the GitHub Classroom interface for the same class, but now the 'Assignments' tab is active and highlighted with a red box. The 'Assignments' section features two main cards. The first card, 'Create an assignment to get started', explains that creating an individual assignment generates a repository for each student, while a group assignment allows collaborative work. It includes a 'Create an assignment' button and a link to 'Learn more about individual and group assignments'. The second card, 'Need to teach Git & GitHub fundamentals?', mentions that the Classroom team has created a starter assignment for teaching Git & GitHub fundamentals, with a 'Use starter assignment' button and a 'Learn more' link. The footer is identical to the previous screenshot.

○ 課題名を設定

Classroom GitHub Education

Classrooms / urayama-ac-jp-20220204-demo / New assignment

- Assignment basics
- Starter code and environment
- Grading and feedback

Let's set up the basics for your assignment.

Assignment title
20220204-kadai
Student assignment repositories will have the prefix: 20220204-kadai

Deadline (optional)
[Empty field]

Individual or group assignment
Individual assignment

Repository visibility
Private repositories will only be visible to the student and the classroom owners. Public repositories will be visible to everyone, including other students.
 Private Public

Grant students admin access to their repository
Editing this after assignments are created will not retroactively change permissions.

Cancel Continue

© 2022 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

○ 学生に配布するリポジトリを選択（ここで選択されたリポジトリがコピーされて、学生が課題を行うリポジトリが作成される）

Classroom GitHub Education

Classrooms / urayama-ac-jp-20220204-demo / New assignment

- ✓ Assignment basics
- Starter code and environment
- Grading and feedback

Add your starter code and choose an optional online IDE.

Add a template repository to give students starter code
Your assignment will be created with empty student repositories if you don't add starter code. Changes to starter code after students have accepted the assignment will not retroactively change existing student repositories.
Note: All starter code must use a [template repository](#). Your starter code repository must be either in the same organization as this classroom or a public repository if elsewhere. [Learn about transferring your repositories.](#)

urayama-ac-jp/practice1-level1-step1

Add a supported editor
Automatically include a link to an editor in students' repositories to give them a one-click experience for getting started coding, running, and collaborating on their code.
Select an editor

← Back Cancel Continue

© 2022 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

- フィードバックを行う設定を追加

Classroom GitHub Education

Classrooms / urayama-ac-jp-20220204-demo / New assignment

Assignment basics ✓
Starter code and environment ✓
Grading and feedback ●

Set up autograding and feedback.

Add autograding tests
Autograding tests help provide feedback for students immediately upon submission using GitHub Actions. Add a test to enable autograding.

Add test ▾

Enable feedback pull requests
A pull request will automatically be created on all student repository submissions. Pull requests allow you to answer questions and provide feedback.

← Back Cancel **Create assignment**

© 2022 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

- 課題を配布するためのURLが発行されるのでコピー
※別途プラットフォームを使って配布

Classroom GitHub Education

Classrooms / urayama-ac-jp-20220204-demo / 20220204-kadai1

"20220204-kadai1" has been created!

20220204-kadai1

Individual assignment ● Active

https://classroom.gi [Copy] [Edit] [Download]

Search by GitHub username or student identifier

Assignment is ready
Share the invitation URL with students and visit it to view how students accept an assignment.
Got it!

Classroom roster Unlinked account Passing Sort

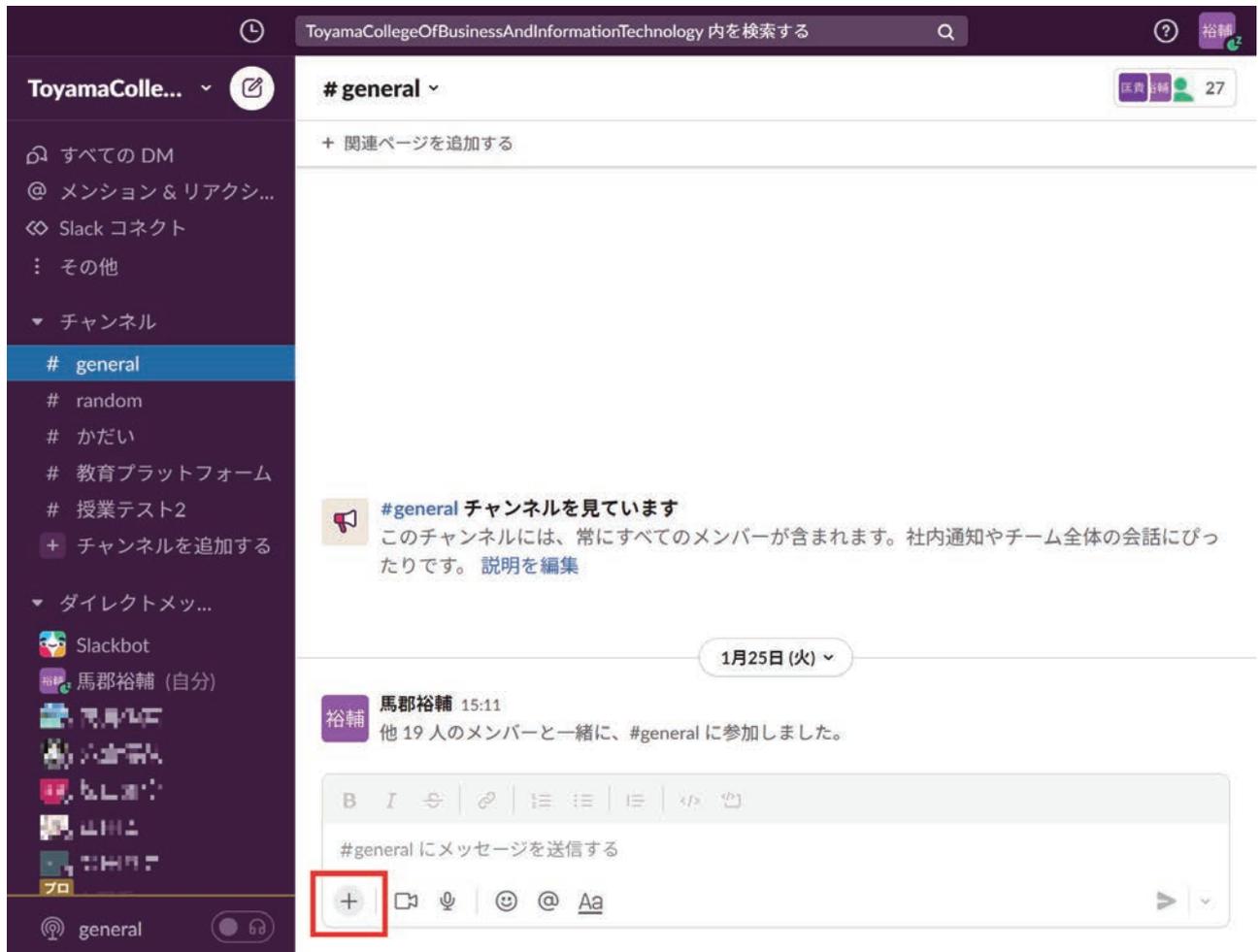
Share the invitation link with your students so they can accept the assignment.
Copy invitation link

© 2022 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

授業の作成

Slackの操作

- 教育プラットフォーム用のアプリを使って授業を作成
ショートカットから起動



- 「授業」と入力し、選択

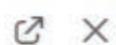


- 授業を選択





授業作成



slack授業チャンネル作成

セレクトボックスからslackに取り込みたい授業を選択してください。

授業

授業を選択してください



プログラミング特別授業

授業テスト1

授業テスト2

- 授業用のチャンネルが作成される



- 授業を作成した先生のみに進捗と成績を確認できるURLが通知される

#授業テスト2 チャンネルをどんどん活用していきましょう！
今日、@aspire さんがこのチャンネルを作成しました。説明を追加する

👤+ メンバーを追加する

✉️ このチャンネルにメールを転送する

 **aspire** アプリ 22:48
#授業テスト2 に参加しました。

👁️ あなただけに表示されています

 **aspire** アプリ 22:48
💡 マスタURLのご案内

授業の情報が記載されているシートURLです。
成績や進捗はこちらのシートからご覧ください。

授業テスト2

https://docs.google.com/spreadsheets/d/1DplHzFKDeETI-_S_qDhBaSaD787z4abQbtvrmn8kov0

 **馬郡裕輔** 22:48
aspireさんにより他2人のメンバーと一緒に #授業テスト2 に追加されました。

B *I*       

#授業テスト2 にメッセージを送信する

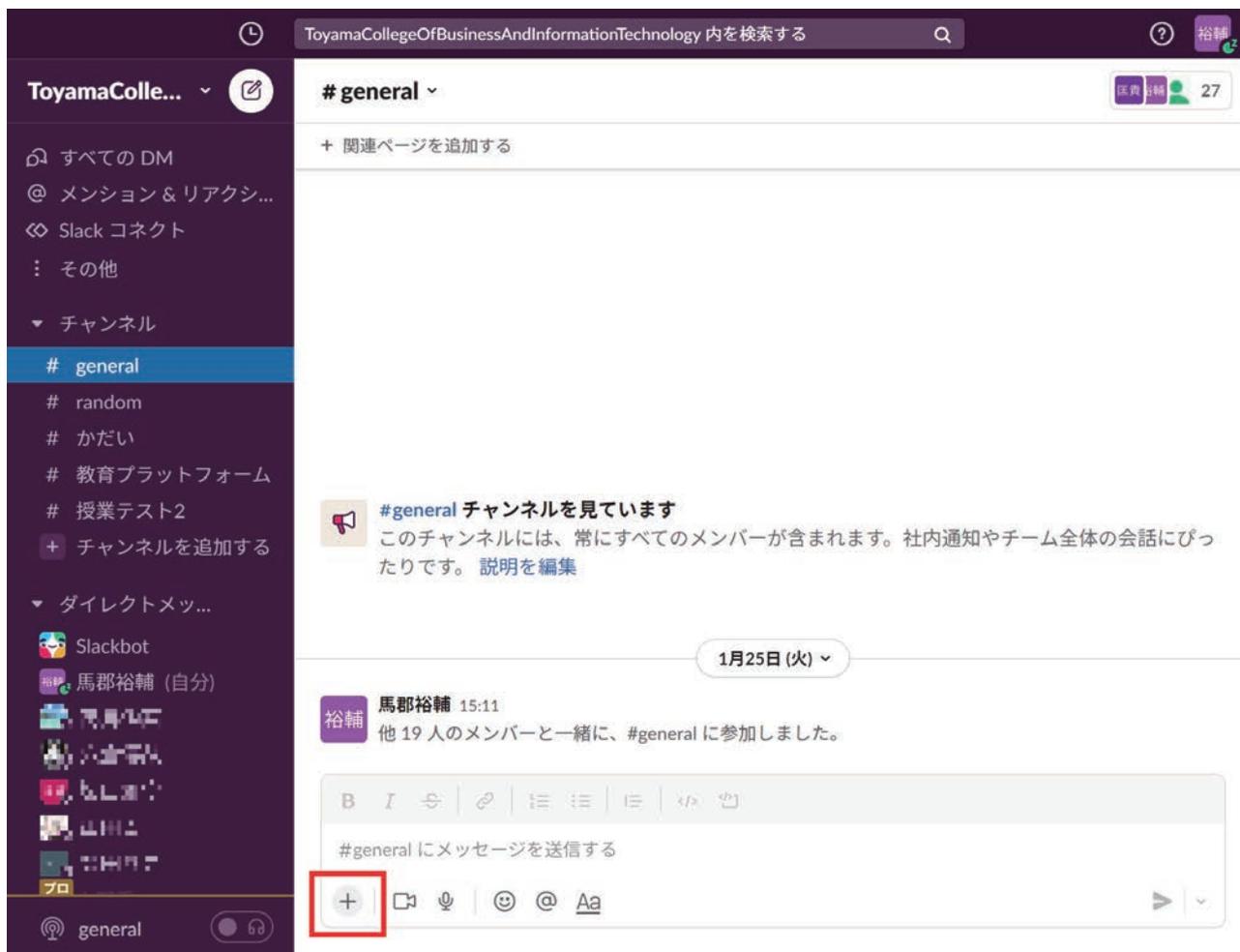
- 通知のあったURLから進捗確認と成績を参照

	A	B	test-1947		test-2141				
1	学生名	全体進捗	進捗	点数(手入力)	進捗	点数(手入力)	進捗	点数(手入力)	進捗
3	上山	50%	100%		未提出				
4	馬郡	0%	未提出		未提出				
5	藤崎	0%	66%		未提出				
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									

課題の作成

Slackの操作

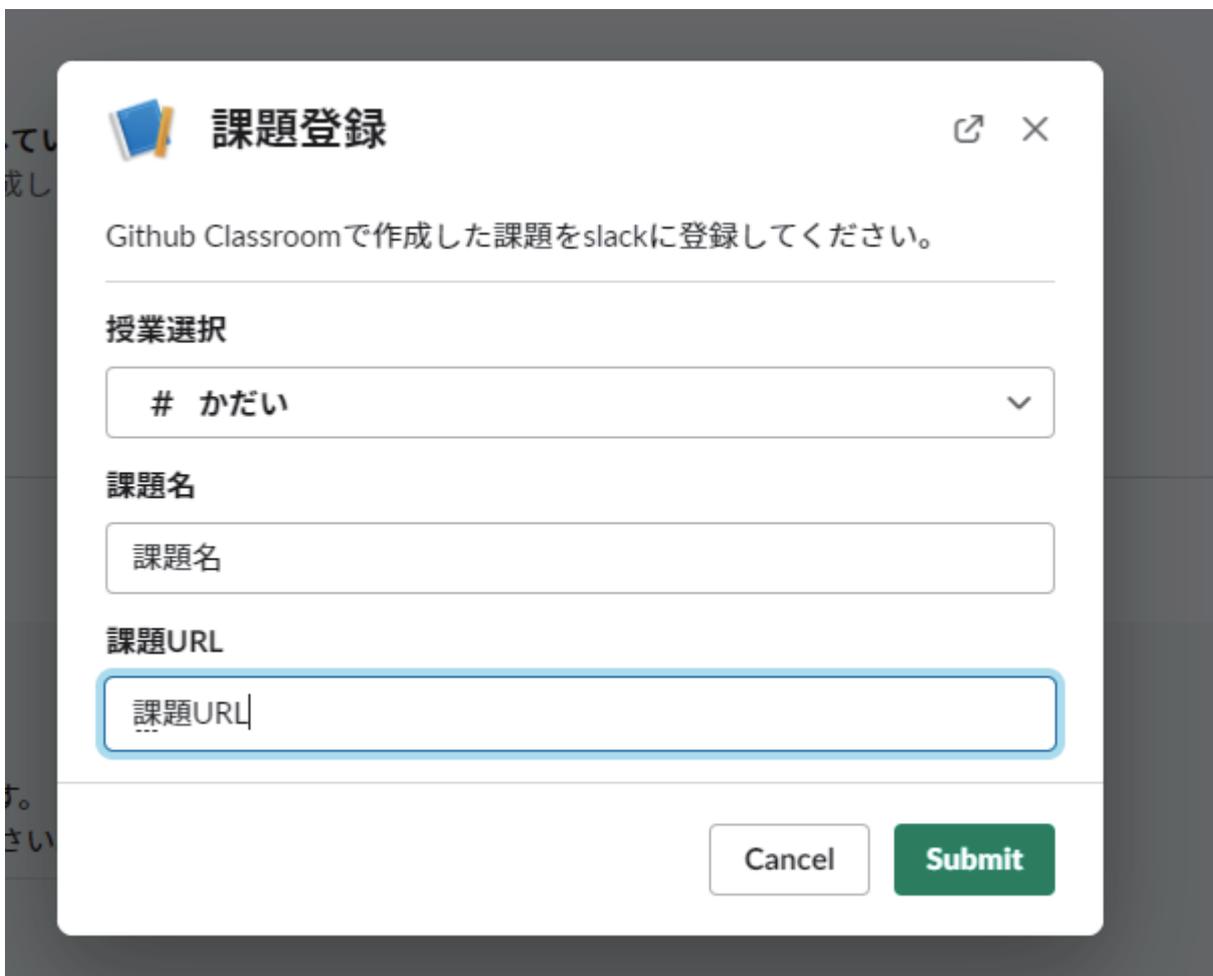
- 教育プラットフォーム用のアプリを使って課題を作成
ショートカットから起動



- 「課題」と入力し、選択



- 作成する課題の設定（GitHub Classroomで作成した課題URLを入力）



- 授業選択で指定した授業（チャンネル）の学生に課題が配布される

#かだい チャンネルをどんどん活用していきましょう！
今日、@aspire さんがこのチャンネルを作成しました。 [説明を追加する](#)

[メンバーを追加する](#)
[このチャンネルにメールを転送する](#)

今日

aspire アプリ 19:48
他3人のメンバーと一緒に、#かだいに参加しました。

aspire アプリ 19:48
NEW 新しい課題のお知らせ **NEW**
下記のURLからアクセスし、課題に取り組んでください。
質問はこちらのチャンネルどうぞ！

課題名
かだい

課題URL
<https://classroom.github.com/a/m-wQxXFc>

B I

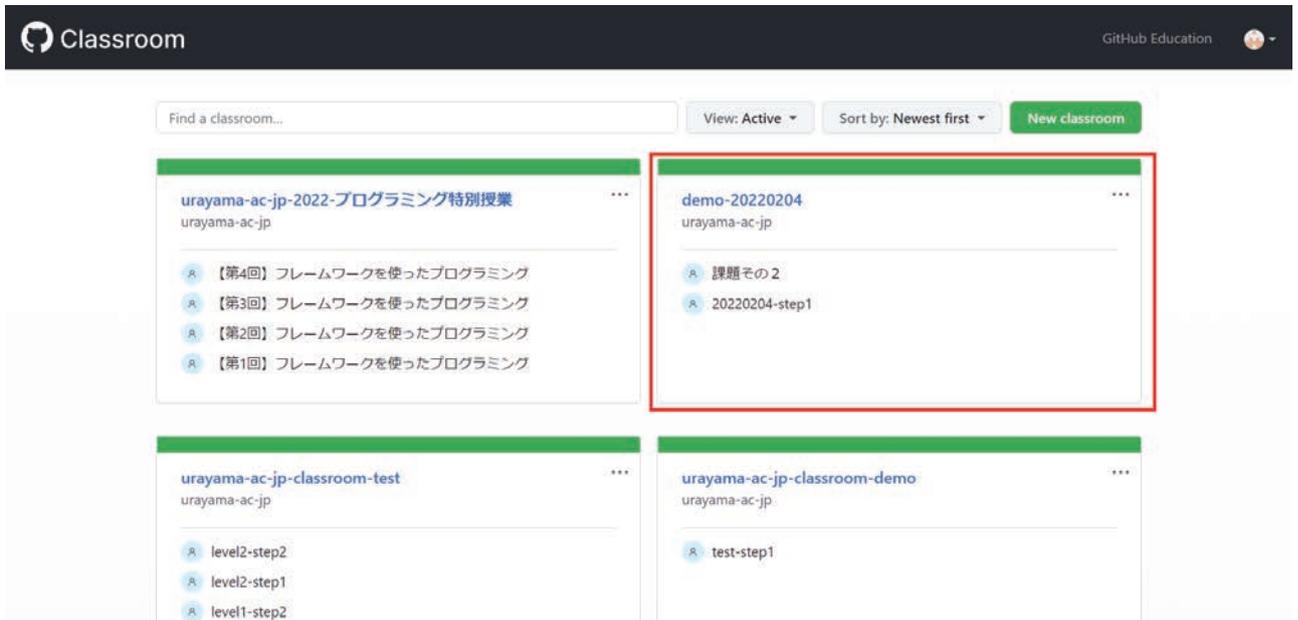
#かだいにメッセージを送信する

Aa

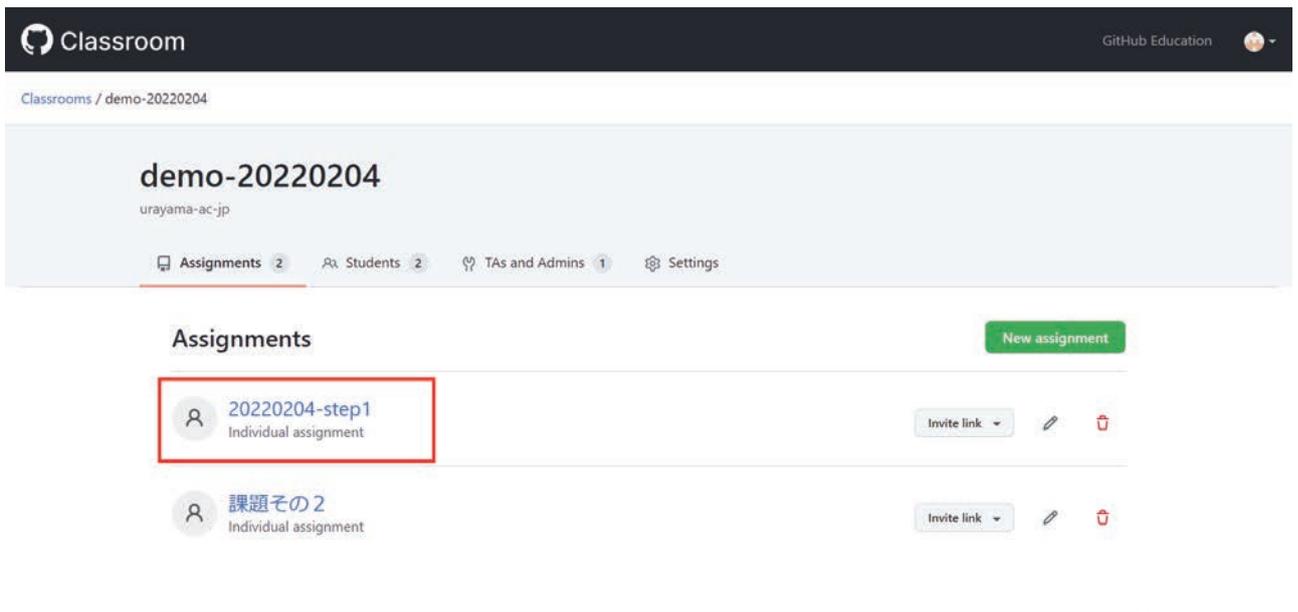
課題のフィードバック

GitHub Classroomの操作

- GitHub Classroomから、フィードバックを行う授業を選択



- 授業から、フィードバックを行う課題を選択



- 対象の学生を確認し、フィードバック用のボタンを押下

Classroom

GitHub Education

Classrooms / demo-20220204 / 20220204-step1

20220204-step1

Individual assignment Active

<https://classroom.gitl> Edit Download

Search by GitHub username or student identifier

Classroom roster		Unlinked accounts	Accepted	Submitted	Passing	Sort
	ymagour Link to student identifier	5 commits				
	上山 @jops-akamiya	2 commits				

- 学生の更新履歴の一覧からフィードバックを行う更新を選択

urayama-ac-jp / 20220204-step1-ymagour Private

generated from urayama-ac-jp/practice1-level1-step1

Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

Feedback #1

Open github-classroom wants to merge 4 commits into Feedback from main

Conversation 0 Commits 4 Checks 1 Files changed 2 +2 -2

github-classroom bot commented 22 days ago • edited

GitHub Classroom created this pull request as a place for your teacher to leave feedback on your work. It will update automatically. **Don't close or merge this pull request**, unless you're instructed to do so by your teacher. In this pull request, your teacher can leave comments and feedback on your code. Click the **Subscribe** button to be notified if that happens. Click the **Files changed** or **Commits** tab to see all of the changes pushed to `main` since the assignment started. Your teacher can see this too.

Notes for teachers
Subscribed: @ymagour

github-classroom bot and others added 4 commits 22 days ago

- Setting up Github Classroom Feedback Verified X 03f7f16
- test ✓ 9906640
- update StepController.java Verified X 0841ef5
- Update README.md Verified X 176bcea

Add more commits by pushing to the `main` branch on `urayama-ac-jp/20220204-step1-ymagour`.

All checks have failed 1 failing check Hide all checks

- Submission CI / build-with-upload-artifact (push) Failing after 29s — build-with-upload-artifact Details

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Write Preview H B I `< >`

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close pull request Comment

Remember, contributions to this repository should follow our GitHub Community Guidelines.

ProTip! Add comments to specific lines under Files changed.

Reviewers: No reviews. Still in progress? Convert to draft.

Assignees: No one—assign yourself.

Labels: None yet.

Projects: None yet.

Milestone: No milestone.

Linked issues: Successfully merging this pull request may close these issues. None yet.

Notifications: Customize. Subscribe. You're not receiving notifications from this thread.

1 participant

Lock conversation

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

- 更新内容を確認

The screenshot shows a GitHub Pull Request interface. At the top, there's a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. The repository name is 'urayama-ac-jp / 20220204-step1-ymagour' and it's marked as 'Private'. Below this, there are buttons for Watch, Fork, and Star. The main navigation bar includes Code, Issues, Pull requests (1), Actions, Projects, Wiki, Security, Insights, and Settings.

Feedback #1

github-classroom wants to merge 4 commits into `feedback` from `main`

Conversation 0 | Commits 4 | Checks 1 | Files changed 2 | +2 -2

Update Step1Controller.java

error

main

ymagour committed 19 days ago Verified | commit 0841ef50ea7a91dd1cc66b382fbd1f0e2ddddd0c

12	@GetMapping	12	@GetMapping
13	public String index(Model model) {	13	public String index(Model model) {
14	// 課題 level1 step1	14	// 課題 level1 step1
15	- model.addAttribute("title", "hello world!");	15	+ model.addAttribute("title", "hello!");
16	return "step1/index";	16	return "step1/index";
17	}	17	}
18	}	18	}

© 2022 GitHub, Inc. | Terms | Privacy | Security | Status | Docs | Contact GitHub | Pricing | API | Training | Blog | About

- コードに対してフィードバックのコメントを記載し、登録

The screenshot shows a GitHub Pull Request page for the repository `urayama-ac-jp / 20220204-step1-ymagour`. The pull request is titled "Feedback #1" and is currently open. It shows a diff for the file `Step1Controller.java`. The diff highlights a change on line 15, where a line of code is being added: `model.addAttribute("title", "hello world!");`. Below the diff, there is a comment box with a red border, containing the text "Leave a comment". The comment box also includes a "Start a review" button. The footer of the page shows the GitHub logo and copyright information for 2022.

課題の開始手順について

- 課題を始める際は、以下の操作を行ってください
- なお、2.の画面は初回のみ表示されます

手順

1. Slackの課題チャンネルから赤枠の課題URLをクリック



The screenshot shows a Slack channel interface. At the top, there is a channel header for "#かだい" with a message from @aspire: "#かだい チャンネルをどんどん活用していきましょう！ 今日、@aspire さんがこのチャンネルを作成しました。説明を追加する". Below this are two action buttons: "メンバーを追加する" and "このチャンネルにメールを転送する". A date separator for "今日" is visible on the right. The main message is from "aspire アプリ" at 19:48, stating "他3人のメンバーと一緒に、#かだいに参加しました。" Below this is another message from "aspire アプリ" at 19:48 with a "NEW" badge, titled "新しい課題のおしらせ" and containing the text: "下記のURLからアクセスし、課題に取り組んでください。 質問はこちらのチャンネルにどうぞ！". Underneath is a "課題名" section with the name "かだい". A "課題URL" section is highlighted with a red border, showing the URL "https://classroom.github.com/a/m-wQxXFc". At the bottom, there is a text input field with the placeholder "#かだいにメッセージを送信する" and a toolbar with icons for adding attachments, emojis, and mentions.

2. 以下の画面で自分の名前を選択
その際、ポップアップが表示された場合はOKを押下する
(この画面は初回の一度のみ表示)



Join the classroom:

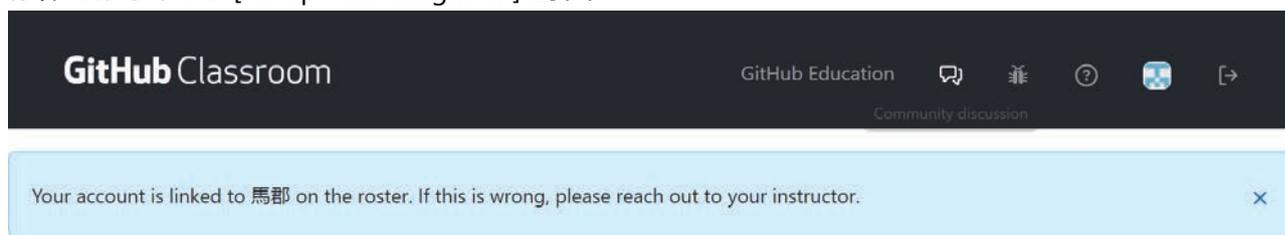
urayama-ac-jp-20220204-demo

To join the GitHub Classroom for this course, please select yourself from the list below to associate your GitHub account with your school's identifier (i.e., your name, ID, or email).

[Can't find your name? Skip to the next step →](#)

Identifiers	
██████████	>
██████████	>
██████████	>
██████████	>
██████████	>
██████████	>
██████████	>
██████████	>
██████████	>

3. 赤枠にあるボタン[Accept this assignment]を押下



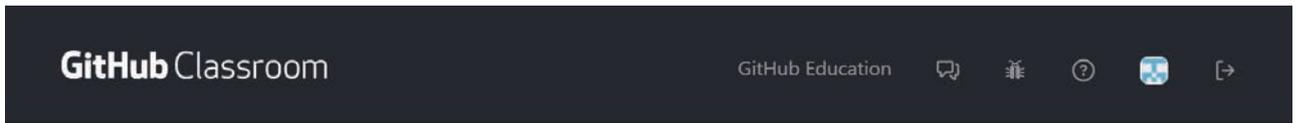
urayama-ac-jp-classroom-test

Accept the assignment — test-2141

Once you accept this assignment, you will be granted access to the `test-2141-urayamahub` repository in the `urayama-ac-jp` organization on GitHub.

Accept this assignment

4. 以下の画面が表示されたら画面の再読み込み（F5キーを押下）を行う



You accepted the assignment, **test-2141** . We're configuring your repository now. This may take a few minutes to complete. Refresh this page to see updates.

Note: You may receive an email invitation to join [urayama-ac-jp](#) on your behalf. No further action is necessary.



Join the GitHub Student Developer Pack

Verified students receive free GitHub Pro plus thousands of dollars worth of the best real-world tools and training from GitHub Education partners — for free. [Learn more](#)

Apply

5. 以下の画面が表示されるので、赤枠にあるリンクを押下
※表示されるリンク名は異なります



You're ready to go!

You accepted the assignment, **test-2141**.

Your assignment repository has been created:

 <https://github.com/urayama-ac-jp/test-2141-urayamahub>

We've configured the repository associated with this assignment (update).

Note: You may receive an email invitation to join `urayama-ac-jp` on your behalf. No further action is necessary.



Join the GitHub Student Developer Pack

Verified students receive free GitHub Pro plus thousands of dollars worth of the best

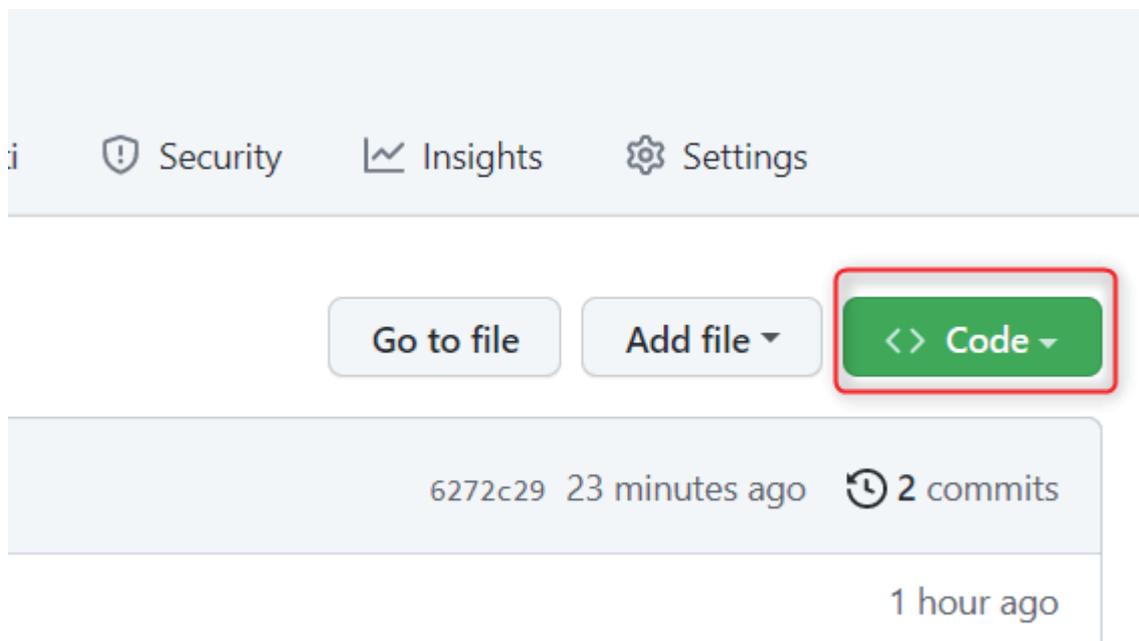
6. 以下のGitHubの画面が表示されることを確認
画面をスクロールして課題内容を確認してください

The screenshot shows a GitHub repository page for 'urayama-ac-jp / test-2141-urayamahub'. The repository is private and was generated from 'urayama-ac-jp/practice1-level1-step1'. The page displays a list of files and folders, all with 'Initial commit' status and a timestamp of '1 minute ago'. The files include configuration files like '.devcontainer', '.github', '.vscode', '.dockerignore', '.editorconfig', and '.gitignore', as well as source files like 'README.md', 'build.gradle', and 'gradlew'. The right sidebar shows repository statistics: 0 stars, 0 watching, and 0 forks. It also includes sections for 'Releases' (no releases published) and 'Packages' (no packages published).

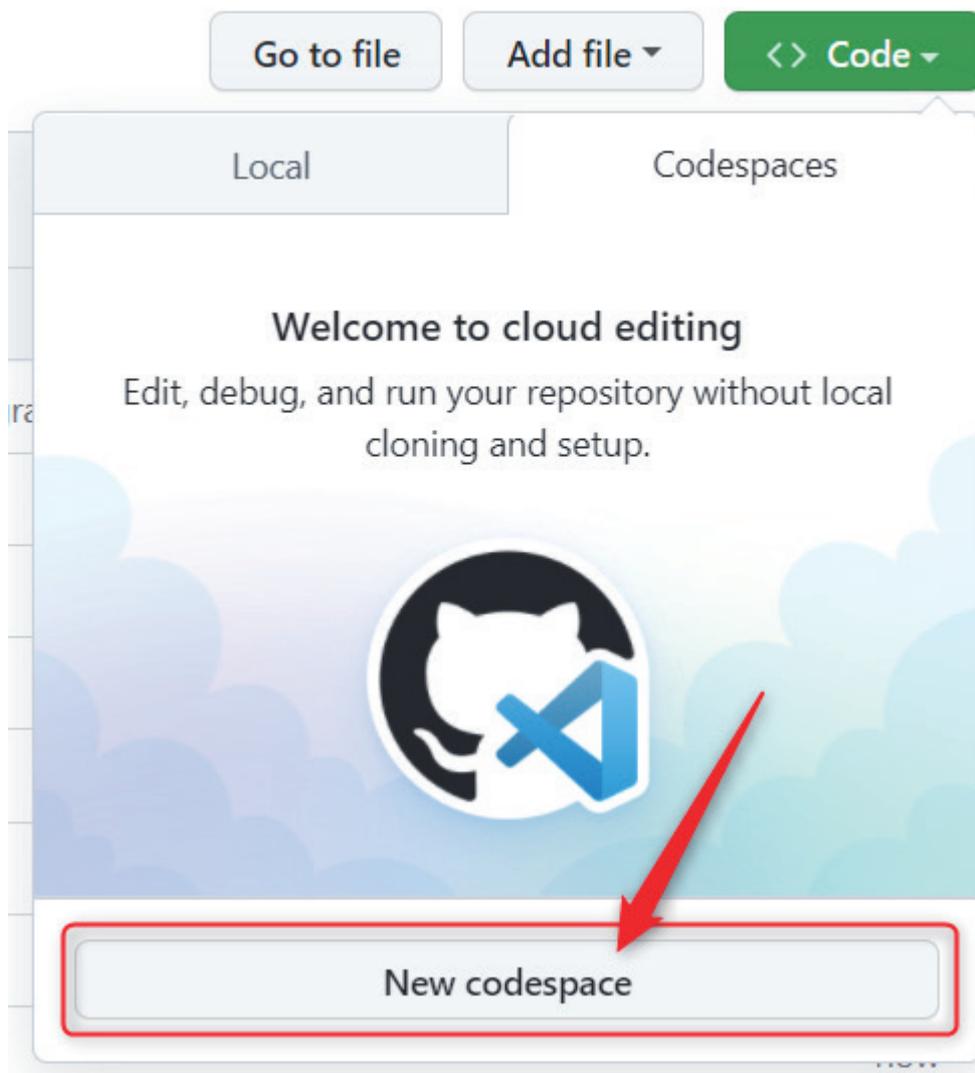
File/Folder	Commit Message	Time
github-classroom[bot]	Setting up GitHub Classroom...	1 minute ago
.devcontainer	Initial commit	1 minute ago
.github	GitHub Classroom Feedback	1 minute ago
.vscode	Initial commit	1 minute ago
gradle/wrapper	Initial commit	1 minute ago
src	Initial commit	1 minute ago
styles	Initial commit	1 minute ago
.dockerignore	Initial commit	1 minute ago
.editorconfig	Initial commit	1 minute ago
.gitignore	Initial commit	1 minute ago
README.md	Initial commit	1 minute ago
build.gradle	Initial commit	1 minute ago
gradlew	Initial commit	1 minute ago

Codespacesの実行手順

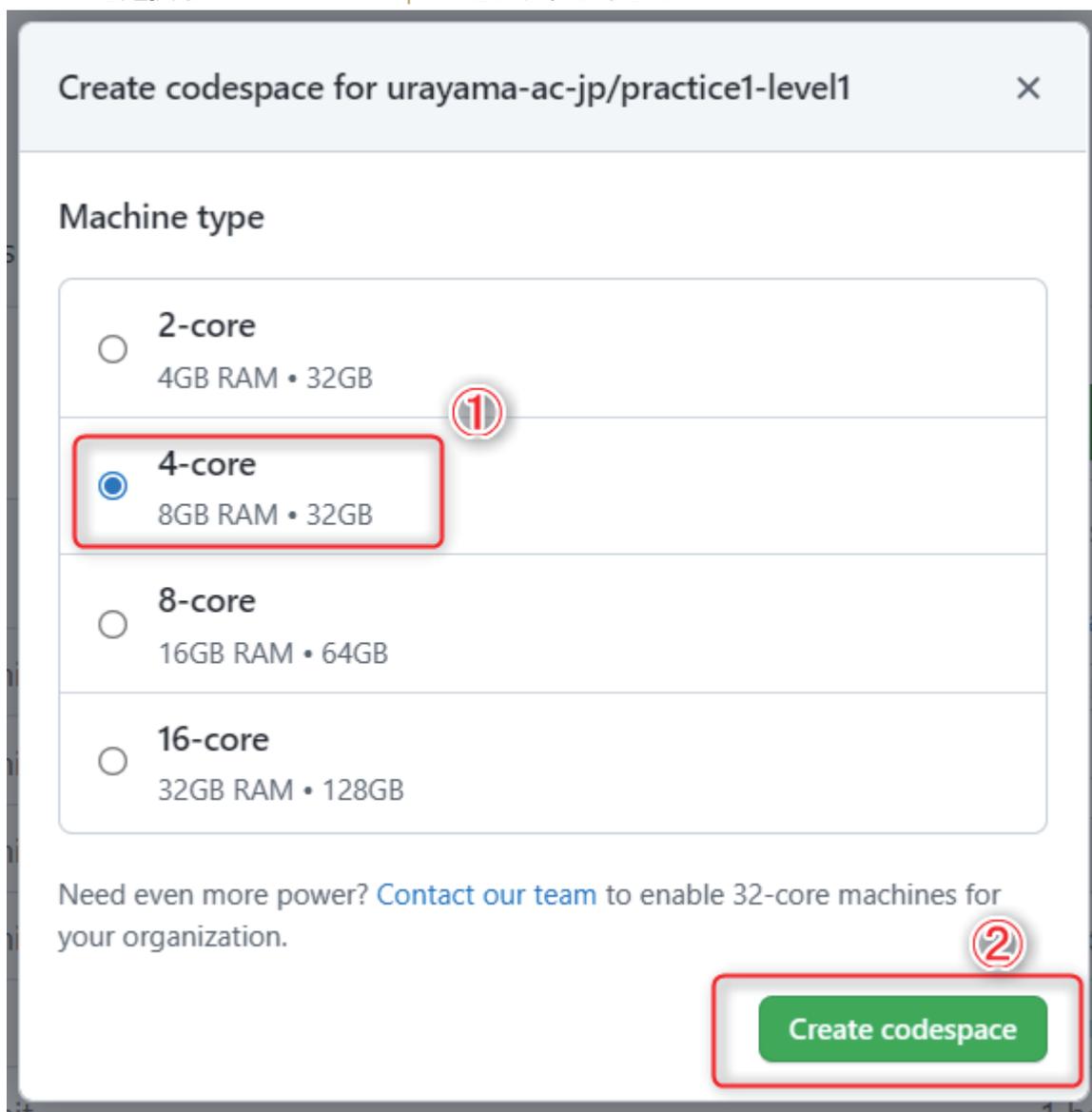
1. Codeをクリックします。



2. New codespaceを選択します。



3. 4-coreを選択し、Create codespaceをクリックします。

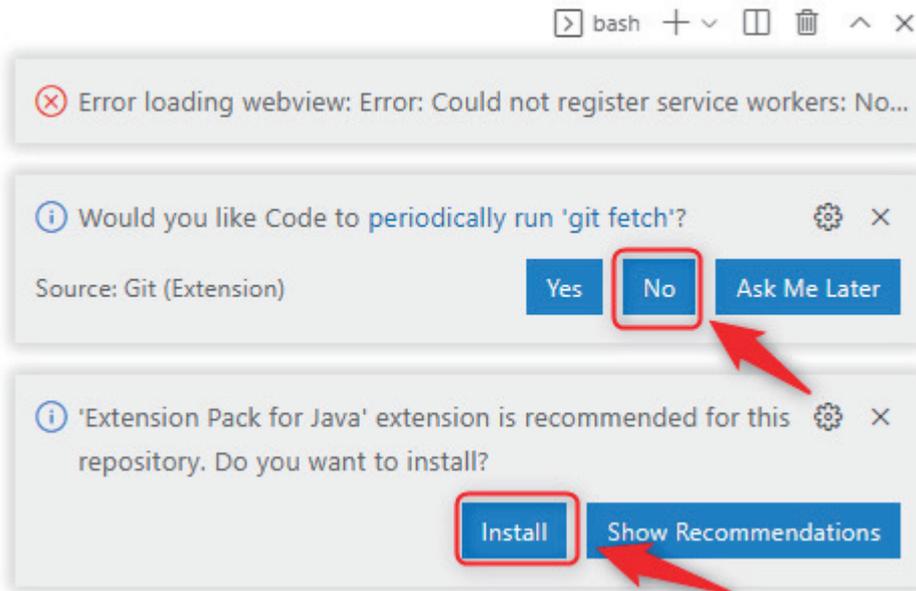


4. しばらく待ちます。

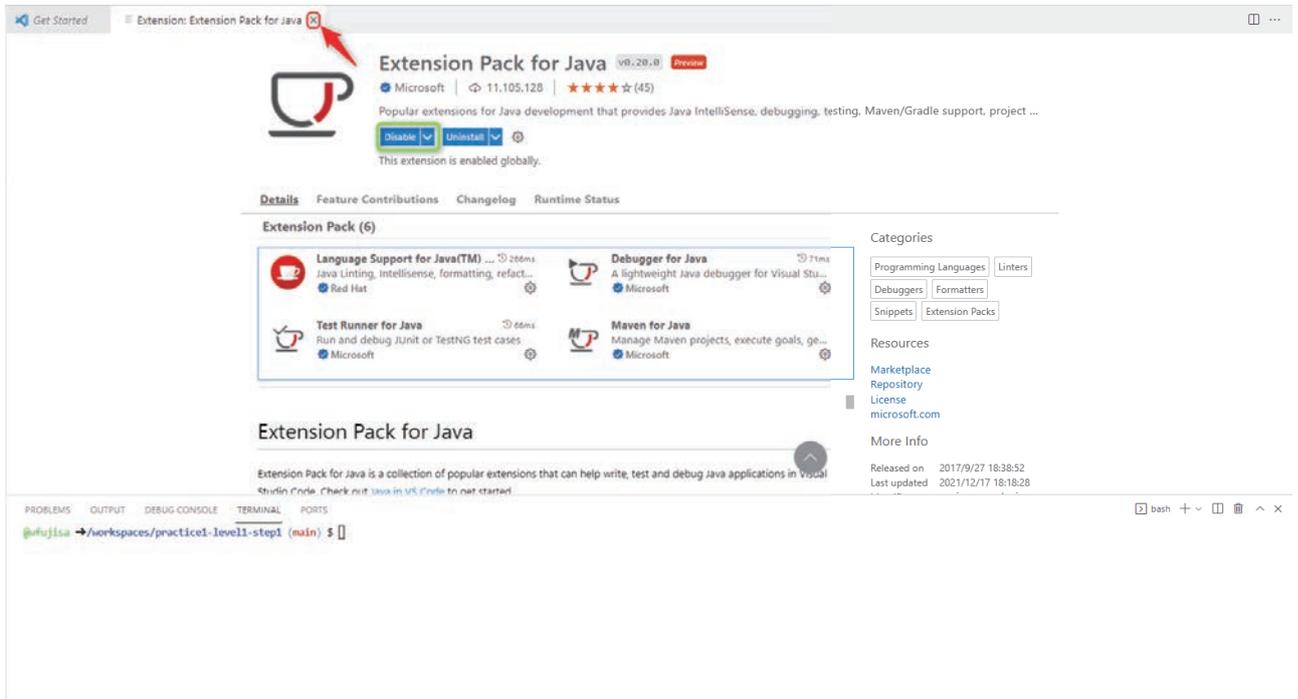
Setting up your codespace



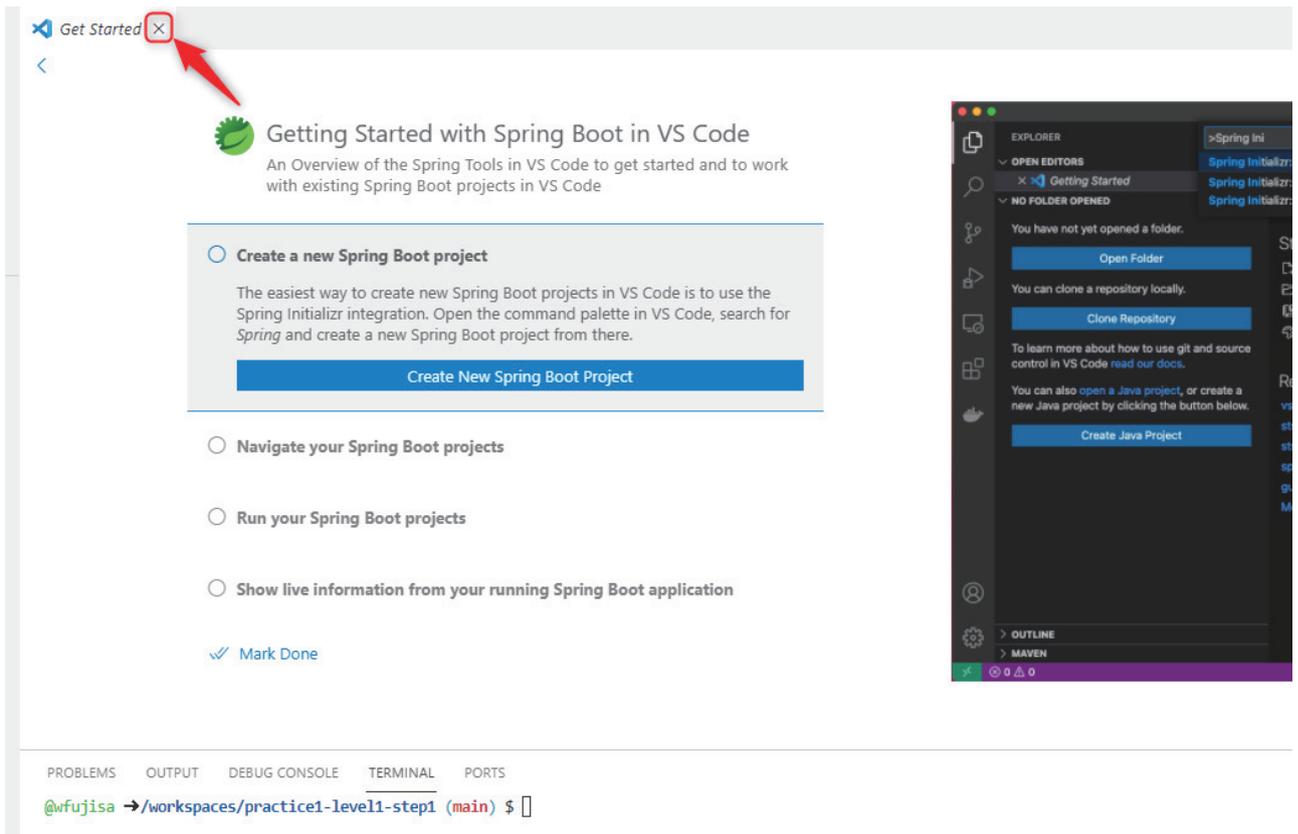
5. 画像のダイアログが表示されたら、以下のボタンを押してください。



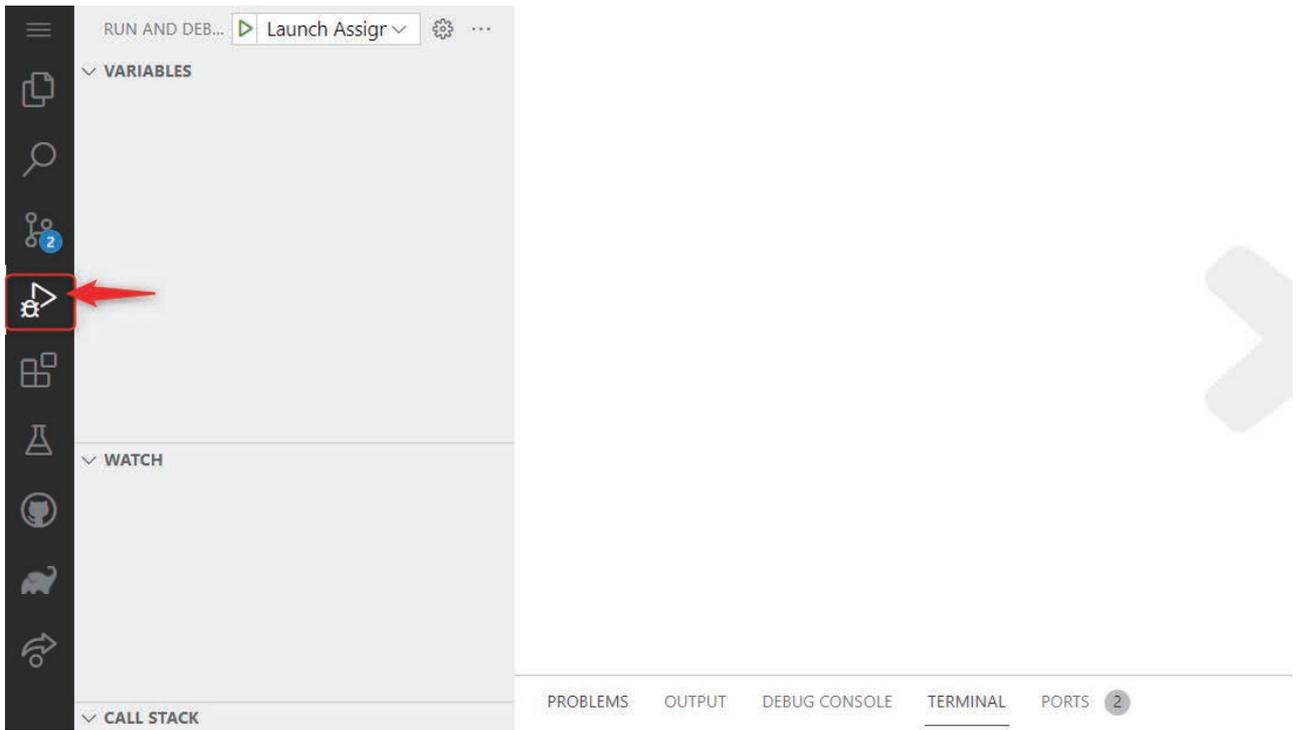
6. インストールボタンを押された後、新しいタブが開きます。緑で囲んだ部分がDisableになったらタブを閉じましょう。



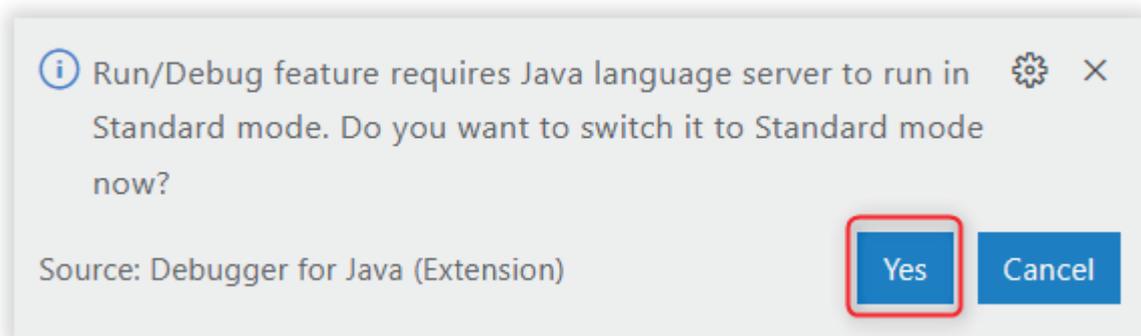
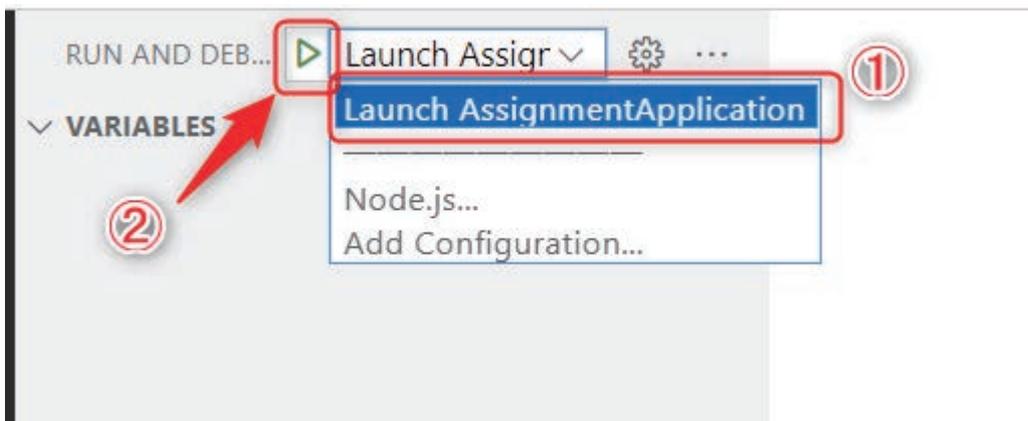
- Get Startedタブも閉じましょう。



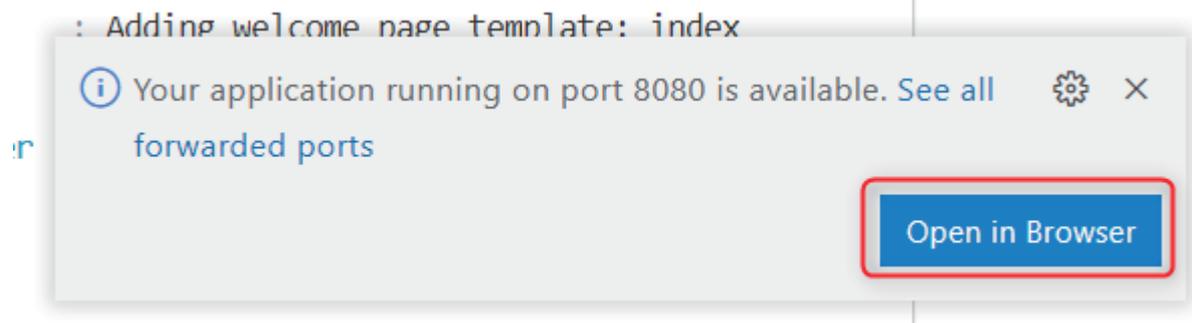
7. 虫と再生ボタンのアイコンを押します。



8. `Launch AssignmentApplication`が選択されていることを確認し再生ボタンを押します。初回はダイアログが表示されます、YESを選択してください。



9. **Open in Browser**を押します。(表示されるまでしばらく時間が掛かります。)



10. Webページが表示されます。

GitHub Classroom
Framework for Spring boot.

Class roomの課題 level1

レベル1 (基礎)

実際に動作するベースプログラムに対して、一部の命令を書き換えることにより、要件定義を満たすプログラム。

STEP - 1

課題の提出手順

課題の提出には以下の作業が必要になります

Commit(コミット)

- Commitは、追加・変更したファイルをGitに登録するための操作です。

Push(プッシュ)

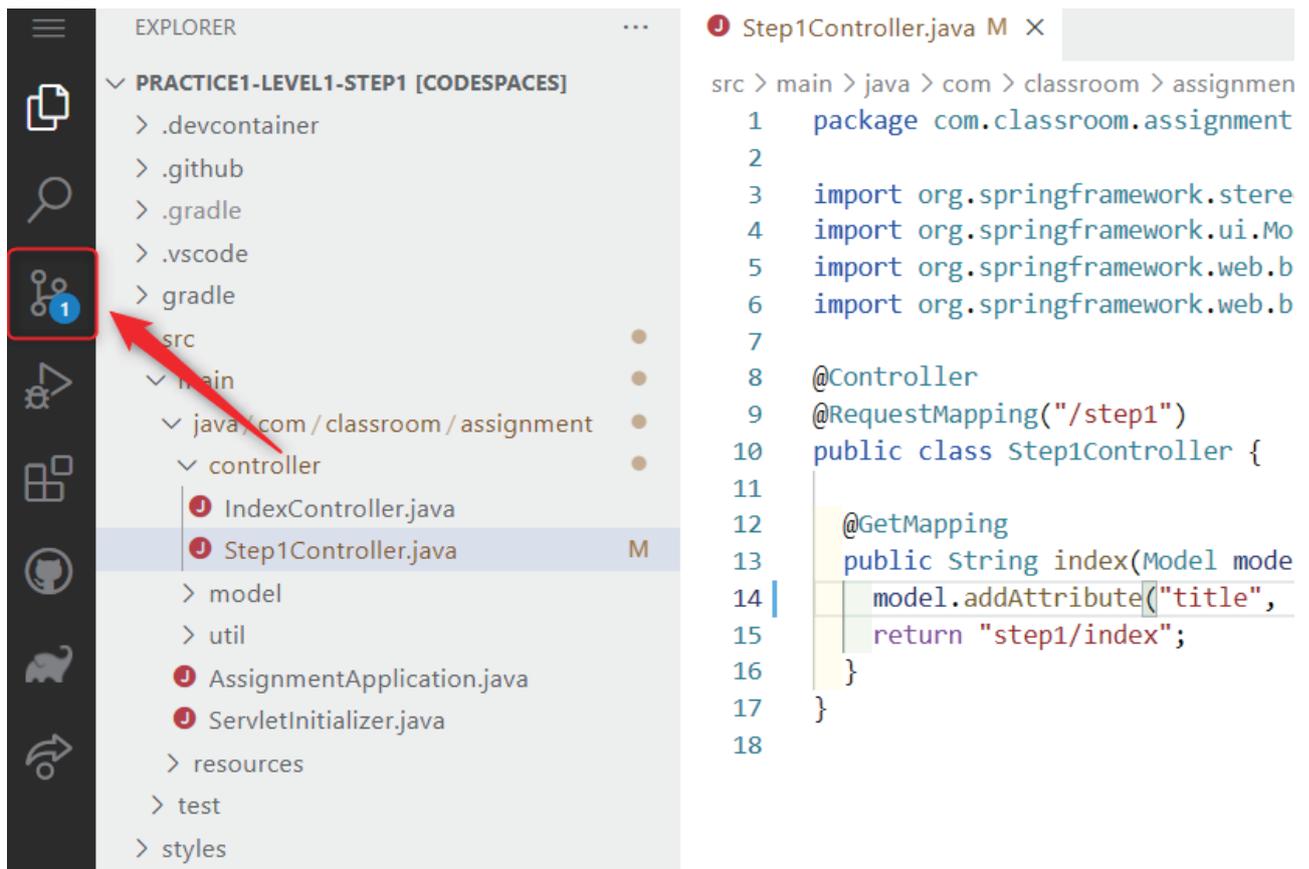
- pushは、コミットした内容をローカルレポジトリからリモートレポジトリに送信する操作です。

Pull Request(プルリクエスト)

- Pull Requestは、開発者のローカルレポジトリでの変更を他の開発者に通知する機能です。

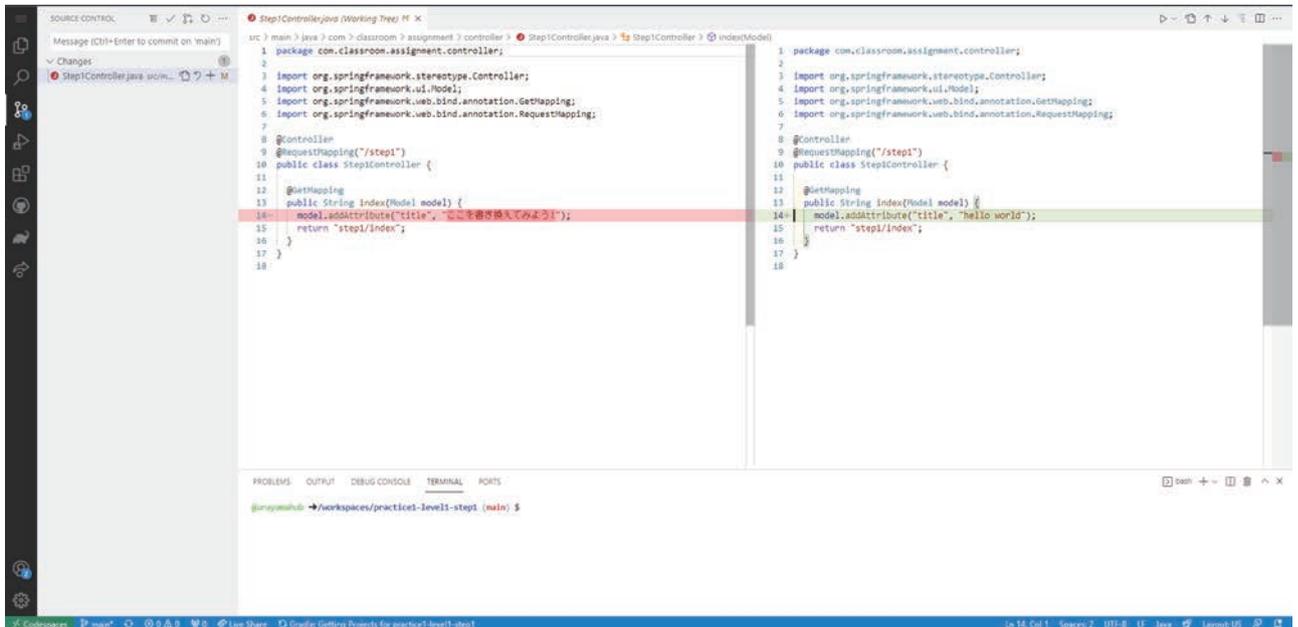
作業手順

1. 木の枝のようなマークを押します。

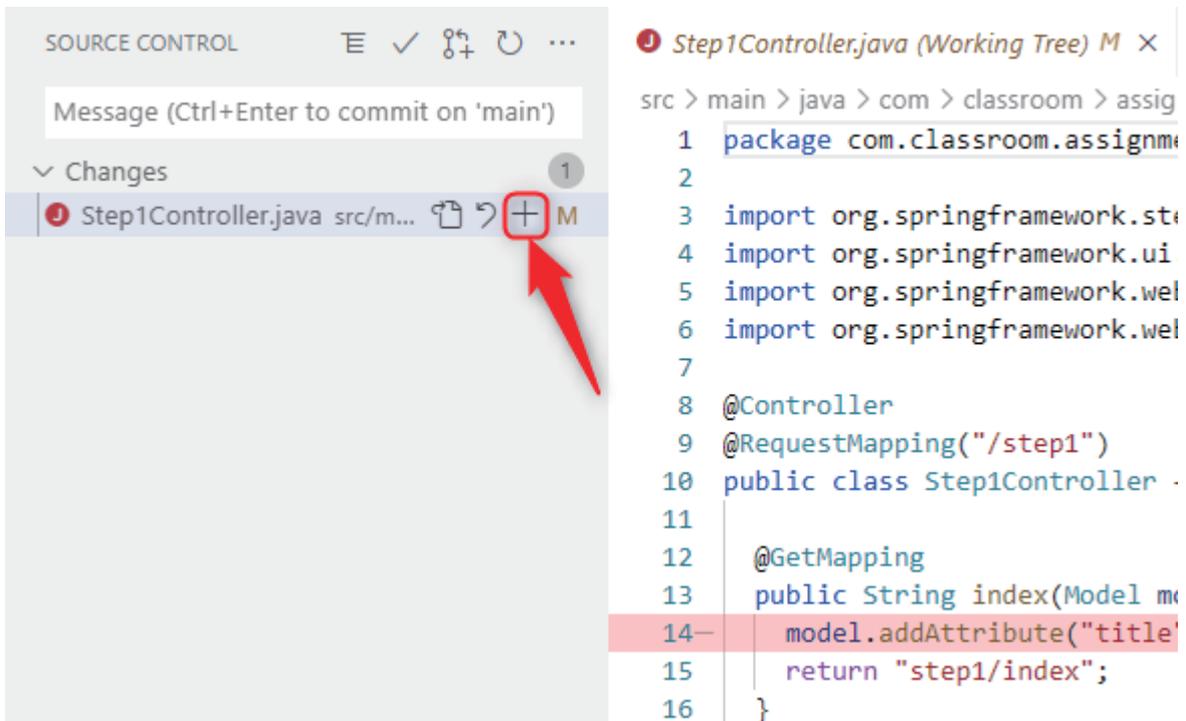


```
src > main > java > com > classroom > assignment
 1  package com.classroom.assignment
 2
 3  import org.springframework.stereotype.Controller;
 4  import org.springframework.ui.Model;
 5  import org.springframework.web.servlet.mvc.annotation.annotation.AnnotationMethodProcessingHandlerMethod;
 6  import org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerMethod;
 7
 8  @Controller
 9  @RequestMapping("/step1")
10  public class Step1Controller {
11
12      @GetMapping
13      public String index(Model model) {
14          model.addAttribute("title", "Step1");
15          return "step1/index";
16      }
17  }
```

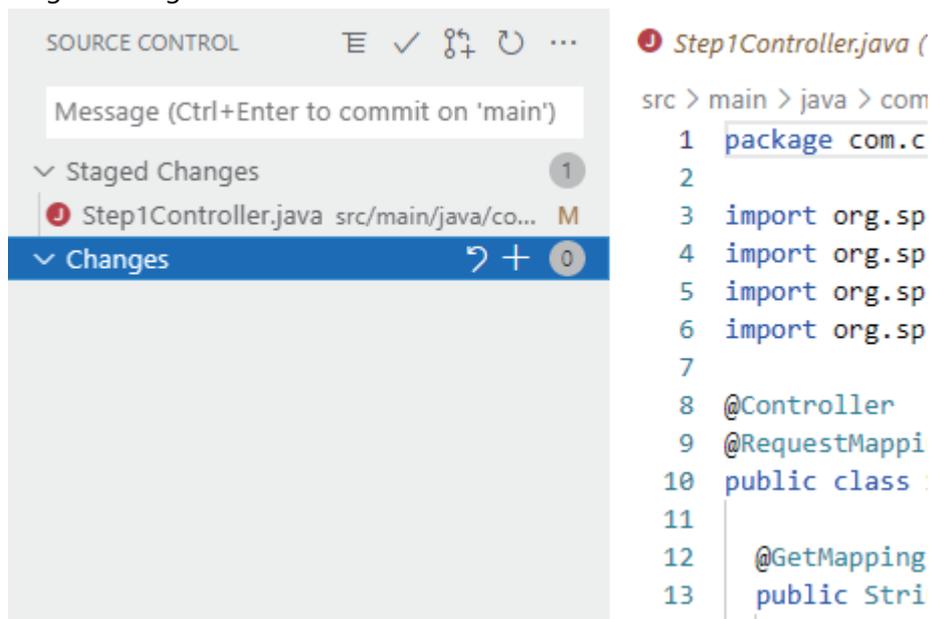
2. これまでに行ったファイルへの変更ないようが表示されています。変更内容に問題がないか今一度確認しましょう。



3. コミットするファイルを選択しましょう。コミットしたいファイルの横の+ボタンを押しましょう。

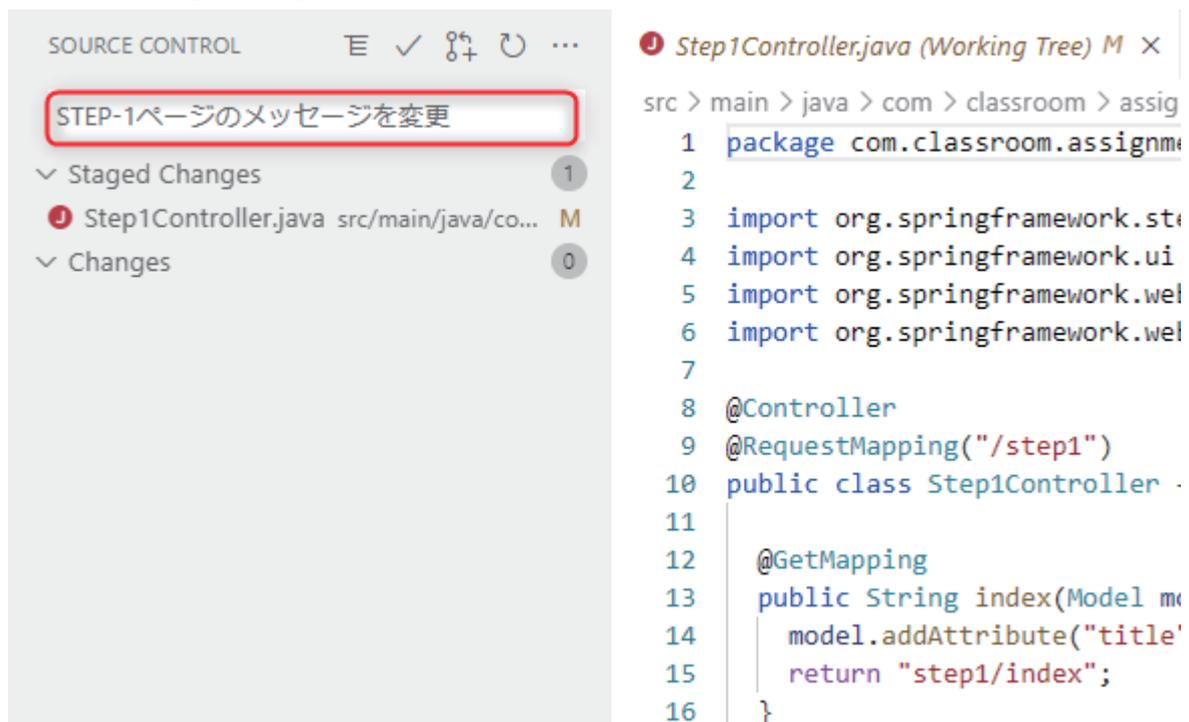


4. Staged Changesにコミットしたいファイルが含まれていることを確認しましょう。

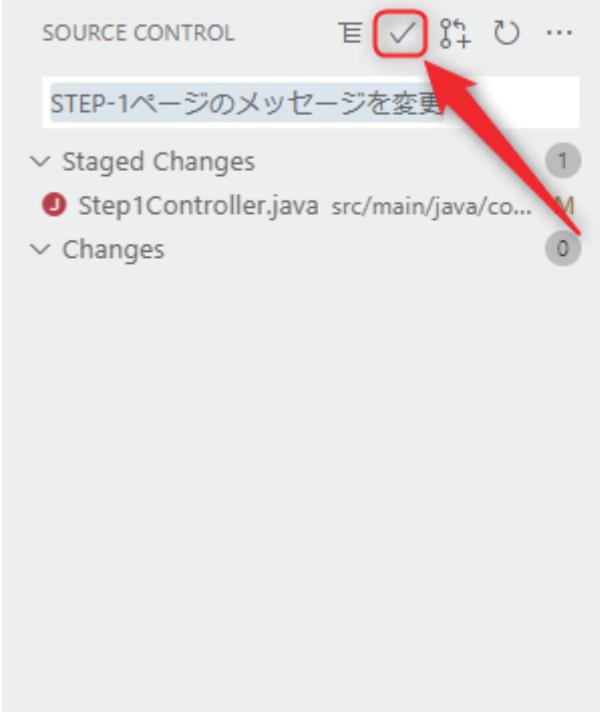


5. Message欄にCommit Message(コミットメッセージ)を入力しましょう。

- Commit Messageには、今回のコミットの追加・変更内容について記載します。
- 課題のテーマを記載しましょう



6. チェックボタンを押してコミットしましょう。



SOURCE CONTROL

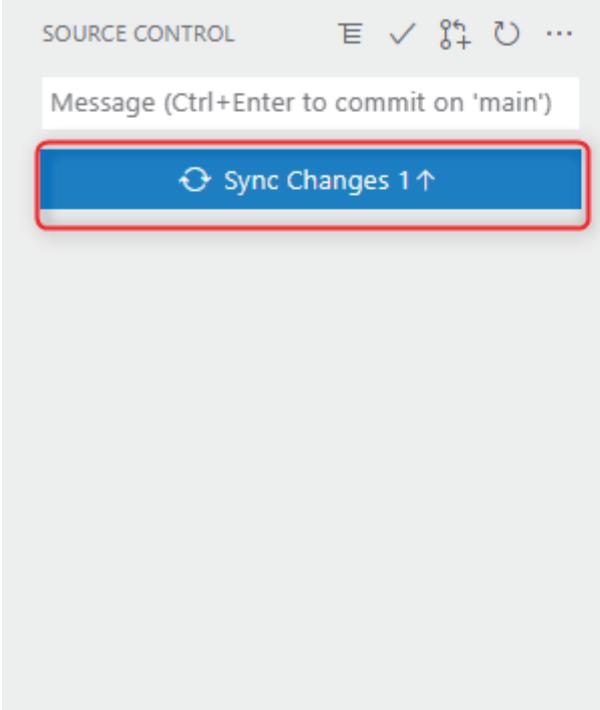
STEP-1ページのメッセージを変更

✓

Step1Controller.java src/main/java/co... M 1

```
src > main > java > com > classroom > assignme
1 package com.classroom.assignme
2
3 import org.springframework.ste
4 import org.springframework.ui
5 import org.springframework.wel
6 import org.springframework.wel
7
8 @Controller
9 @RequestMapping("/step1")
10 public class Step1Controller {
11
12     @GetMapping
13     public String index(Model mo
14         model.addAttribute("title'
15         return "step1/index";
16     }
```

7. Sync Changesボタンを押してコミットをプッシュしましょう。



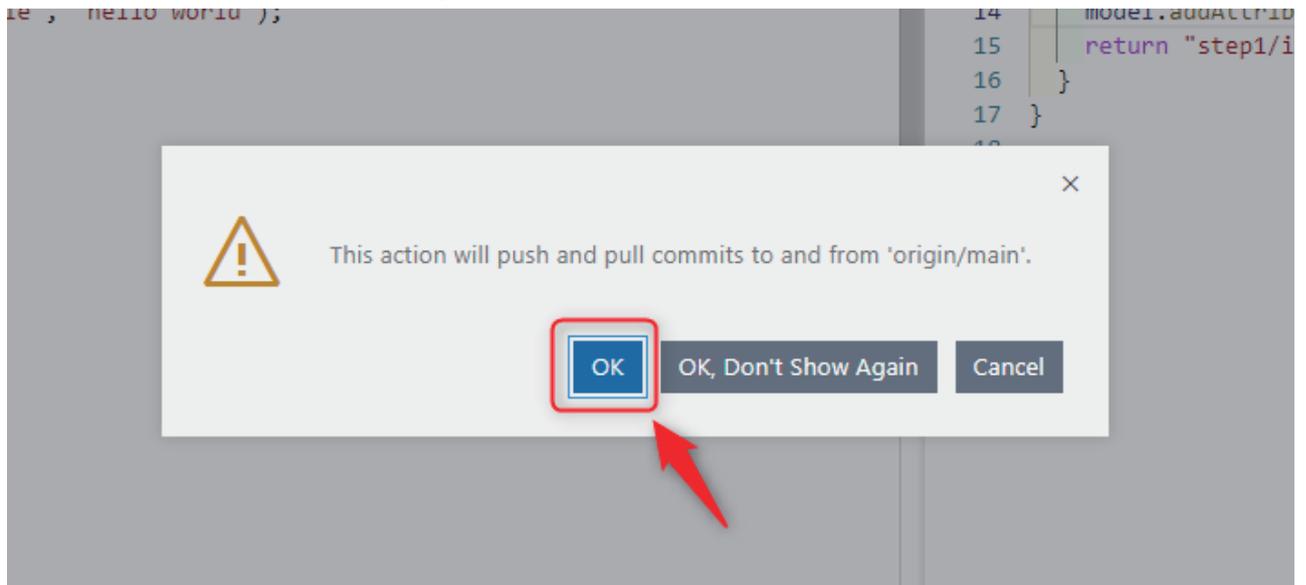
SOURCE CONTROL

Message (Ctrl+Enter to commit on 'main')

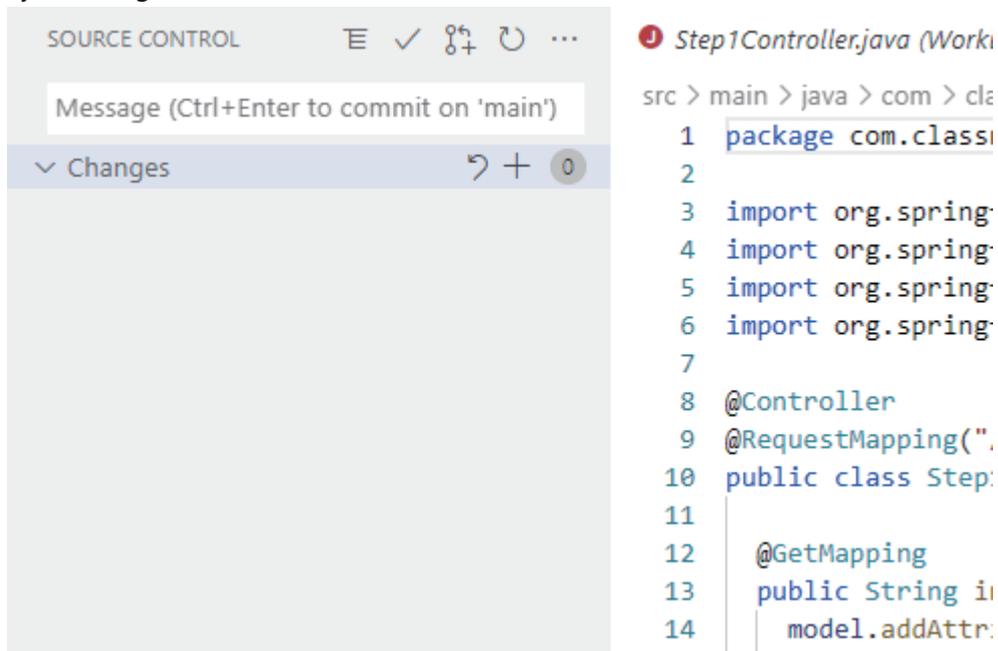
Sync Changes 1 ↑

```
src > main > java > com > classroom > assignme
1 package com.classroom.assignme
2
3 import org.springframework.ste
4 import org.springframework.ui
5 import org.springframework.wel
6 import org.springframework.wel
7
8 @Controller
9 @RequestMapping("/step1")
10 public class Step1Controller {
11
12     @GetMapping
13     public String index(Model mo
14         model.addAttribute("title'
15         return "step1/index";
16     }
```

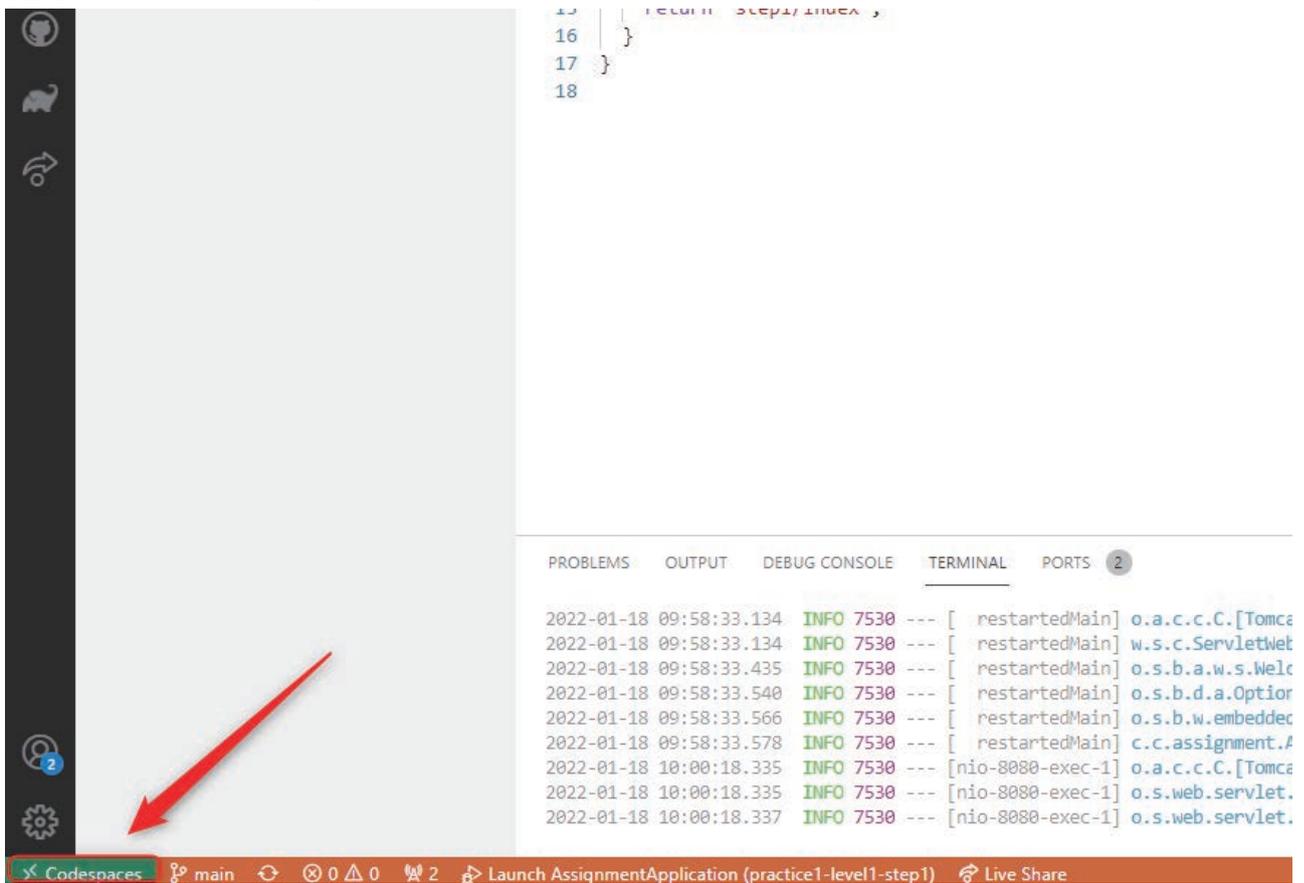
8. ダイアログが出ますので「OK」を押しましょう。



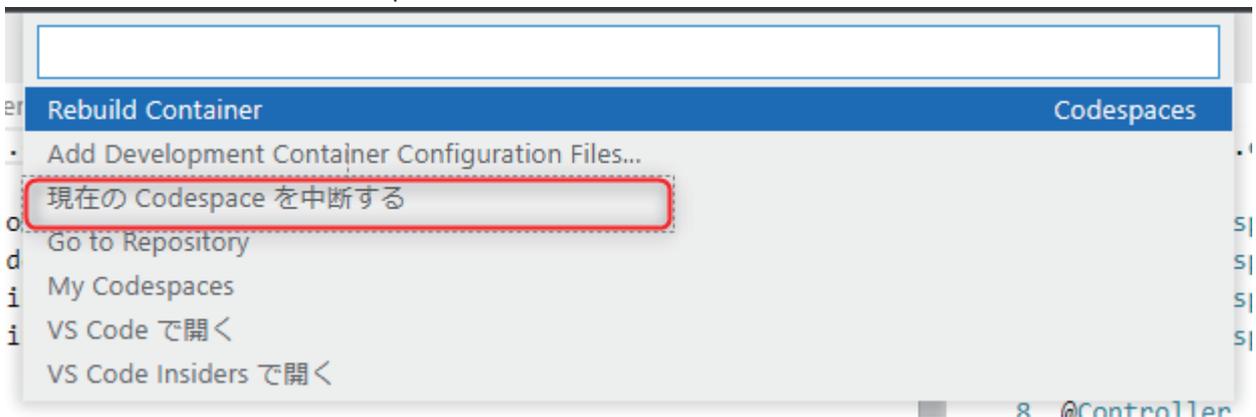
9. Sync Changesボタンが消えていることを確認してください。



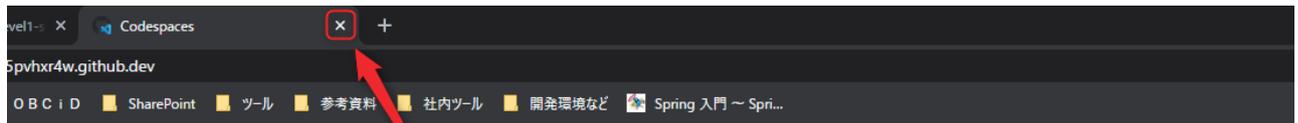
10. 画面左下の緑のボタンを押しましょう。



11. 画面中央上部の「現在のCodespaceを中断する」を押します。



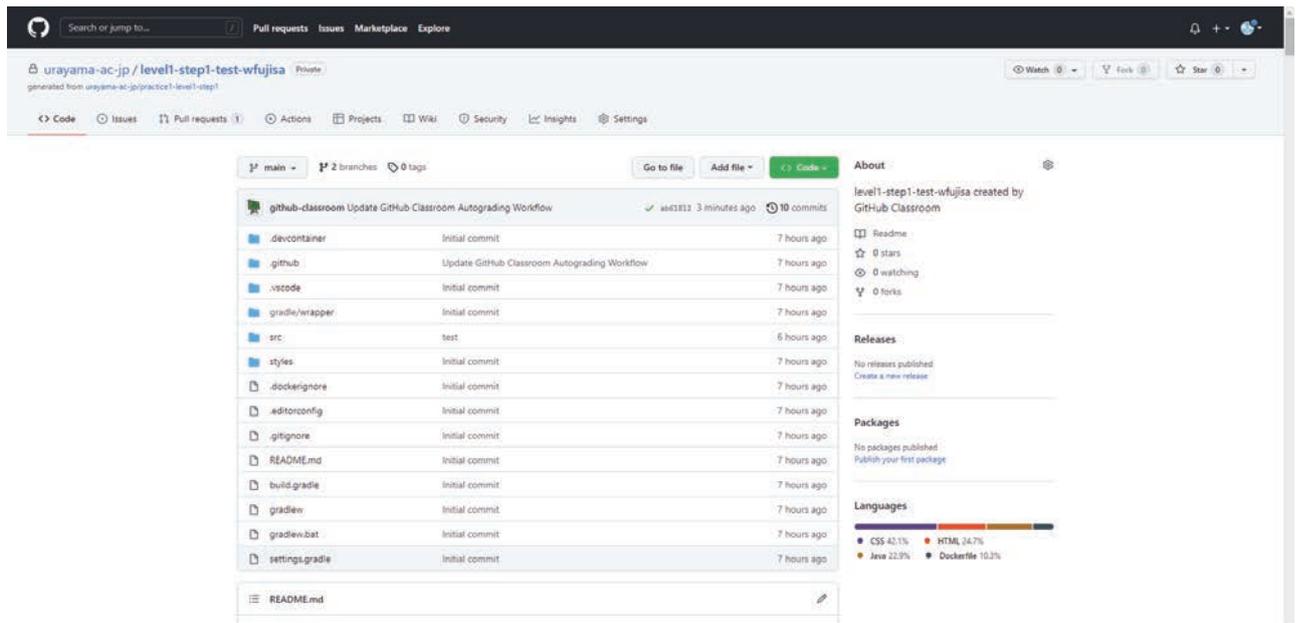
12. タブを×ボタンで閉じましょう。



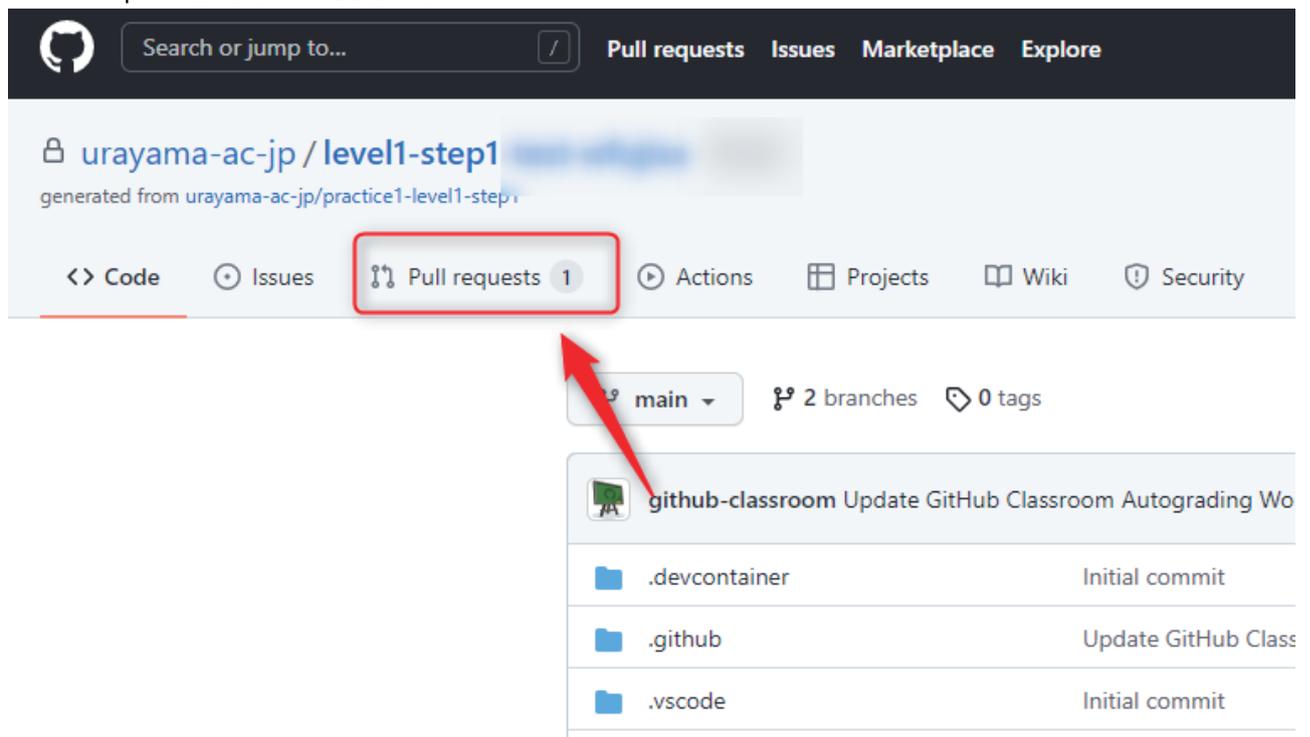
Codespace is stopped

Restart codespace

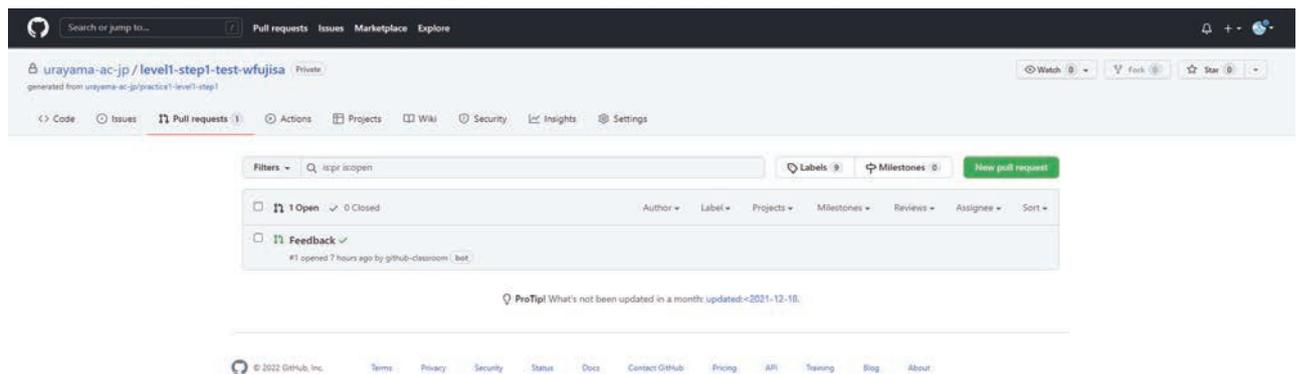
13. 画面コピーの画面に戻ります。



14. 「Pull requests」 ボタンを押します。



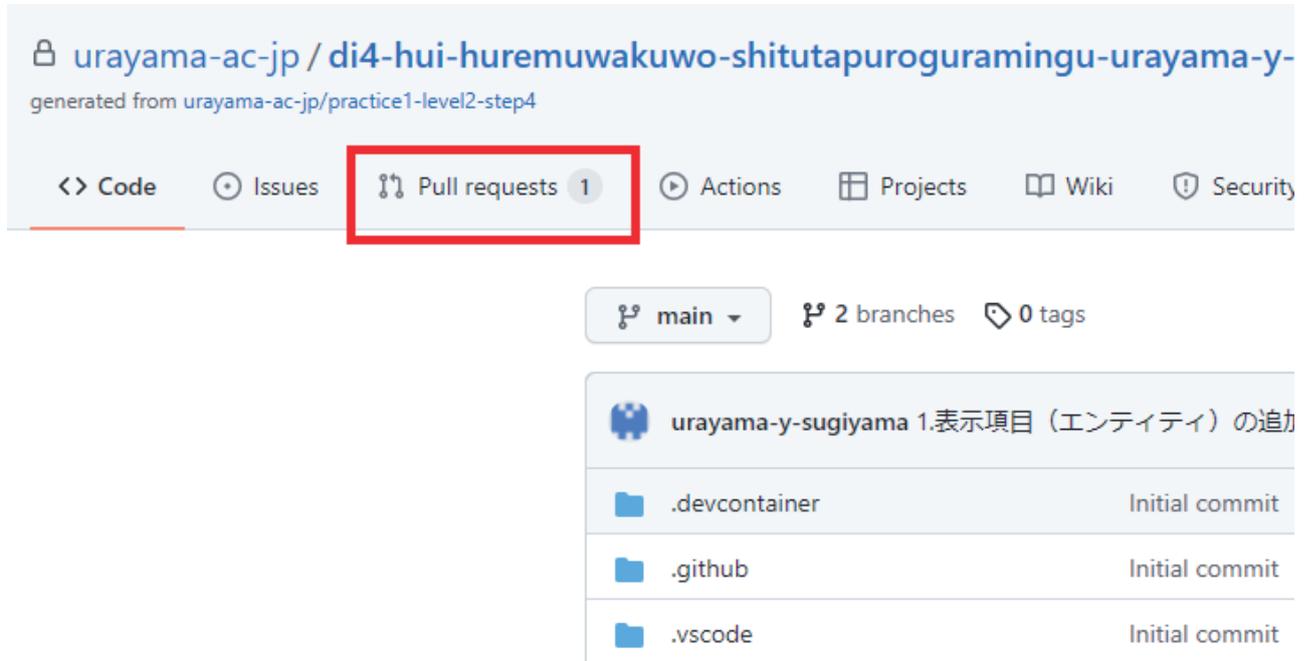
15. 講師からのフィードバックがある場合は、この画面に表示されます。



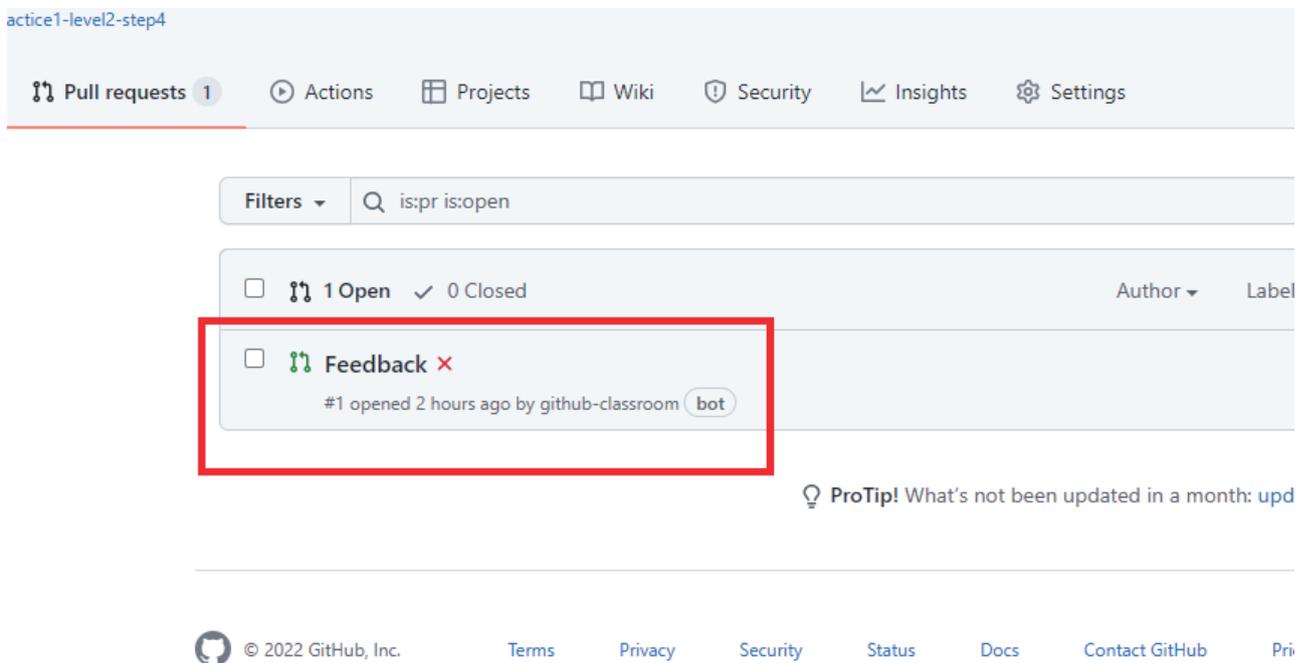
先生からのフィードバックの確認方法

手順

- 課題の画面で赤枠の「Pull requests」を押下



- 赤枠の「Feedback」を押下



- 赤枠内にコメントが書いてあります（画像はモザイクかけてます）

Feedback #1

[Open](#) github-classroom wants to merge 2 commits into `feedback` from `main` [🔗](#)

Conversation 2 Commits 2 Checks 1 Files changed 3

github-classroom bot commented 2 hours ago • edited

🔔! GitHub Classroom created this pull request as a place for your teacher to leave feedback on your work. It will update automatically. **Don't close or merge this pull request**, unless you're instructed to do so by your teacher. In this pull request, your teacher can leave comments and feedback on your code. Click the **Subscribe** button to be notified if that happens. Click the **Files changed** or **Commits** tab to see all of the changes pushed to `main` since the assignment started. Your teacher can see this too.

▶ **Notes for teachers**

Subscribed: @urayama-y-sugiyama

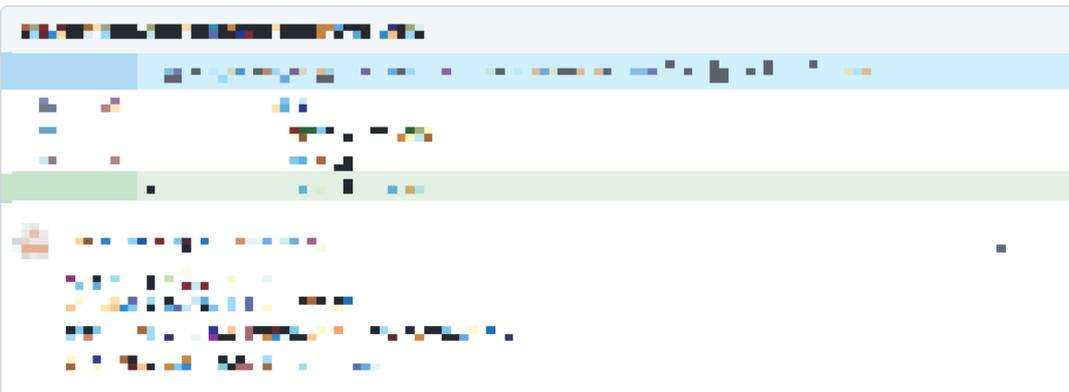
github-classroom bot and others added 2 commits 2 hours ago

- Setting up GitHub Classroom Feedback Verified ✖ d110244
- 1. 表示項目（エンティティ）の追加 ✖ 63942d2

te30473y-magori reviewed 5 minutes ago [View changes](#)

te30473y-magori left a comment

もう一歩です！



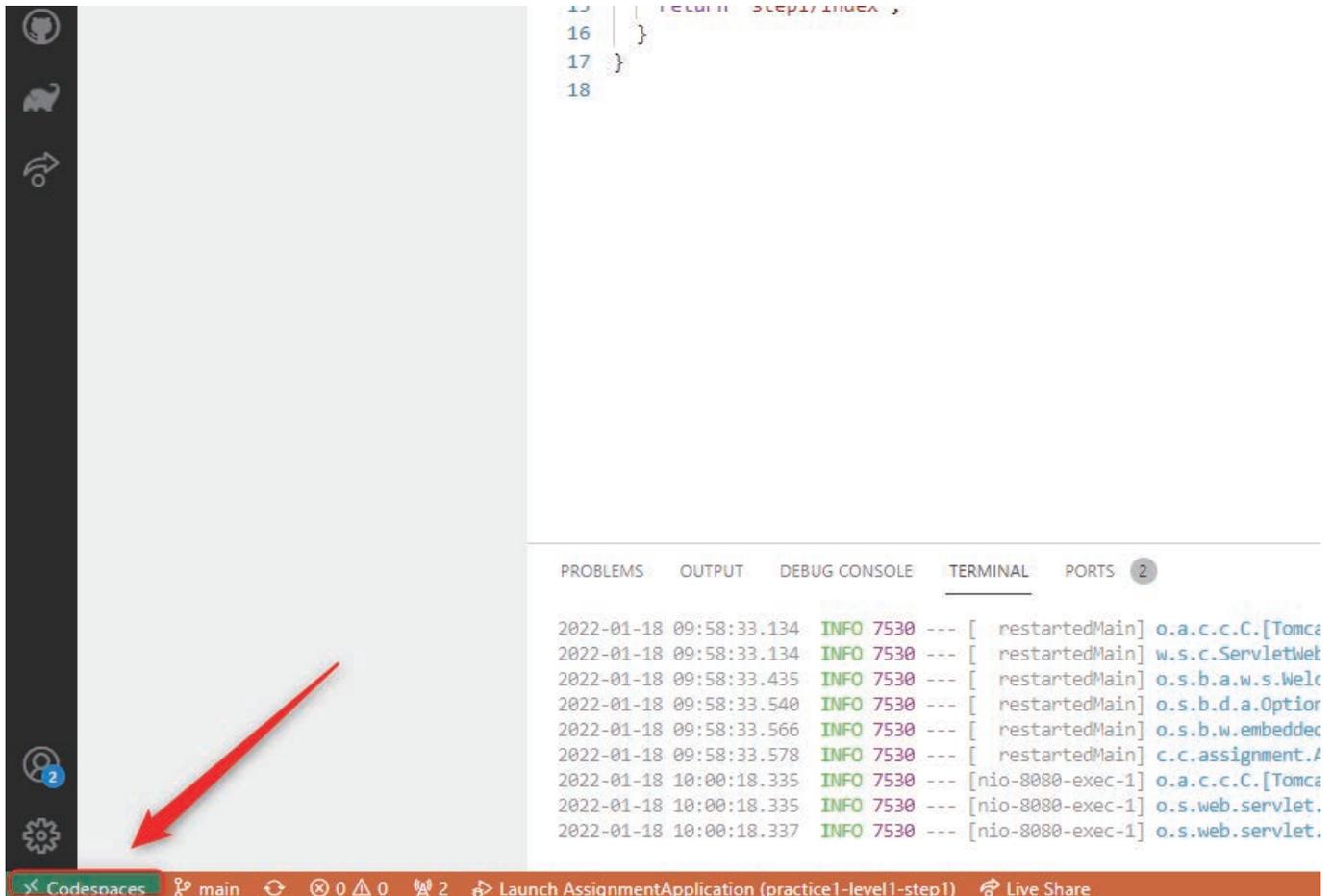
Reply...

Resolve conversation

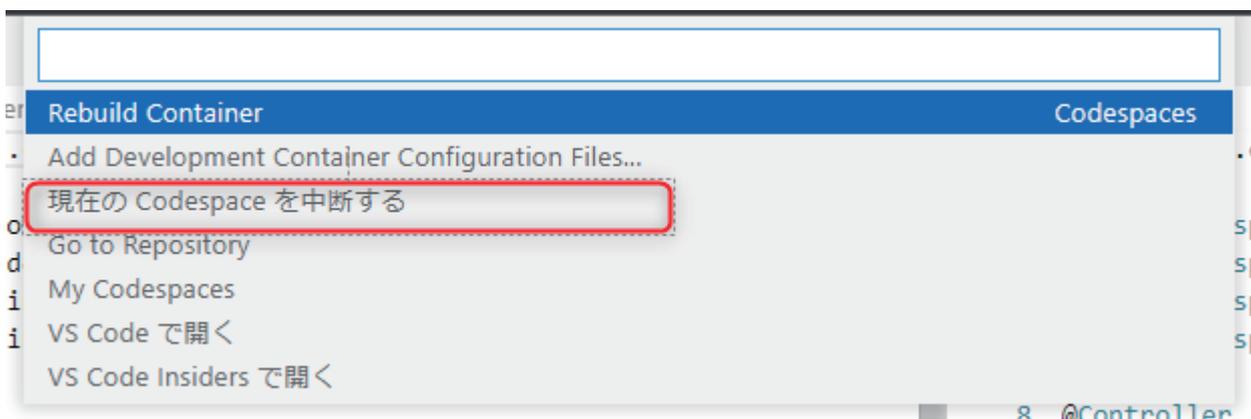
Codespacesの終了方法について

Codespacesはブラウザタブの×ボタンで閉じないようにしてください、以下の手順で終了してください。

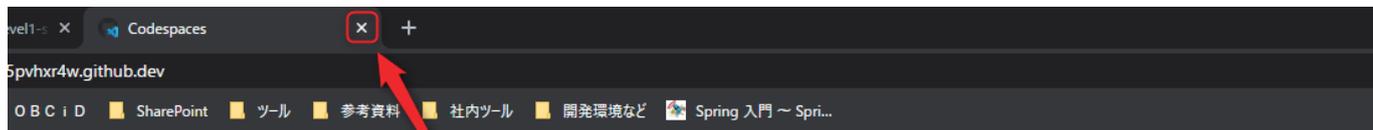
1. 画面左下の緑のボタンを押しましょう。



2. 画面中央上部の「現在のCodespaceを中断する」を押します。



3. タブを×ボタンで閉じましょう。



Codespace is stopped

Restart codespace

演習教材

演習課題とりまとめ

演習課題について

- 遠隔授業として学生には以下の課題を設定
 - フレームワークを用いたプログラミング
 - 実際の業務を意識した課題を設定
 - フレームワークを持ちいたプログラムの課題を配布
 - 配布されたプログラムは基本的に（エラー含んだ状態で）動作するものとし課題に合わせてプログラムの改修を行う
 - プログラムを改修する過程で、フレームワークの動きや機能について学習
 - 動かしてみても動作を確認させる

演習課題について

- レベルは2段階とし、以下を想定
 - レベル1
 - 実際に動作するベースプログラムに対して、一部の命令を書き換えることにより、要件定義を満たすプログラム
 1. フレームワークを利用したプログラムの実行と表示文言の変更
 2. GETパラメータの取得
 3. 画面デザイン (Bootstrapの使用)
 4. バリデーション
 - レベル2
 - 実際に動作するベースプログラムに対して、一部分の処理を参考に追加することにより、要件定義を満たすプログラム
 1. 入力フォームの追加
 2. 表示項目 (エンティティ) の追加
 3. サービスクラスの作成
 4. データベースの利用

レベル1_演習課題1

- テーマ
 - フレームワークを利用したプログラムの実行と表示文言の変更
- 講義内容
 - フレームワークを使った簡単なWEBページの表示方法について
- 課題
 - 画面に表示されるメッセージを変更
- 課題用ファイル一式
 - level1-1.zip

レベル1_演習課題1 (課題の実行画面)

- 「ここを書き換えてみよう!」という文言を「hello world」に変更



レベル1_演習課題1 (解答)

対応前

```
src/main/java/com/classroom/assignment/controller/Step1Controller.java
@@ -12,7 +12,7 @@
12  @GetMapping
13  public String index(Model model) {
14      // 課題 level1 step1
15  -   model.addAttribute("title", "ここを書き換えてみよう!");
16      return "step1/index";
17  }
18  }
```

対応後

```
12  @GetMapping
13  public String index(Model model) {
14      // 課題 level1 step1
15  +   model.addAttribute("title", "hello world");
16      return "step1/index";
17  }
18  }
```

レベル1_演習課題1 (テスト)

- 学生が課題を提出すると以下のテストが自動で実施される
 - 表示された画面に「hello world」が含まれていること

```
@Test
public void shouldReturnDefaultMessage() throws Exception {
    // given
    // when
    // then
    mockMvc.perform(get("/step1")).andExpect(status().isOk())
        .andExpect(content().string(containsString("hello world")));
}
```

レベル1_演習課題2

- テーマ
 - GETパラメータの取得
- 講義内容
 - GETパラメータについて
 - フレームワークを使ったプログラミングでGETパラメータによる振る舞いの制御について
- 課題
 - 画面に表示されるメッセージをGETパラメータの値に応じて表示内容の切り分ける
- 課題用ファイル一式
 - level1-2.zip

レベル1_演習課題2（課題の実行画面）

- GETパラメータで朝の挨拶は正しく動作



レベル1_演習課題2（課題の実行画面）

- 夜の挨拶はエラーになる
- 以下のように正しく動作するよう改修



レベル1_演習課題2 (解答)

対応前

```
src/main/java/com/classroom/assignment/controller/Step2Controller.java
23  @RequestMapping("morning") public String morning(@RequestParam(name = "name", required = false) String name
24  }
25
26  // 課題 level1 step2
27  @GetMapping(" ") // 書き換えてください
28  public String evening(@RequestParam(name = "name", required = false) String name, Model model) {
29  // 「/step2/evening?name=oo」をURLの最後に指定した時に「ooさん、こんばんは!」と返すよう処理を追加してください。
30  // !!!!ここに処理を追加しましょう。!!!!
31
32  // !!!!ここに処理を追加しましょう。!!!!
33  return "step2/greeting";
34  }
```

対応後

```
24  }
25
26  // 課題 level1 step2
27  @GetMapping("/evening")
28  public String evening(@RequestParam(name = "name", required = false) String name, Model model) {
29  // 「/step2/evening?name=oo」をURLの最後に指定した時に「ooさん、こんばんは!」と返すよう処理を追加してください。
30  // !!!!ここに処理を追加しましょう。!!!!
31  model.addAttribute("greeting", name + "さん、こんばんは!");
32  // !!!!ここに処理を追加しましょう。!!!!
33  return "step2/greeting";
34  }
```

レベル1_演習課題2 (テスト)

- 学生が課題を提出すると以下のテストが自動で実施される
 - 夜の挨拶のGETパラメータをリクエスト
 - 表示された画面に「(パラメータ)さん、こんばんは!」が含まれていること

```
@Test
public void shouldReturnEveningGreetingMessage() throws Exception {
    // given
    // when
    // then
    this.mockMvc.perform(get("/step2/evening?name=テスト")).andDo(print()).andExpect(status().isOk())
        .andExpect(content().string(containsString("テストさん、こんばんは!")));
}
```

レベル1_演習課題3

- テーマ
 - 画面デザイン (Bootstrapの使用)
- 講義内容
 - CSSフレームワーク「Bootstrap」を使用してWEBページへの装飾について
- 課題
 - 画面に表示されるテーブルを一行ごとに色を変えて見やすくする
- 課題用ファイル一式
 - level1-3.zip

レベル1_演習課題3 (課題の実行画面)

- テーブルのデザインを1行ずつ色が変わるよう修正



レベル1_演習課題3 (解答)

対応前

```
src/main/resources/templates/step5/index.html
@@ -22,7 +22,7 @@ <h1>STEP-5 WEBページを装飾しよう</h1>
22     <p>テーブルの行を一行ごとに色を変えて見やすくしましょう。</p>
23     <p th:text="${title}">Title</p>
24     <!-- ↓class属性に値を追加しましょう↓ -->
25 -    <table class="table table-dark">
26         <!-- ↑class属性に値を追加しましょう↑ -->
27         <thead>
28             <tr>
```

対応後

```
22     <p>テーブルの行を一行ごとに色を変えて見やすくしましょう。</p>
23     <p th:text="${title}">Title</p>
24     <!-- ↓class属性に値を追加しましょう↓ -->
25 +    <table class="table table-dark table-striped">
26         <!-- ↑class属性に値を追加しましょう↑ -->
27         <thead>
28             <tr>
```

レベル1_演習課題3 (テスト)

- 学生が課題を提出すると以下のテストが自動で実施される
 - 表示される画面のコンテンツに「table-stripe」が含まれていること

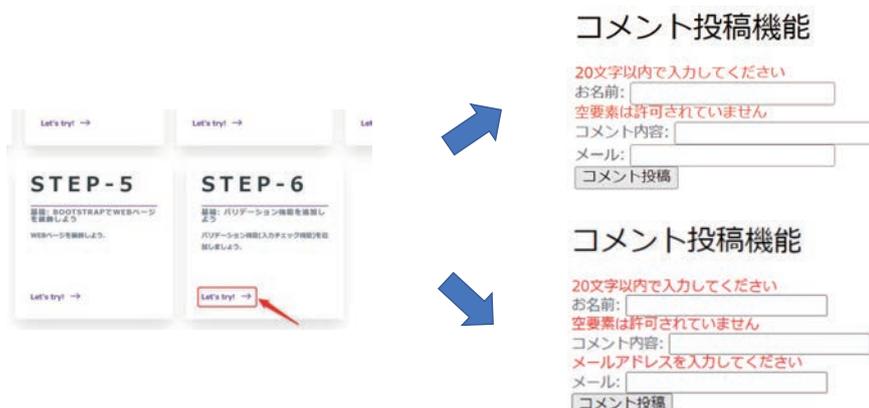
```
@Test
public void classFactorSpecifiedHasConfirm() throws Exception {
    // given
    // when
    // then
    MvcResult result = mockMvc.perform(get("/step5")).andExpect(status().isOk())
        .andReturn();
    String content = result.getResponse().getContentAsString();
    assertThat(content).contains("table-stripe");
}
```

レベル1_演習課題4

- テーマ
 - バリデーション
- 講義内容
 - 入力チェックを行うバリデーション機能について
- 課題
 - メールアドレスに必須チェックのバリデーションを追加する
- 課題用ファイル一式
 - level1-4.zip

レベル1_演習課題4（課題の実行画面）

- メールアドレスを空の場合、エラーメッセージが表示されない
- 以下のようなチェックを行うよう改修



レベル1_演習課題4 (解答)

対応前

```
src/main/java/com/classroom/assignment/model/request/Comment.java
13 private String content;
14
15 // アノテーションを追加してください
16 -
17 // アノテーションを追加してください
18 @Email(message = "メールアドレスの形式ではありません")
19 private String mail;
```

対応後

```
13 private String content;
14
15 // アノテーションを追加してください
16 + @NotEmpty(message = "メールアドレスを入力してください")
17 // アノテーションを追加してください
18 @Email(message = "メールアドレスの形式ではありません")
19 private String mail;
```

レベル1_演習課題4 (テスト)

- 学生が課題を提出すると以下のテストが自動で実施される
 - 名前に「test」、コメントに「test」、メールアドレスに空欄となるリクエストされたとき、エラーメッセージは「メールアドレスを入力してください」が表示されること

```
@Test
public void affiliationDepartmentItemExistenceConfirmation() throws Exception {
    // given
    // when
    // then

    // パラメータ設定
    Comment comment = new Comment("test", "test", "");
    // 実行
    ResultActions results = mockMvc.perform(post("/step6/comment")
        .flashAttr("comment", comment))
        // 検証
        // modelのいずれかにエラーがあること
        .andExpect(model().hasErrors())
        // postFormにエラーが存在すること
        .andExpect(model().attributeHasErrors("comment"));

    BindingResult bindResult = (BindingResult) results.andReturn().getModelAndView().getModel()
        .get(BindingResult.MODEL_KEY_PREFIX + "comment");
    String mes = bindResult.getFieldError().getDefaultMessage();
    assertThat(mes).isEqualTo("メールアドレスを入力してください");
}
```

レベル2_演習課題1

- テーマ
 - フォームの制御
- 講義内容
 - フレームワークを使ったプログラミングのフォームクラスについて
- 課題
 - 画面に新しい入力フォームを追加する
- 課題用ファイル一式
 - level2-1.zip

レベル2_演習課題1（課題の実行画面）

- 入力フォームにメールアドレスを追加する



レベル2_演習課題1 (解答)

対応前

```
src/main/java/com/classroom/assignment/controller/Step3Controller.java
@@ -22,6 +22,7 @@ public String evening(Comment comment, Model model) {
22 model.addAttribute("content", comment.getContent());
23 // 課題 level2 step3
24 // メールアドレスの項目を追加してください。

25 return "step3/confirm";
26 }
27 }
```

対応後

```
22 model.addAttribute("content", comment.getContent());
23 // 課題 level2 step3
24 // メールアドレスの項目を追加してください。
25 + model.addAttribute("mail", comment.getMail());
26 return "step3/confirm";
27 }
28 }
```

レベル2_演習課題1 (解答)

対応前

```
src/main/java/com/classroom/assignment/model/request/Comment.java
@@ -4,10 +4,12 @@
4 private String name;
5 private String content;
6
7
8 + public Comment(String name, String content) {
9 +     this.setName(name);
10 +     this.setContent(content);
11 + }
12
13 public String getName() {
14
15 + @@ -25,4 +27,12 @@ public String getContent() {
16 +     public void setContent(String content) {
17 +         this.content = content;
18 +     }
19
20 }
```

対応後

```
4
5 private String name;
6 private String content;
7 - private String mail;
8
9 + public Comment(String name, String content,String address) {
10 +     this.setName(name);
11 +     this.setContent(content);
12 +     this.setMail(mail);
13 + }
14
15 public String getName() {
16
17
18
19
20 public void setContent(String content) {
21     this.content = content;
22 }
23
24
25
26
27 + public String getMail() {
28 +     return mail;
29 + }
30
31
32 + public void setMail(String mail) {
33 +     this.mail = mail;
34 + }
35
36 }
```

レベル2_演習課題1 (解答)

対応前

```
src/main/resources/templates/step3/confirm.html
@@ -22,6 +22,7 @@
22 <h1 class="information-head">コメント投稿確認 確認画面</h1>
23 <p>名前:{{name}}</p>
24 <p>コメント内容:{{content}}</p>
25 <a href="javascript:history.back();">戻る</a>
26 </div>
27 </div>

src/main/resources/templates/step3/index.html
@@ -25,6 +25,8 @@ <h1 class="information-head">コメント投稿確認</h1>
25 <input id="name" name="name" type="text"><br>
26 <label for="content">コメント内容:</label>
27 <input id="content" name="content" type="text"><br>
28 <input type="submit" value="コメント投稿">
29 </form>
30 </div>
```

対応後

```
22 <h1 class="information-head">コメント投稿確認 確認画面</h1>
23 <p>名前:{{name}}</p>
24 <p>コメント内容:{{content}}</p>
25 + <p>メール:{{mail}}</p>
26 <a href="javascript:history.back();">戻る</a>
27 </div>
28 </div>

25 <input id="name" name="name" type="text"><br>
26 <label for="content">コメント内容:</label>
27 <input id="content" name="content" type="text"><br>
28 + <label for="mail">メール:</label>
29 + <input id="mail" name="mail" type="text"><br>
30 <input type="submit" value="コメント投稿">
31 </form>
32 </div>
```

レベル2_演習課題1 (テスト)

- 学生が課題を提出すると以下のテストが自動で実施される
 - 追加したフォーム（メールアドレス）に値を設定しリクエスト
 - 表示された画面にフォームに設定した内容が含まれていること

```
@Test
public void mailItemFormExistenceConfirmation() throws Exception {
    // given
    // when
    // then
    this.mockMvc.perform(
        MockMvcRequestBuilders.post("/step3/comment").param("mail", "test_mail").andDo(print())
            .andExpect(status().isOk())
            .andExpect(content().string(containsString("test_mail")));
    }
}
```

レベル2_演習課題2

- テーマ
 - エンティティの追加
- 講義内容
 - エンティティクラスについて
 - フレームワークを使ったプログラミングのエンティティクラスについて
- 課題
 - 画面の表示項目を増やす
※今回はDBを用いずにController内で疑似的にデータを作成
- 課題用ファイル式
 - level2-2.zip

レベル2_演習課題2（課題の実行画面）

- 所属学科を追加する

Class roomの課題

STEP-2 STEP-3 STEP-4

STEP-4 学生一覧を表示しよう

レベル4 (初級)

学籍番号	名前
321A0001	山田 太郎
321A0002	田中 一郎
321A0003	佐藤 次郎

STEP-4 学生一覧を表示しよう

レベル4 (初級)

学籍番号	名前	所属学科
321A0001	山田 太郎	プログラミング専攻
321A0002	田中 一郎	ロボット・IoT専攻
321A0003	佐藤 次郎	クラウド専攻

レベル2_演習課題2 (解答)

対応前

```
src/main/java/com/classroom/assignment/controller/Step4Controller.java
@@ -17,9 +17,9 @@
17 public String index(Comment comment, Model model) {
18     List<Student> studentList = new ArrayList<Student>();
19
20 - studentList.add(new Student("321A0001", "山田 太郎"));
21 - studentList.add(new Student("321A0002", "田中 一郎"));
22 - studentList.add(new Student("321A0003", "佐藤 次郎"));
23 // 課題 level12 step4
24 // 所属学科を追加してください。
25
.....
↓
```

対応後

```
17 public String index(Comment comment, Model model) {
18     List<Student> studentList = new ArrayList<Student>();
19
20 + studentList.add(new Student("321A0001", "山田 太郎", "プログラミング専攻"));
21 + studentList.add(new Student("321A0002", "田中 一郎", "ロボット・IoT専攻"));
22 + studentList.add(new Student("321A0003", "佐藤 次郎", "クラウド専攻"));
23 // 課題 level12 step4
24 // 所属学科を追加してください。
25
```

レベル2_演習課題2 (解答)

対応前

```
src/main/java/com/classroom/assignment/model/request/Student.java
@@ -3,10 +3,12 @@
3 public class Student {
4     private String id;
5     private String name;
6
7 - public Student(String id, String name) {
8     this.setId(id);
9     this.setName(name);
10 }
11
12 public String getName() {
13     @@ -25,5 +27,14 @@ public void setto(String id) {
14         this.id = id;
15     }
16 }
17
18
19 }
```

対応後

```
3 public class Student {
4     private String id;
5     private String name;
6 + private String subjects;
7
8 = public Student(String id, String name, String subject) {
9     this.setId(id);
10    this.setName(name);
11 +    this.setSubject(subject);
12 }
13
14 public String getName() {
15
16     this.id = id;
17 }
18
19
20 = public String getSubject() {
21     return subjects;
22 }
23
24 + public void setSubject(String subject) {
25     this.subject = subject;
26 }
27
28 =
29
30 }
```

レベル2_演習課題2 (解答)

対応前

```
src/main/resources/templates/step4/index.html
@@ -26,10 +26,12 @@ <h1 class="information-head">STEP-4 学生一覧を表示しよう</h1>
26     <tr>
27         <th>学籍番号</th>
28         <th>名前</th>
29     </tr>
30     <tr th:each="student : ${studentList}">
31         <td th:text="${student.id}"></td>
32         <td th:text="${student.name}"></td>
33     </tr>
34 </table>
35 </div>
```

対応後

```
26     <tr>
27         <th>学籍番号</th>
28         <th>名前</th>
29         + <th>所属学科</th>
30     </tr>
31     <tr th:each="student : ${studentList}">
32         <td th:text="${student.id}"></td>
33         <td th:text="${student.name}"></td>
34         + <td th:text="${student.subject}"></td>
35     </tr>
36 </table>
37 </div>
```

レベル2_演習課題2 (テスト)

- 学生が課題を提出すると以下のテストが自動で実施される
 - 追加した項目 (所属学科) が表示されることを確認
※本来はDB値との突合せを行う

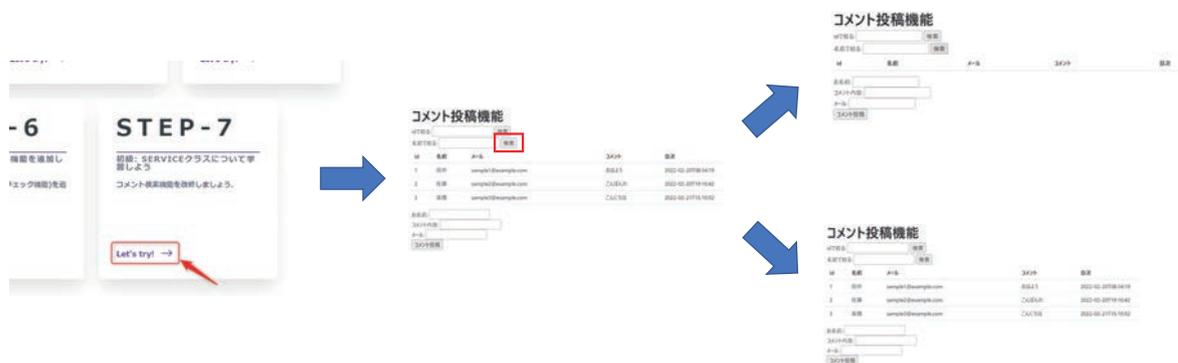
```
@Test
public void affiliationDepartmentItemExistenceConfirmation() throws Exception {
    // given
    // when
    // then
    this.mockMvc.perform(get("/step4")).andDo(print()).andExpect(status().isOk())
        .andExpect(content().string(containsString("所属学科")));
}
```

レベル2_演習課題3

- テーマ
 - サービスクラスの作成
- 講義内容
 - フレームワークを使ったプログラミングのサービスクラスについて
- 課題
 - 検索欄に何も入れずに検索した場合、全件表示されるよう、Serviceに処理を追加する
- 課題用ファイル一式
 - level2-3.zip

レベル2_演習課題3（課題の実行画面）

- 名前で絞るの検索欄に何も入れずに検索した場合、全件表示されるよう改修



レベル2_演習課題3 (解答)

対応前

```
src/main/java/com/classroom/assignment/service/commentServiceImpl.java
@@ -29,7 +29,12 @@ public void save(Comment comment) {
29  @Override
30  public List<Comment> getCommentSerchByName(String name) {
31  // STEP7 検索欄に何も入れずに検索した場合、全件検索されるように分岐処理を追加して
    <-- ください。
32  - List<Comment> list = dao.getCommentSerchByName(name);

33  return list;
34  }
35
```

対応後

```
@Override
30 public List<Comment> getCommentSerchByName(String name) {
31 // STEP7 検索欄に何も入れずに検索した場合、全件検索されるように分岐処理を追加して
    <-- ください。
32 + List<Comment> list = new ArrayList<Comment>();
33 + if (name.isEmpty()) {
34 + list = dao.getAll();
35 + } else {
36 + list = dao.getCommentSerchByName(name);
37 + }
38 return list;
39 }
40
```

レベル2_演習課題3 (テスト)

- 学生が課題を提出すると以下のテストが自動で実施される
 - 名前で絞るのフォームに空文字を設定しリクエスト
 - 表示された画面に事前に登録してある名前（田中、佐藤、高橋）が含まれていること

```
@Test
public void allSearchOperationCheck() throws Exception {
// given
// when
// then
this.mockMvc.perform(get("/step7/search-name").param("searchName", "").andDo(print()))
.andExpect(status().isOk())
.andExpect(content().string(containsString("田中")))
.andExpect(content().string(containsString("佐藤")))
.andExpect(content().string(containsString("高橋")));
}
```

レベル2_演習課題4

- テーマ
 - データベースの利用
- 講義内容
 - フレームワークを使ったプログラミングのDAOクラスについて
- 課題
 - データの削除機能を追加する
- 課題用ファイル一式
 - level2-4.zip

レベル2_演習課題4（課題の実行画面）

- idを入力し削除ボタンを押下すると対象のデータが削除されるよう改修する

STEP-8

初級: DAOクラスについて学習しよう

コメント削除機能を追加しましょう。

Let's try! →

コメント投稿機能

名前で検索:

id	名前	メール	コメント	日時
1	田中	sample1@example.com	おはよう	2022-02-20T08:34:19
2	佐藤	sample2@example.com	こんばんは	2022-02-20T19:10:42
3	高橋	sample3@example.com	こんばんは	2022-02-21T15:10:52

名前:

コメント内容:

メール:

削除するコメントのidを入力してください:

コメント削除

削除するコメントのidを入力してください:

レベル2_演習課題4 (解答)

対応前

```
src/main/java/com/classroom/assignment/repository/CommentDaoImpl.java
@@ -65,6 +65,6 @@ public void insertComment(Comment comment) {
65     public void deleteComment(int id) {
66         // STEP-8 削除処理のSQLを追加しましょう。
67         // 削除も「jdbcTemplate.update」を使ってください。
68     }
69 }
70 }
```

対応後

```
65     public void deleteComment(int id) {
66         // STEP-8 削除処理のSQLを追加しましょう。
67         // 削除も「jdbcTemplate.update」を使ってください。
68         jdbcTemplate.update("DELETE from comment where id = ?", id);
69     }
70 }
```

レベル2_演習課題4 (テスト)

- 学生が課題を提出すると以下のテストが自動で実施される
 - 削除対象のデータが存在することを確認
 - 削除処理を行って対象のデータが非表示 (削除) になること確認

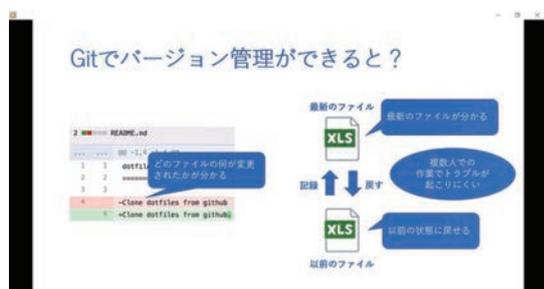
```
@Test
public void commentDeletingFunctionCheck() throws Exception {
    // given
    // when
    // then

    // 実行
    // 削除対象のコメントが存在していることを確認する。
    this.mockMvc.perform(get("/step8"))
        .andExpect(content().string(containsString("田中")));

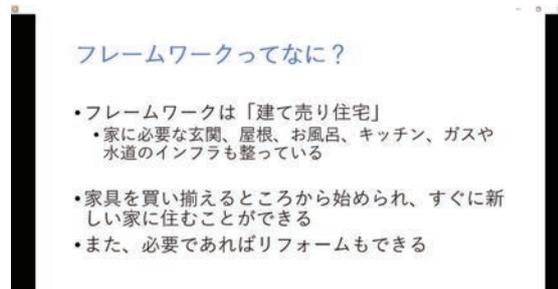
    // コメントを一件削除する。
    this.mockMvc.perform(post("/step8/delete").param("deleteId", "1"))
        .andExpect(status().is(302));

    // コメントが削除され、コメント件数が3件⇒2件になっていることを確認する。
    assertEquals(2, JdbcTestUtils.countRowsInTable(jdbcTemplate, "comment"));
}
```

動画



プラットフォーム利用動画



スピード開発学習動画

遠隔教育プラットフォーム
を使った遠隔授業の成果検証報告

教育プラットフォームを使った 遠隔授業の成果検証報告

本検証の目的

- 実践的なIT教育を支えるプログラミング演習・実習授業を遠隔で行える教育プラットフォームを構築
- そのプラットフォームを利用した遠隔での学習と対面での学習に同等以上の教育効果があるかを検証する
- またプラットフォームの使いやすさ・分かり易さについても評価を行う

検証期間と対象

- 日時
 - 対面 : 2/8 9:00 - 12:10 (3時間)
 - 遠隔 : 2/7 ~ 2/10 (4日間)
- 対象者
 - 富山情報ビジネス専門学校 1年情報システム学科の学生 45名
 - その内、22名を対面での学習者、23名をプラットフォーム(遠隔)による学習者とした
 - 学生はプラットフォームで使用するツール (GitHub、Codespaces、Slack) 、および学習課題となるフレームワーク (Spring) の利用経験はなし
- 事前作業
 - 遠隔のみ事前にプラットフォームの利用に必要な作業を実施
 - GitHubのアカウント作成とエデュケーション申請
 - Slackのインストールとサインイン

プラットフォームを利用した学習効果の検証

検証方法

- 学習効果の検証にあたり、フレームワークを用いたプログラミングを題材とした授業（遠隔では、同等の説明資料を提供）の後、課題を提出してもらう
- 実施後に以下を収集し、その結果をもとに検証を行う
 - 収集内容
 - 対面・遠隔双方の参加者にフレームワークを用いたプログラミングに関するアンケート
 - アンケートでは、フレームワークを用いたプログラミングについて理解できなかった、あまり理解できなかった、理解できなかった、使えなかった、使いこなせたの5段階で評価してもらい、良かった点と改善点について回答してもらう
 - 対面・遠隔両方のプログラム、テスト結果、不良件数、内容
 - 対面・遠隔両方の開発速度（時間）
- 課題はレベルの異なる課題(以下)を4本用意し、対面・遠隔ともに同等なものを与える
 - レベル1（2本）
 - 実際に動作するベースプログラムに対して、一部の命令を書き換えることにより、要件定義を満たすプログラム
 - レベル2（2本）
 - 実際に動作するベースプログラムに対して、一部分の処理を参考に追加することにより、要件定義を満たすプログラム

検証結果

- 課題ごとの完了数と品質の比較

形式	対象者	課題	完了		未完了		未着手
			バグなし	バグ修正後再提出	バグあり	未提出	
遠隔	22人(※)	課題1(レベル1)	20人	0人	0人	2人	0人
		課題2(レベル1)	18人	0人	1人	3人	0人
		課題3(レベル2)	12人	0人	1人	9人	0人
		課題4(レベル2)	10人	1人	0人	11人	0人
対面	22人	課題1(レベル1)	21人	0人	0人	1人	0人
		課題2(レベル1)	19人	0人	0人	3人	0人
		課題3(レベル2)	8人	0人	0人	14人	0人
		課題4(レベル2)	0人	0人	0人	0人	22人

※ 23名中、1名がPCトラブルにより不参加となったため対象外とし対象者は22名

検証結果

• 課題ごとの完了までに要した時間の比較

形式	課題	課題の完了者	課題の開始から完了（提出）になるまでに要した時間					
			～30分未満	30分～1時間未満	1時間～2時間未満	2時間～4時間未満	4時間～8時間未満	8時間以上
遠隔	課題1(レベル1)	20人	3人	7人	1人	4人	1人	4人
	課題2(レベル1)	18人	5人	2人	3人	4人	0人	4人
	課題3(レベル2)	12人	3人	7人	2人	0人	0人	0人
	課題4(レベル2)	11人	7人	1人	1人	0人	1人	1人
対面	課題1(レベル1)	21人	21人	0人	0人	0人	0人	0人
	課題2(レベル1)	19人	19人	0人	0人	0人	0人	0人
	課題3(レベル2)	8人	※1					
	課題4(レベル2)		※2					

※1 授業時間内で説明だけ行い、課題は宿題としたため計測不可

※2 授業時間内では時間が足りず、未実施

考察

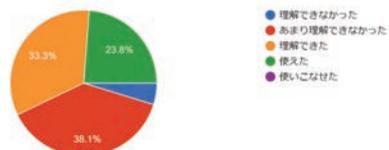
- 対面では、授業内でレベル1の課題1、2とレベル2の課題3の説明までしか行えなかった。そのため、課題3については有志による宿題とし、課題4については未実施となった。
- レベル1の課題においては、課題1の完了者が対面では21/22人で遠隔が20/22人、課題2の完了者が対面では19/22人で遠隔18/22人となり、ほぼ同等の結果が得られた。レベル2の課題3の完了者が対面では8/22人で遠隔が12/22人、課題4の完了者は遠隔が11/22人（対面は未実施）だった。レベル2は、レベル1に比べ完了者が減少する傾向が見られ、対面の学生から「3問目の難易度が急に上がって難しかったです」という意見もあった。課題3の完了者数を比較すると遠隔のほうが多い結果となった。これは、対面は宿題だったことで先生のフォローはなし、遠隔はSlackでのフォローがあった差と見えるが、遠隔の学生から課題3に関する問い合わせは特になかったため、フォロー有無は影響しなかったと考えられる。また対面では課題3の提出を有志としたため完了者が減ったと思われるが、それを鑑みても遠隔での授業は対面と同等の成果があったと思われる。
- 遠隔と対面で課題完了までの時間を比較すると、対面は課題1、2ともに30分以内での完了（提出）に至ったが、遠隔では30分以内で完了できた学生は、課題1で3人、課題2で5人のみだった。このことから対面に比べ遠隔のほうが課題に時間を有したことが分かる。この要因としては、以下が挙げられる。
 1. 学生が課題に対してしっかりと時間をかけて取り組んでいた
 2. 課題を行う環境（プラットフォーム）に慣れていなかった
 3. 課題について理解しにくい、分からない、詰まった部分が多々あった
- 上記1については、時間に縛られないという遠隔授業のメリットであり問題ないと考えられる。上記2については、慣れるまでにある程度時間は必要と考えられるが、プラットフォームの使いやすさが向上することで改善される時間とも言える。上記3については、アンケートで「教員への質問がしにくい」という意見があったことから、手が止まったときの対応（Slackを使って報告）がなかったため、結果的に放置または解決までに時間を要してしまったと思われる。今後は、質問のし易さを改善することで、遠隔で課題に要する時間の短縮を図ることが可能と考えられる。

検証結果

• アンケート結果の比較

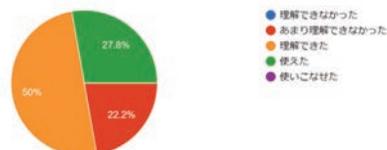
遠隔

今回の授業内容である「フレームワークを使ったプログラミング」についてご回答ください。
21件の回答



対面

今回の授業内容である「フレームワークを使ったプログラミング」についてご回答ください。
18件の回答



検証結果

• アンケートの理解度と課題の完了数の関係を比較

形式	アンケート回答者	課題の完了数	理解できなかった	あまり理解できなかった	理解できた	使えた	使いこなせた
遠隔	21名	0個	1人	1人	0人	0人	0人
		1個	0人	2人	0人	0人	0人
		2個	0人	2人	3人	0人	0人
		3個	0人	0人	0人	1人	0人
		4個	0人	3人	4人	4人	0人
対面	18名	0個	0人	0人	0人	0人	0人
		1個	0人	0人	1人	0人	0人
		2個	0人	4人	4人	3人	0人
		3個	0人	0人	4人	2人	0人
		4個	未実施				

考察

- フレームワークを使ったプログラミングについて「理解できた」「使用できた」と回答した学生は、対面が77.8%(14/18人)で遠隔が57.1%(12/21人)だった。対面では、課題の1、2について全員実施できたが、課題3については宿題とし有志の学生のみ実施、課題4は未着手となってしまった。そのため、アンケートへの回答が課題1、2を終えた上での回答となる学生が多くなったと考えられ、「理解できた」「使用できた」の割合が増えた可能性があることを考慮する必要がある。また、遠隔では初期の段階で課題を4つ提示していたため、完遂できなかった学生については「理解できなかった」「あまり理解できなかった」を選ぶ割合が高まったことが考えられる。
- 遠隔で「理解できなかった」「あまり理解できなかった」と回答した学生の課題完了数を見ると0/4個だった学生が2人、1/4個できた学生が2人、2/4個できた学生が2人、4/4個できた学生が3人だった。4/4個できた学生が「あまり理解できなかった」と回答しているが、これはフレームワークに関する理解が進まないまま、課題に臨んでいたためだと考えられ、なぜフレームワークを利用するかという部分と各課題ごとの説明をより充実させていく必要がある。
- 0/4個～2/4個できた6名の学生が「理解できなかった」「あまり理解できなかった」と回答しているが、うち1名の学生からアンケートにて「プログラムの意味がいまいち理解できず、問題が解けない箇所があった。」との回答があったため、プログラム部分で躓いたものと思われる。加えて、Slackで質問や詰まっている旨のコメントはなかったため、Slackを活用した質問へのハードルが高かったものと思われる。これは初期段階でSlackを使った質問のチュートリアルやHow Toの資料を作成することで対応できるものとする。他の5名の学生については、GitHub、Codespaces、Slackに対して、それぞれ「理解できない」、「あまり理解できなかった」と回答していることが多かった。このことから、プラットフォームのツール部分での躓きが学習へのモチベーションを下げてしまい、結果それがフレームワークの理解度の下げることにつながってしまったと考えられる。今後は、プラットフォームの利用をフォローする説明資料や動画の拡充を図る必要がある。

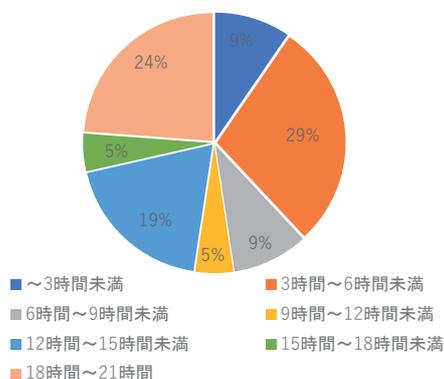
プラットフォームの使いやすさ・分かり易さの検証

検証方法

- プラットフォームの使いやすさ、分かり易さの検証にあたり、プラットフォームを利用した学生にアンケートを実施し、その結果を元に検証を行う
 - アンケートでは、GitHub、Codespaces、Slackについて理解できなかった、あまり理解できなかった、理解できなかった、使えた、使いこなせたの5段階で評価してもらい、良かった点と改善点について回答してもらう。また合わせてプラットフォームの利用時間についても質問を行った。
- また、以下を収集し検証に用いる
 - 対面・遠隔両方コミュニケーション頻度と内容
 - 遠隔でのチャットbotの利用頻度と内容

検証結果

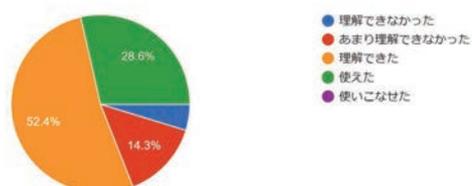
遠隔授業に取り組んでいた時間を教えてください。(21件の回答)



- プラットフォームの利用時間
 - 平均値：10時間20分
 - 中央値：9時間
 - 最小値：1時間30分
 - 最大値：21時間

検証結果

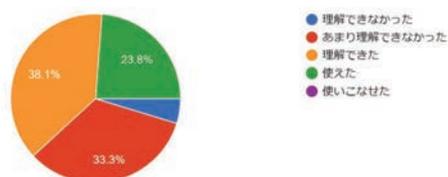
今回の特別授業で使用したGitHubについてご回答ください。
21件の回答



- GitHubを利用してみて良かった点
 - バージョン付けて作成したコード等を管理できる
 - 初めてやることなので新鮮で楽しかった
 - 前提となるプログラミングが詳細に書かれていてわかりやすかった
 - 理解すれば使いやすい
- GitHubの利用に関する改善点
 - 使い方がいまいちわからなかったです。慣れれば使いやすいと思います。
 - 学生証読み取れるようにしてほしい。
 - 理解するまでが難しい
 - 設定や操作が難しい点。

検証結果

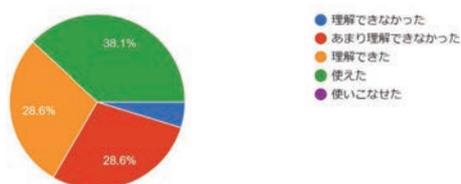
今回の特別授業で使用したCodespaces(クラウドのVSCode)についてご回答ください。
21件の回答



- Codespacesを利用してみて良かった点
 - webでできるのがよかった。
 - おおよそどこにプログラミングを書けばよいかわかりやすかった
 - コードを書くところが分かりやすかった。
 - ソフトウェアをインストールする必要が無い
 - 使いやすく今後も使ってみようと思った
- Codespacesの利用に関する改善点
 - どうしても待ち時間が発生してしまう
 - データが重かった。
 - 使い方がいまいちわからなかったです。慣れれば使いやすいと思います。
 - 若干が使いづらいところがあった。

検証結果

今回の特別授業で使用したSlackについてご回答ください。
21件の回答



- Slackを利用してみて良かった点
 - コミュニケーションが取りやすく使いやすい
 - コミュニケーションが取りやすい環境だった
 - コミュニケーションの履歴が見やすくよかった
 - 他の人の考えや手順の進め方を詳しく知ることができた
 - 質問する場所が決まっていたので分かりやすかった。
- Slackの利用に関する改善点
 - discordなどではいけないのかと感じました。
 - グループに参加する方法で少し詰まった
 - 使い方がいまいちわからなかったです。慣れれば使いやすいと思います。

検証結果

対面と遠隔 (Slackとチャットbot) の質問数と内容を比較

- 対面：計 35回
 - デバッグが動きません (同様の質問が15回)
 - 修正するソースの場所が分からない (同様の質問が7回)
 - 実行後、ポートの競合により画面が見れない (同様の質問が2回)
 - 何を提出すればよいか、また提出ファイルが分からない。(同様の質問が4回)
 - 課題にある例題の実行方法(URLの書き換え方について)が分からない (同様の質問が3回)
 - 編集対象のファイルの場所について (同様の質問が3回)
 - 修正対象となるファイルを教えてほしい
- 遠隔：11回 (うち、チャットbotへの質問数は0回)
 - 自分の名前を選ぶ画面ってどこにありますか？ログインしたあとに違う画面がでます。
 - ログイン画面が表示されているんですけど、usernameがすいません、忘れてしまってどんな形式でした？
 - 一応手順通りやって終わりましたがこれで大丈夫ですかね？
 - いつでも実施していいとのことなんですけど基本的に期限などはありますか？
 - この画面になってしまいました (codespacesの上限)
 - インストールのダイアログが出てきません
 - すみません、私もインストールのダイアログが出ません。数回立ち上げなおしたんですけどダイアログが出なかったです。
 - (課題の実行に関して) これってどうやればいいですか
 - フィードバック失敗したんですがどうしたらいいですか
 - すみません、この赤い×はどうしたら直りますか？
 - 虫と再生ボタンを押したのですがOpen in Browserが表示されません。どうしたらいいですか？

検証結果

(遠隔授業のアンケート結果)

- 普段の対面での授業と今回の遠隔での授業を比べてみて、良かった点
 - 意見や質問をしやすい
 - 好きな時間に受けられる点。
 - 好きな時間に課題に取り組めるのがよかったです
 - 時間を気にせず課題に集中できた。
 - 自分の好きなタイミング、時間で行うことができました。
 - 交通など気にせずに授業を受けることができる
 - 1人の意見が履歴に残っているので後から質問する必要が無い点。
 - いつでも取り組める
 - 自分のペースで進められる
- 普段の対面での授業と今回の遠隔での授業を比べてみて、悪かった点
 - 授業内容の理解度が少し足りなくなった点。
 - アプリの使い方がいまいちわからず、リモートの前に使い方をちゃんと知りたかったです。メッセージだけでは不安でした。
 - わからないことがあった時直接聞いたり指導して頂いた方がわかりやすい時がある
 - 個人での質問がしにくかった。文字化けや資料の情報不足について自分だけなのか焦りました。家のネットの環境が悪くて少しイラッとした。
 - 分からないことを聞きにくかったので対面の方がやりやすいなと思いました

検証結果

(遠隔授業のアンケート結果)

- 特別授業を受けてみた感想
 - 授業はいつもと違った進め方やオンラインでJava、htmlを使える利便性が気持ちよかった
 - 実際にIT系の仕事で使うツールを体験できて将来をイメージしやすくなったと思いました。質問をした時の回答が早くて丁寧で良かったです。
 - 使い方を理解することができましたが、行き詰ることがありました。
 - 質問しやすい環境で授業に取り組みやすかったです
 - 初めての授業の形態で楽しかった
 - 対面の方が受けやすいと感じました。
 - 理解するまでが難しく感じたが、特に不満なことはなかった。
 - GitHubは名前だけ聞いたことがありましたが実際に使って見ることである程度使い方を理解できたので仕事においても上手く使いこなせるようにこれからも学習していきたいです。
 - オンラインでやると難しいと思いました

考察

- 教育プラットフォーム（GitHub、Codespaces、Slack）の利用時間について、平均で10時間20分（中央値9時間）という結果が得られた。対面での授業（3時間）と比較すると3倍以上の時間となった。この要因として、自分の好きなタイミングとペースで学習できていたことが挙げられる。普段の対面授業と今回の遠隔での授業を比べてみて良かった点のアンケートでこの意見が多く挙がっており、遠隔授業のメリットが活かせていたと言える。また別の要因として学生は今回初めてプラットフォーム利用するため資料確認、操作に慣れる、他の学生の状況を伺っていたなどである程度時間が増えたことが考えられる。
- GitHubについて「理解できた」「使用できた」と回答した学生は、81%（17/21人）に達した。また、実際に課題1と2では、それぞれ20人と18人の学生が課題に取り組めており、GitHubについては一定の使いやすさ・分かり易さがあったと言える。これらの要因としては、GitHubに関する説明の資料で一定の理解が得られた、またはGitHubの操作（コミット、プッシュ）に関する手順が詳細に記載されていたことで、躓くことが少なかったことが考えられる。「理解できなかった」「あまり理解できなかった」と回答した学生については、GitHubだけでなくCodespacesやSlackに関する理解度も低い傾向にあった。このことから今回用意した資料や説明だけでは不十分だったとも考えられるため、事前作業段階で詳細な説明、もしくは遠隔での授業開始時にチュートリアルを設けるなどして、確実に触れる・動かせる状態にまでもっていく必要があると思われる。
- Codespacesについて「理解できた」「使用できた」と回答した学生は、61.9%（13/21人）で、GitHubの81%に比べると約19%も下回る結果となった。この要因としては、Codespacesを立ち上げた際に手順とは異なった動きをしたケースが多かったことが挙げられる。学生から手順にあるインストール時のダイアログが表示されない（再現性がなくいまだ原因不明）、Codespacesが立ち上がらない（1つの課題で複数個同時に立ち上げていたことが原因）という報告があった。GitHubでは、このようなイレギュラーな事象は特に発生しなかったことから、アンケートの理解度に差が生まれたものと思われる。また、約8割の学生がCodespacesを使って課題1と2を完了できていることから、Codespacesの操作については特に問題はなかったと言える。

考察

- Slackについて「理解できた」「使用できた」と回答した学生は、66.7%（14/21人）で、「理解できなかった」「あまり理解できなかった」と回答した学生は33.3%（7/21人）だった。「理解できなかった」「あまり理解できなかった」と回答した7名のうち、Slackの利用（コメントやリアクション）が何もなかった学生は2人だった。これらの学生への対応としては、事前作業の段階で、Slackのチュートリアルを実施し、確実に触れてもらうことで、利用へのハードルを下げるのが有効と考える。Slackを利用はされたが、「理解できなかった」「あまり理解できなかった」と回答した学生については、実証実験の中で誤った運用（課題1に関する質問を課題4のところへコメント）をする学生がいたことから、Slackを含むプラットフォームを使う上でルールが分からず、結果それが理解度の低下に繋がったと思われる。プラットフォームの利用する上でルールを明確にした上で、その場所について周知（もしくは固定）することで解消できるものとする。
- 対面と遠隔の質問数を比較したところ、対面が3時間で35回に対して遠隔は4日間で11回だった。遠隔が大きく質問の回数を減らした要因として、Slackの特徴であるやり取りした内容が履歴として残り共有されたことで、同じ質問を人が減ったことが挙げられる。学生からのアンケートでも同様の回答があり、これはSlackをうまく活用できていたと言える。その一方で、対面に比べて質問しづらかったという意見も見受けられたため、先述したSlackを利用する上でハードルを下げる試みが必要と考える。
また、今回Slackに学生からの質問に回答するチャットbotを設けたが、受け付けた質問は0件だった。これは、チャットbotに何を質問すればいいかが分からず、先生に聞いてしまったほうが早いとなってしまったと考えられる。今後、チャットbotを運用していくにあたり、今回の検証で学生から出た質問をチャットbotで回答できるようにすること、またその質問のQA例を提示して利用方法を周知するような対応を行う必要がある。

結語

- 対面での授業と教育プラットフォームを利用した遠隔での授業を行い学習効果の検証を行ったところ、「実際に動作するベースプログラムに対して、一部の命令を書き換えることにより、要件定義を満たすプログラム」レベルであれば、同じ効果が得られた。また、「実際に動作するベースプログラムに対して、一部分の処理を参考に追加することにより、要件定義を満たすプログラム」についても、十分な検証とはならなかったものの、対面と同等の効果が見られた。
- 教育プラットフォームについては使用するツールによって差があるものの、初めての人でも6割以上の人で使用できたという結果が得られた。また、課題も明確になっており、初期段階でのフォローやチュートリアル of 拡充など行い理解度をより高めていくことで、この割合を上げることができると考える。

令和3年度「専修学校における先端技術利活用実証研究」

専修学校遠隔教育導入モデル構築プロジェクト

遠隔教育によるチームプログラミング導入モデルの構築

事業成果報告書

令和4年3月発行

学校法人浦山学園 富山情報ビジネス専門学校

〒939-0341 富山県射水市三ヶ613

TEL:0766-55-1420 FAX:0766-55-0757

●本書の内容を無断で転記、掲載することは禁じます。