



# プログラミング学習における リメディアル教育手法の考察

—学生の ICT に対する意識調査を基に—

鞆 大輔 ・ 上條 英樹 ・ 増田 浩之

**概略** 高等学校におけるプログラミング教育必修化を受け、大学におけるプログラミング教育をどのように改定してゆくべきか。この課題に対するアプローチとして文系学部の大学生を対象としたアンケート調査を実施し、情報教育に対するリメディアル教育の必要性を確認した。またプログラミング教育の特性から学生が抱きがちなプログラミングへの忌避感を解消するためにロボット教材とビジュアルプログラミング言語を用いた教育カリキュラムを提唱し、今後の研究スケジュールと共に概略を示した。なお本研究は近畿大学と TDC ソフト株式会社との共同研究によるものである。

**Abstract** How should programming education in universities be revised due to the impact of mandatory programming education in high schools? We conducted a questionnaire survey of university students in Faculty of Sociology and confirmed the necessity of remedial education for information education. As a result, we proposed an educational curriculum using robotics materials and a visual programming language. This study was conducted in collaboration with Kindai University and TDC SOFT Inc.

**キーワード** プログラミング教育, リメディアル教育, ロボットプログラミング, 産学連携  
**原稿受理日** 2022年1月11日

## 1章. プログラム教育への問題提起

### 1. 研究背景

平成29年に改定された小学校学習指導要領に基づく義務教育でのプログラミング教育が令和2年度より開始された。また平成30年に改定された高等学校学習指導要領に基づき、高等学校における新しい情報科教育である「情報Ⅰ」「情報Ⅱ」が令和4年度より実施される。これらの改訂による最大の変化はプログラミング教育の必修化であり、令和7年度以降に大学へ進学する全ての学生が入学前にプログラミング教育を受けた経験を持つことになる。これまで大学、特に経営学部のような文系学部におけるプログラミング教育は初学者を対象としたアルゴリズムの構築やフローチャートの作成、プログラミング言語習得に必要となる基礎知識等の基礎中の基礎からプログラミング教育を行ってきた。しかし今後、全ての学生が入学までにそれらの基礎的な学習を完了していることが前提となるため、大学におけるプログラミング教育はより高度な内容へと変化することが求められると予想される。

しかし一方で大学入学前に学んだ内容を全ての学生が十分な精度で習得しているとは考えるのは困難である。実際、英語や数学といった大学入試に必要な科目ですら十分な習得が行えていない学生も多く、大学での学びなおし、すなわちリメディアル教育が必要とされるケースも多くに大学で発生している。リメディアル教育は2000年代半ばから行われている学び直しを目的とした科目であり英語や数学、国語と言った科目を対象に実施している大学は多い。IT分野でもこの傾向は顕著であり、2003年に高等学校に科目「情報」が新設されたのち、当初は大学での情報リテラシー科目は遠からず不要になると考えられていたにも関わらず、大学入学者のICTスキルは情報科目の設置前と大きく改善するどころかむしろ低下している。これはスマートフォンの普及によるPC離れが一因であるほか、専門的な知識を持つ情報科教員の不足や「情報」が大学入試科目に含まれないことから数学等の他の入試科目へ振り替えられる等の学教教育現場における問題に起因するものと考えられる。これらの十分な教育を受けられなかった学生の中にはICTに対する忌避感を持つものも少なくなく、将来の進路選択に支障を来すケースも見受けられる。一方で情報科目の教育に熱心な教員が存在する学校、あるいは商業高校や工業高校のようにICTに関連する資格の取得を推奨する学校の出身者のICTスキルは向上する傾向があり、現状でも大学入学者のICTスキルには大きな格差が生じた状態となっている。プログラミング教育

においてもこれらと同種の事象が起こることは容易に想像され、またドロップアウトした学生のプログラミングに対する忌避感も PC 操作の比ではないと予想される。

また、文系学部の将来進路として IT 業界を選択する学生は多く、特にコロナ禍の就職活動においてはこれまで IT 業界に興味を持たなかった学生が IT 業界への就職を行っている。これは IT 業界で必要とする人材が必ずしも ICT スキルやプログラミングスキルのみを求めている訳ではなく、一定の ICT スキルと業務に関わる専門分野の知識をバランスよく習得した人材がシステムエンジニアとして適していることによるものである。

そこで問題となるのが今後大学入学前の課程においてプログラミング教育が必修化された結果、文系学部へ進学する学生に「プログラミング恐怖症」とでも呼ぶべき状態の学生が多く含まれる可能性である。現状でも文系学部には ICT に対する忌避感を持つ学生が一定数存在するが、この傾向がさらに強まることで学生の進路として重要な選択肢となりうる IT 業界が希望進路から外れる事は決して好ましいものではない。そのため、この問題を解決するためにも今後大学におけるプログラミング学習として、リメディアル教育手法の確立が急務であると考ええる。

## 2. 「情報 I」必修化の影響

令和 4 年度より実施される高等学校学習指導要領では必修科目である「情報 I」におけるプログラミング教育について、知識および技術の習得目標として「アルゴリズムを表現する手段、プログラミングによってコンピュータや情報通信ネットワークを活用する方法について理解し技能を身に付けること。」を、思考力、判断力、表現力の習得目標として「コンピュータで扱われる情報の特徴とコンピュータの能力との関係について考察すること。」「目的に応じたアルゴリズムを考え適切な方法で表現し、プログラミングによりコンピュータや情報通信ネットワークを活用するとともに、その過程を評価し改善すること。」「目的に応じたモデル化やシミュレーションを適切に行うとともに、その結果を踏まえて問題の適切な解決方法を考える事。」と定めている。[文部科学省, 2018]

情報 I の具体的な授業内容の例として、文部科学省が公開している高等学校情報科「情報 I」教員研修用教材を引用して解説を行う。

■二分探索  
 リストの中から探索範囲を半分ずつ狭めながら目的のデータを探し出す探索方法である。  
 ここでは7個のデータ a[0]～a[6] が昇順にソートされており、探索値が「43」とする。

| リストの要素 | a[0] | a[1] | a[2] | a[3] | a[4] | a[5] | a[6] |
|--------|------|------|------|------|------|------|------|
| データ    | 25   | 33   | 43   | 51   | 66   | 71   | 88   |

図表4 探索対象のリスト

①中央値>探索値より、上限添字の入れ替え  
 まず、探索範囲の中央を求める。リストの下限は a[0]、上限は a[6] なので、中央は「(下限添字+上限添字) ÷ 2=(0+6) ÷ 2=3」のように求められる。従って、探索範囲の中央にある a[3] のデータ「51」と探索値「43」を比較する。  
 「43」は a[3] の「51」より小さいので、探索範囲は a[0]～a[2] に絞られる。

②中央値<探索値より、下限添字の入れ替え  
 新たな探索範囲の下限は a[0]、上限は a[2] なので中央は「(0+2) ÷ 2=1」のように求められる。従って、a[1] のデータ「33」と探索値「43」を比較する。  
 「43」は a[1] の「33」より大きいので、探索範囲は a[2] に絞られるため、下限添字を中央添字の「1」を1つ増やした値の「2」に変更する。

③中央値=探索値より、検索終了  
 新たな探索範囲の下限は a[2]、上限は a[2] なので中央は「(2+2) ÷ 2=2」である。a[2] のデータ「43」と「43」を比較する。「43」は a[2] の「43」と一致するので、「43」が a[2] にあることが分かる。  
 ※人間であれば、下限が a[2] で上限が a[2] であれば、探索値が a[2] であることは当たり前を感じる。しかし、コンピュータは、定められたアルゴリズムによって動くため、下限と上限が一致するということは中央値もそれと一致するというにしかならない。アルゴリズムに従って、次のステップで中央値と探索値が一致して、探索が完了する。

図表5 二分探索

図1. 「情報Ⅰ」教員用教材抜粋 [文部科学省, 2018]

二分探索とはリスト中から目的にデータを発見するための探索方法の一種であり、先頭から順次データのマッチング判定を行う線形探索よりも複雑な処理を求められるアルゴリズムである。前提条件としてデータが昇順もしくは降順に整列されている必要はあるが、線形探索よりも少ない平均探索回数で目的のデータを探し出す事が可能となる。

この二分探索に関する課題を達成するためには基礎的な要素である変数や配列の理解、条件分岐や繰り返しのアルゴリズムに対する理解、効率的なアルゴリズムを考案する思考力、フローチャートの作成と読解、プログラミング言語でアルゴリズムを表現する技術力が必要となる。これらはいずれも情報通信を専門に学ぶ学生や ICT 関連の職種で働く社会人は常識として会得している知識や能力ではある。しかし裏を返せば情報通信を専門としない学生や社会人にとってはプログラムの作成以前にテキストに記述されている内容の理解すら困難な内容である事は想像に難くない。

この教材は情報科の教員が授業を行うために必要な能力を育成するための教材であるため、掲載した内容がそのまま高校生に提示されるものではないが、学習活動として設定されている課題として「二分探索のアルゴリズムについて考え、そのプログラムを作成する」

と言うものが設定されており、実際に生徒達は必修授業の中でこのアルゴリズムを自ら構築し、プログラムを完成させる事を求められる事になる。

なお選択履修科目である「情報Ⅱ」はさらに高度な内容が設定されており、データサイエンス等の単元と共にプログラミング教育としては情報処理システムの設計から開発、テスト、運用の一連のプロセスやプロジェクトマネジメントについても学ぶとされている。これらは実際にシステム開発を業務として行う IT 系技術者、特にシステムエンジニアが担当する業務を網羅しており、現在本学経営学部において3年次配当の科目として開講されている情報システム論やシステム設計論等に類似する内容となっており、この点も今後の大学における情報教育のカリキュラム検討において注視が必要である。

## 2章. 学生の ICT に関する意識調査

### 1. 基本属性に関する分析

前述のような状況を踏まえ、実際に現在文系学部所属する大学生の情報教育に関する現状はどのようなものであるかを確認するため大学生が ICT 教育をどのように感じているかについて調査を行った。今回のアンケートは近畿大学経営学部経営学科の1年生のうち、情報系科目である「情報倫理」を履修している207名を対象に実施したもので、173名の回答を得た。調査期間は令和3年12月14日から20日まで、調査方法は Google Form を用いた Web アンケート形式である。

まず今回の調査に回答した学生の高校時代の専攻（図2）については94.8%が普通科卒、

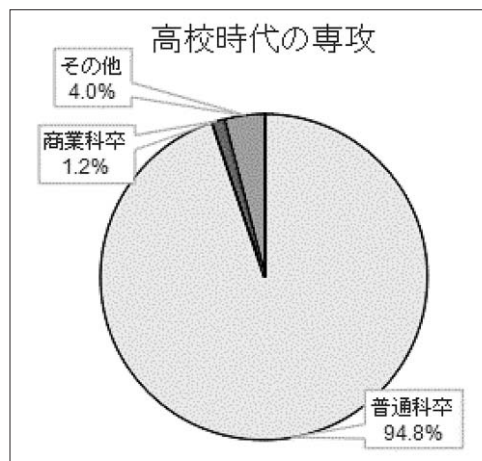


図2. 回答者の高校時代の専攻科

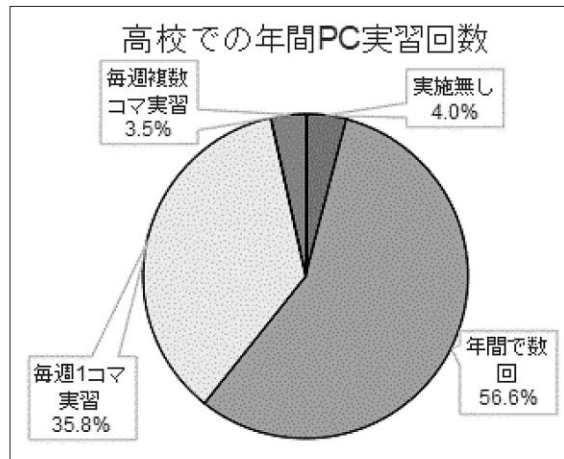


図3. 高校の授業における年間のPC実習回数

1.2%が商業科卒，4.0%がその他（工業科，文理科，国際科，専門学校等）となっており，大多数の学生が普通科出身者となっている。次に高校在籍時に年間どの程度のコンピュータの操作を伴う実習形式の授業を受けたか（図3）については4.0%が「実習は無かった」，56.6%が「年間数回程度の実習」，35.8%が「毎週一コマの実習」，3.5%が「毎週複数コマの実習」と回答している。これら二項目の回答から，今回の調査対象の標準的的学生像は「普通科卒業で年間数コマもしくは週一コマの実習を受講した経験を持つ学生」と考えることができる。平成24年までの普通科高校における情報科目ではコンピュータ操作の実習を必須としない「情報A」を選択する学校が多数を占めていたため，その時点から比較すると高校でコンピュータに触れる機会は大幅に増加していると考えられる。

## 2. 高校での学びに関する分析

次に高校時代の情報科目を通じてPCスキルの習得を実感できているかという問い（図4）については全体の44.5%の学生が「会得できなかった」「あまり会得できなかった」と回答，36.4%の学生が「やや会得した」「会得できた」と回答している。高校時代の実習頻度は学生の実感に大きく影響しており，実習回数が多かった学生ほどPCスキルを身に付けたと感じていることがわかる。また同様に高校時代の実習を通じてPC操作に対してどのような印象を持つようになったかという問い（図5）については全体の11.0%の学生が「苦手になった」「どちらかという苦手になった」と回答しており，本稿で着目する，情報教育を受けた結果としてICTに忌避感を持つ学生の存在が示されている。なおこの項目についても実習回数が多い程忌避感を感じる学生が減少しており，十分な実習回数の確保



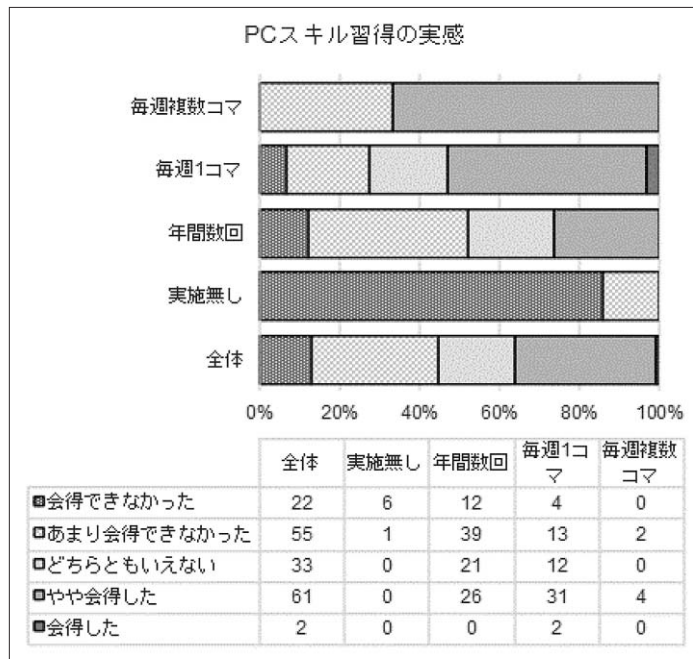


図4. 高校時代の実習を通じた PC スキル習得の実感

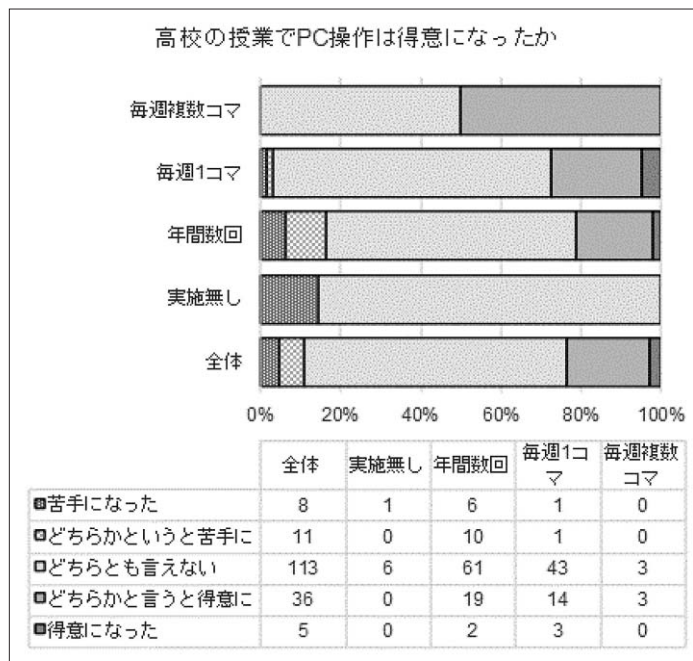


図5. 高校時代の実習を通じた PC 操作に対する心証

が適切なフォローアップを可能とし、その結果学生が ICT に対する忌避感を抱きにくくなると推測される。

情報科目の主観的難易度については高校時代の情報科目と大学での実習を主体としたコンピュータリテラシー科目の双方について調査を行っているが、まず高校での情報科目難易度(図6)については全体の31.2%が「難しかった」「やや難しかった」と回答し、39.3%が「簡単だった」「やや簡単だった」と回答している。実習回数と主管難易度の関係性については年間で数回しか実習を行っていない学生よりも毎週複数コマの実習を行っている学生の方が「簡単だった」「難しかった」と回答している割合が共に増加している。これは実習回数が増加することで科目に対する適性を自覚しやすくなる傾向があるとみることができる。

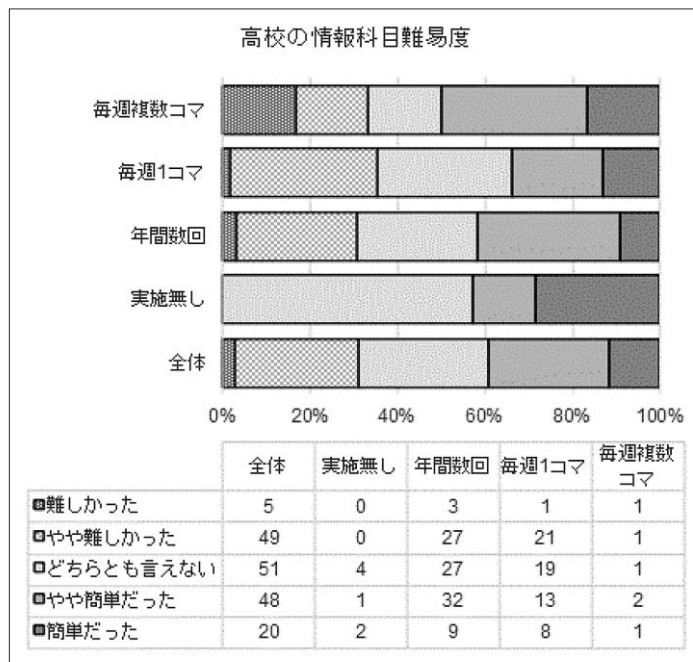


図6. 高校での情報科目に対する主観難易度

### 3. 大学での学びに関する分析

次に大学でのコンピュータリテラシー科目に対する主観難易度(図7)については全体の過半数を超える61.8%が「難しかった」「やや難しかった」と回答し、逆に「簡単だった」「やや簡単だった」と回答した学生は13.9%に留まっている。また大学でのリテラシー科目難易度を高校時代の主観難易度別に集計した結果(図8)では高校時代の主観難易度、



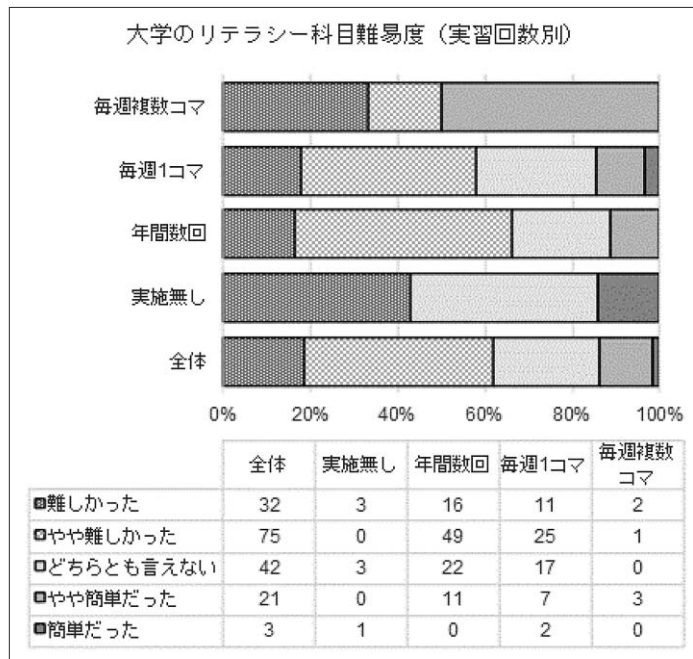


図7. 大学のリテラシー科目に対する主観難易度（高校時代の実習回数別）

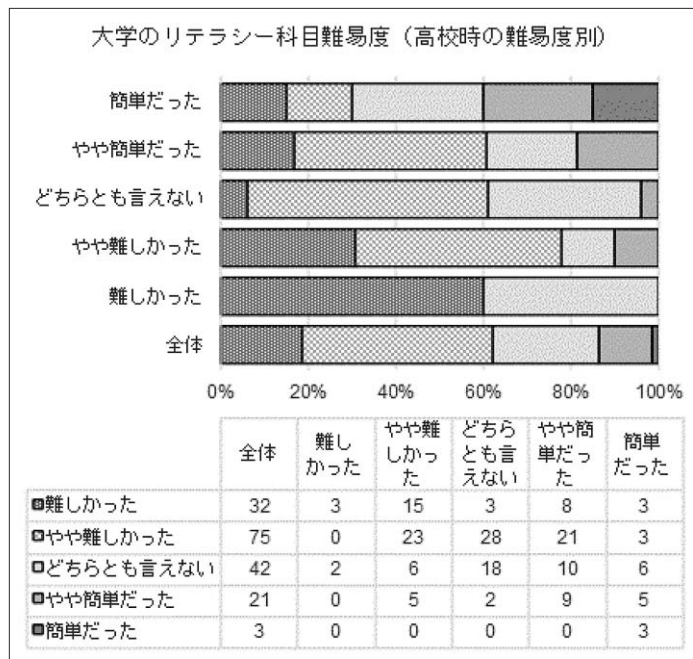


図8. 大学のリテラシー科目に対する主観難易度（高校時代の主観難易度別）

すなわち理解度が大学での科目に対する理解度とある程度相関関係にある事が見受けられる。

高校の情報科目に対する主管難易度と比較すると大学のリテラシー科目の難易度が高いと感じる学生が増加している理由として考えられる要素にはアンケート調査年度における特殊な授業形態、およびカリキュラムによる影響が挙げられる。まず調査を実施した令和3年度は新型コロナウイルスに対する感染症予防対策のため緊急事態宣言が発出されており、春semesterの講義期間はほぼ全て、秋semesterの科目についても講義開始から一月近くの期間がオンライン形式での講義が行われていた。そのため実習科目にもかかわらず講師やTSAによる指導、サポートが受けづらい環境であった点が影響している可能性が考えられる。次に本学におけるコンピュータリテラシー科目が全学部共通のシラバスに基づき実施されているため、経営学部の学生には一部難易度の高い単元が含まれている可能性が考えられる。令和3年度前期の情報リテラシー科目における全学部平均の単位取得率は93%であるが、経営学部の単位取得率は90%であり、文芸学部と共に最も低い値となっている。理工学部の単位取得率が全学科平均で約95%であることを勘案すると、同じカリキュラムでも学部により理解度に差が生じていることは明らかである。

しかし高校時代の理解度と大学での理解度の相関や大学でのリテラシー科目は高校時代の情報科目よりも難易度が高いと感じている学生の多さから勘案すると、大学が想定している理解度に到達していない学生が一定数存在している事は確実であり、理由はどれであれリメディアル教育的な位置づけであるコンピュータリテラシー科目を半数以上の学生が難しいと感じている現状は憂慮すべき状況であることは確かである。

#### 4. プログラミング経験に関する分析

本稿の本題であるプログラミングについて経験の有無と理解度について調査した結果は図9の通りである。全体の2/3以上に当たる67.6%の学生が大学入学までにプログラミングを経験しておらず、経験した学生についても「経験はあるが理解出来ているとは言えない(理解度極低)」が16.8%、「課題をこなすので精一杯(理解度低)」が14.5%と大半が「体験しただけ」というレベルに留まっている。「それなりにできる(理解度並)」と回答した学生は1.2%、「得意である」と回答した学生は皆無であった点から、現状では高校卒行までの課程において実効性のあるレベルでのプログラミング教育は実施されていないとみることができ、今後小中高でプログラミング教育が必修化されることで学生のICTやプログラミングに対する意識や姿勢が大きく変化するであろう事が予想される。

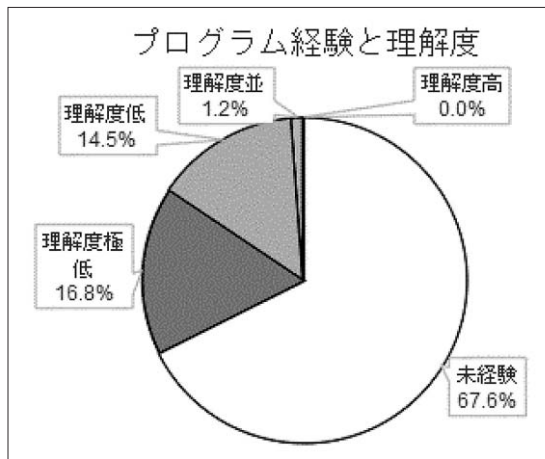


図9. 高校時代のプログラム経験と理解度

### 5. 将来進路に関する分析

経営学科の学生は2年次以降それぞれが希望する専門科目群に応じてコース選択を行うが、今回の調査に回答した学生の進路希望（図10）は経営学科の標準的コースである「企業経営コース」が50.3%，学科科目に ICT 関連科目を追加した「IT ビジネスコース」が35.3%，スポーツ推薦の学生を主とする「スポーツマネジメントコース」14.5%となっている。入学時にほぼコースが固定されるスポーツ推薦の学生を除く一般学生がコース選択を行う際に ICT に対する意識、特に苦手意識や忌避感が影響すると考えられていたが、図11のように今回の調査でも「ICT に苦手意識を持っており、可能であれば避けたい（苦

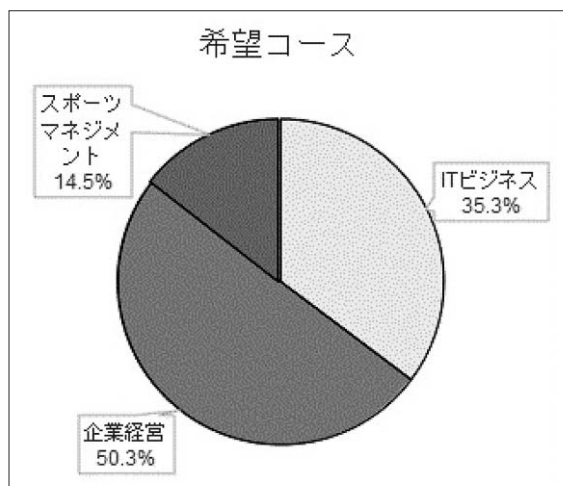


図10. 専攻コースの希望状況

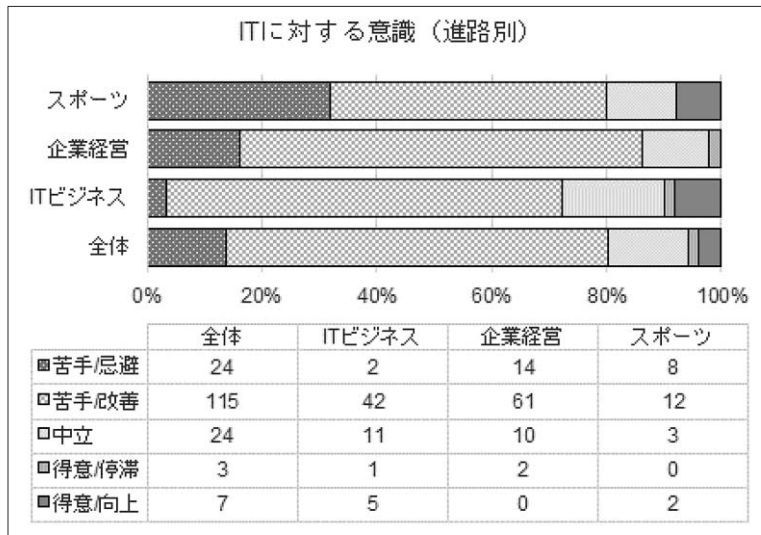


図11. IT に対する苦手意識・得意意識（コース別）

手/忌避)」と回答した学生の割合はコース毎に大きく異なる結果となっている。特に IT ビジネスコースを志す学生は苦手意識を持つ学生であっても苦手/忌避と回答した学生が 3.3%と少なく、企業経営コースの16.1%と比較すると忌避感が少ないことが判る。一方で「苦手だが使えるようになりたい（苦手/改善）」と回答した学生の割合は企業経営コースの70.1%，IT ビジネスコースの68.9%で大きな差は無い。なお「得意だがこれ以上スキルを伸ばす必要性を感じない（得意/停滞）」と回答した学生が企業経営コース2.3%，IT ビジネスコース1.6%と少数だが存在しており、これら学生は大学での情報処理教育に期待していない、もしくは将来の進路に IT を活用する意志がないと考えられる。

以上のようにプログラミング教育が実施されていない「情報」科目において一定数の ICT に対する忌避意識を持つ学生が発生している現状を鑑みればより難易度の高い「情報 I」の必修化後は現状と同じかそれ以上の学生が ICT に対する忌避意識を持った状態で大学に入学してくることはほぼ確実であり、大学における情報系科目のリメディアル教育が必須となるであろう。

## 6. 経営学部生の進路として見たシステムエンジニア

なお本調査と関連した学生の卒業後の進路について本項で記述する。経営学部の卒業生の主たる進路は卸売・小売業，サービス業，金融業などが上げられるが、これらに次いで多いのが情報通信業である。令和2年3月の経営学科卒業生のデータ [近畿大学, 2021]

では卸売・小売業（25.4%）、サービス業（14.4%）、製造業（10.3%）に次いで情報通信業（10.3%）となる。情報通信業の中で最も採用実績の多い職種はシステムエンジニア（SE）である。従来から IT ビジネスコース出身者の多くが SE として就職しているが、令和 2 年度、3 年度の就職活動では IT ビジネスコース以外の学生でも SE を志すケースが増加している。これは外食や観光、交通といった学生人気の高い業界がコロナ禍の影響で採用数を控えていること、またテレワークやオンラインミーティングの普及により IT 系企業の採用マインドが向上していることに起因するもので、経営学部以外の文系学部からも SE を目指す学生は増加している。

一般的に SE は IT 系の技術職に分類される職業であるため、プログラミング能力が必須であると考えられる学生が多い。しかし実際に SE が果たす役割はコーディングのようなプログラム作成そのものではなく、システム設計の上流工程である要求定義、要件定義、各種設計にある。これを行うために必要とされるスキルはクライアントの業務を理解しその内容をプログラマへ伝えることであり、SE にとっては IT の知識と同じかそれ以上に業務知識が必要となる。文系学部出身の SE が担当する業務は企業の基幹業務や専門的な業務であり、多くは経営学、商学、会計学、法学と言った文系学部、特に社会学系学部で学ぶ専門分野に関連する知識を必要とする。そのため、SE として採用される学生の多くはいわゆる文系学部出身の学生であり、採用時には大学で習得した専門分野の習得度と IT 系に対する基本的な適性が問われるケースが多い。そのため文系から学部で SE を目指す学生に教えるべきことは高度なプログラミングスキルではなく IT 系の素養に該当する部分、すなわちアルゴリズムの構築やコンピュータの動作に関する基礎知識、そして何よりもコンピュータやプログラミングに対する拒絶反応を抑える事が主となり、この点からもリメディアル教育の必要性を指摘することができる。

### 3 章. プログラム学習でのリメディアル教育

#### 1. プログラム教育における挫折ポイント

リメディアルとは治療の意味を持つ言葉であり、学生の苦手意識を取り除き、学力だけでなく学習意欲を高める事がリメディアル教育の主たる目的である。そのため、小中高で学んだことと同じ事を大学で単に繰り返すだけではリメディアルにはならず、学生がどこで挫折したかを把握しその問題を解消する必要がある。

現在本学経営学部において開講されているプログラム設計論ではプログラミング教育の

段階としてまず変数や配列，論理演算等のコンピュータプログラムの理解に必要な基本的な要素に関する説明を行い，その上で自然言語によるアルゴリズムの概念とフローチャートによる図示を説明，さらに分岐構造や反復構造と言ったアルゴリズムの基本構造を教えた上で平均・最大・最小・ソート等の代表的な処理についてアルゴリズムを例示しながらコーディングの実習を行っている。プログラム設計論はITビジネスコースに所属する学生のみが履修可能な選択科目であり，基本的にプログラミングに興味のある学生のみが受講するため，前述のような標準的なカリキュラムで講義を行うことで多くの学生が一定のプログラミングスキルを身に付けることができる。

しかし当然のことながらITビジネスコースの学生であっても一度の講義で全てを理解できる訳ではなく，講義後の質問や実習開始後に遡って質問が行われるケースも多い。学生の質問の多くはプログラミングに関する内容が目に見えず，手で触ることも出来ない論理的なものである事に起因しており，プログラミングに必要な構成要素を具体的にイメージできるかどうかプログラミングの習得に大きな影響を与えていると考えられる。

例えば「変数の型」について説明を行う場合，通常のプログラミングに関する教科書では変数とは値を格納できる箱のようなものと記述されている事が多い。しかし，実際のプログラミング言語では変数には整数型や文字列型など「型」が設定されており，それぞれが格納可能な値の種類や使用するメモリ容量が異なる。Visual BasicやVBAで用意されているVariant型変数は「箱」のイメージと合致する挙動で動作するが，整数を扱うint型は通常16ビットないしは32ビットで構成され，実数の場合であれば単精度浮動小数点であるsingleは32ビット，倍精度浮動小数点であるdoubleは64ビットで構成される。文字列を扱うstring型の場合は1文字あたり16ビットのメモリを必要とする可変長の変数となるため，必要文字数に応じたメモリを消費する。一般的な入門者向けのプログラミング教育に用いられる高水準言語ではなぜこのような変数型が存在するのか，それぞれの変数型の使用メモリ容量が何故異なるのか，また何故そのデータサイズが必要であるのかについて深く説明されることは少なく，言語を扱う際の仕様として情報が提示されるだけの場合が多い。これはプログラミングに関する講義があくまでもアルゴリズムの構築やプログラミング言語の習得を主目的としているため変数型のような基礎概念に時間を割く事が困難であるためである。しかし一方でプログラミングを専攻しない学生にとって変数型に関する知識は不要な情報であるため，通常はコンピュータリテラシー講義で詳細な説明が行われることは無く，学生の立場からすれば学習機会が極めて限定的な事柄を常識として理解することを要求される，ともすれば理不尽な構造となっている。



仮に講義で取り扱うプログラミング言語がアセンブラや機械語といった低水準言語の場合であればデータは全て二進数で表現されるため、整数や実数、文字列がそれぞれメモリの中でどのように格納されるか、扱われるかについて詳細に理解しなければプログラムを構築することが不可能であるため、自ずとこれらに対する説明が行われる機会も多くなり理解も深まる。しかし低水準言語は人間に合わせて作られた言語ではないため可読性が低く、初学者に対してこれを推奨することは困難である。すなわち可読性が高い、わかりやすいはずの高水準言語を用いることで逆にプログラミング教育が判りづらくなるという矛盾が生じているのだ。

またプログラムの動作についても、多くのプログラム開発環境では動作確認時には実行結果やエラーが表示されるのみであり、内部でどのような動作が行われているのかを視覚化して表示する機能が十分に用意されているとは言いがたい。通常、技術者はデバッグモードやトレースモード、ステップ実行と呼ばれる機能を用いることでコンピュータの内部動作を確認しながらプログラミングを行うが、プログラムに不慣れな学生にとってはプログラムが「動いた」「動かなかった」と言う結果のみしかわからず、問題のあるプログラムがなぜそのような挙動になっているのかを知るための手がかりを得づらい点も問題である。つまりこれらはコンピュータプログラム、すなわちソフトウェアが論理的な存在であるため画面に表示される情報を元に想像するしか内部の構造や動作を把握する方法が無く、この想像がうまく出来ない学生はプログラミングに挫折し、忌避感を感じていると考えられる。

## 2. ロボットプログラミングの可能性

プログラム学習時の最大の障壁となり得る「ソフトウェアが論理的な存在である」点を解消する方法として有益な手法は、動作を目視で確認でき体感的に学ぶことができる教材を用いた学習手法である。例えば2006年に Scratch 財団とマサチューセッツ工科大学 Media Lab Lifelong Kindergarten Group による共同プロジェクトで開発された Scratch 言語はビジュアル言語と呼ばれており、感覚的にプログラムを作成することができる。そのため Scratch 言語はプログラミングの基礎を学ぶ事に適しているとされており、文部科学省による「小学校プログラミング教育に関する研修教材」においても Scratch によるプログラミング事例が紹介されている。これは Scratch がコーディング、すなわち一般的なキータイプによるコンピュータ言語の入力による構文構築ではなく、マウスやタッチパネルを用いて感覚的にプログラムを作成可能なビジュアル言語である事によるもので、予め用意

された命令ブロックを組み合わせることで処理を記述し、また動作も画面上のキャラクターを動作させるなどの視覚的に確認しやすい教材が用意されている。

もう一つの手法はロボットを用いたプログラミング学習である。1998年に第一世代の製品がリリースされた Lego 社の MINDSTORMS (Lego Mindstorms for Schools) に代表されるロボット教材はプロセッサを搭載したブロック玩具にモーターやセンサーを組み合わせプログラム制御により動作するロボットを組み立てるものである。最新の EV3 ではビジュアル言語を用いた専用プログラミングツール「教育版 EV3 ソフトウェア」を用いることで感覚的にロボットプログラムが可能となっている。

この二つを組み合わせることで、視覚的に動作状況を確認可能なハードウェアであるロボットの制御を高度なプログラミング構文の知識無しに扱う事ができる教材となる。実際、Scratch と学習用ロボットを組み合わせた若年層向けのプログラミング学習用キットは多数発売されており、文部科学省が提供する教材情報においても未就学児から高校生までを対象としたプログラミング教育用の教材情報として Scratch やロボットと組み合わせた事例 [文部科学省, 2021] が紹介されている。

しかし Scratch で記述できるプログラムはさほど複雑なものではなく、小学生向けの教材としては優秀であっても大学生に対して用いるには内容的にも難易度的にもいささか単

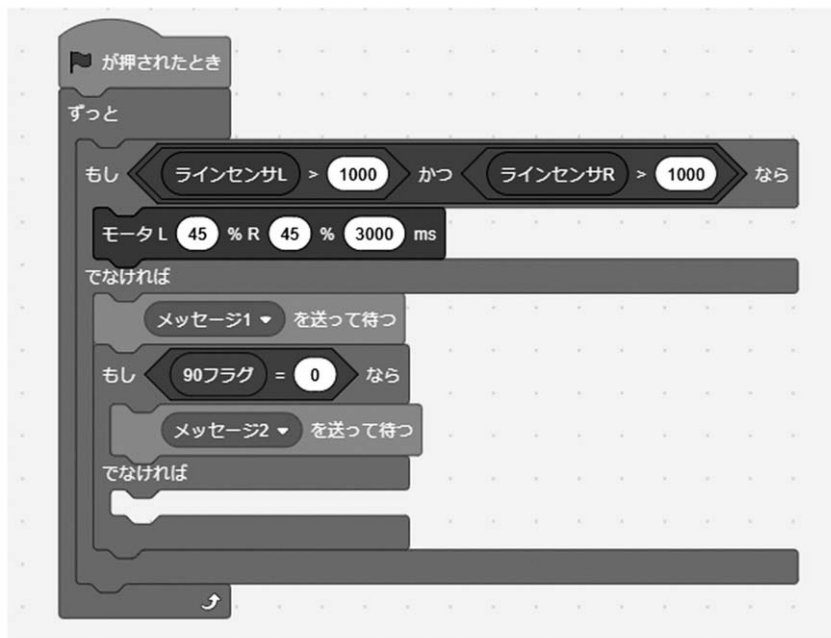


図12. Scratch による実際のプログラム

純すぎるくらいがあり、また MINDSTORMS で構築されるロボットやプログラムは各種のセンサーやジャイロ、駆動機関の制御を伴うため工学的な要素が強く、これをそのままシステムエンジニアを志す文系学部の学生に対する教材として使用するにはやや方向性に問題がある。そのため、これらを単純に組み合わせて既存のコースウェアを大学生に提供するだけでは十分なリメディアル教育とはならない可能性が考えられる。

### 3. ロボットプログラミングコースウェアの概略

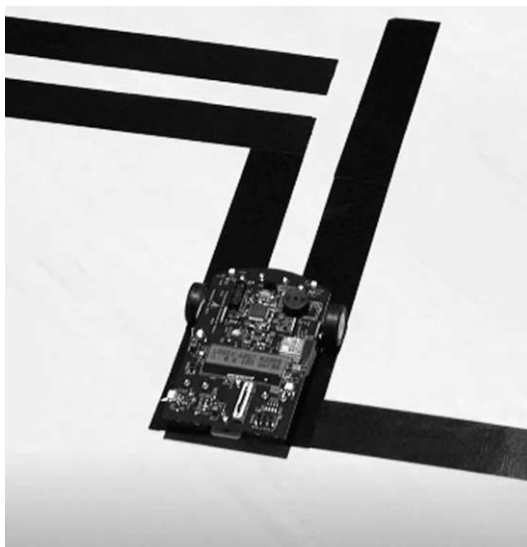


図13. 学習用ロボットカー「iCar」

これらを踏まえた本稿で提示するロボットプログラミングコースウェアでは基本的なセンサーと比較的単純な機構を持つカーネル・ソフト・エンジニアリング社のロボットカー「iCar」(図12)を教材として利用し、これを Scratch で制御を行う方式を採用する。

このコースウェアでは文系学部生の現状やアンケート結果を踏まえ、忌避感を生むポイントを取り除くことでプログラム学習を円滑にステップアップできることを目的に設計を行うこととした。プログラミングへのステップアップは、アルゴリズムとプログラミングを関連付けることで習得する。コースウェアの理想像としては受講前に各自の習熟度について自己申告もしくはアセスメントを実施することで以下3段階から適切なステップから受講することを可能とする形式を想定する。

レベル1：楽しむフェーズ

レベル2：アルゴリズムとプログラムを関連づけるフェーズ

レベル3：プログラムしたいものをプログラムするフェーズ

また本コースウェアでは教材としてロボットを用いるがあくまでもプログラム学習が主目的であるため、ロボットのハードウェアやシステムセットアップといった部分については詳しく触れず、苦手意識を回避するような配慮を行う。

#### 4. リメディアル向けコースウェア案

##### (1) 楽しむフェーズ

このステップはプログラムへの心理的なハードルを排除するステップであり、課程の到達目標を自分が作成したプログラムによってロボットが実際に動作することを体感し、プログラミングの楽しさを感じる事とする。そのため初心者にとって平易なビジュアルプログラミング言語を用い、プログラムとロボットの実際の挙動を見比べることにより視覚的に楽しみながら体験、プログラミングに対する忌避感や心理的なハードルを取り除く。そのため複雑な動作や処理は行わずロボットの基本的な動作（前進、後退、左折、右折、停止など）や一部のセンサーのみ使用する。

##### (2) アルゴリズムとプログラムを関連づけるフェーズ

アルゴリズムとロボットプログラムのパターンを習得するステップであり、課程の到達目標をアルゴリズムやプログラムと動くロボットの動作の関係を理解する事とする。このステップでは、センサーからの入力を元に条件分岐や繰り返しといったアルゴリズムを用いてロボットの制御を行い、基本的な動作の組み合わせによるロボットの制御に習熟することを目的とする。プログラミングで用いられる代表的なアルゴリズムとプログラミングパターンを準備し、これらの習得を次ステップへの進級条件とする。

##### (3) プログラムしたいものをプログラムするフェーズ

創造的なプログラミングへの学習段階として、実現したいものをアルゴリズムで表現しロボットプログラムによる実現するステップであり、この課程では学生が自ら考えた要件や課題をプログラミングすることで要件のパターンとプログラミングを学びながら習熟度を確認する。習得したアルゴリズムのパターンを全て取り込んだ課題をクリアすることで本コースウェアを終了できるものとする。

## 4章. 今後の展望

### 1. 企業研修への転用案

本稿が想定するコースウェアは大学におけるリメディアル教育を目的として開発を行っているが、既卒者向けの研修教材としても転用が可能であると考えられる。経済産業省委託事業の調査によると2030年までに IT 人材の需要がもっとも悪いシナリオで79万人、中間のシナリオでも45万人が不足すると予測されており、またコロナによりこれまで以上に他業種から IT 業界への転職が増加している。IT 企業の場合新卒採用者に対する教育制度は充実しているケースが多く、学生時代にプログラミング経験が少ない文系学部出身者でも業務に従事することが可能である。一方で既卒者の転職においては即戦力である経験者の採用が中心であるため、業務知識を持つプログラミング未経験者のキャリア採用への教育方法はまだ未整備な場合が多いと考えられる。実際、本件の共同研究パートナーである TDC ソフト株式会社においてもキャリア採用はプログラミング経験者に限定されており、キャリア向け IT 教育は実施されていない。しかし、今後の人材不足を勘案するとプログラミング経験者のキャリア採用の拡大は必須で有、このカリキュラムの転用がその対策となると考えられる。また IT 人材の需要はユーザー企業が直接 IT 人材を確保するケースが増えており、ユーザー企業の人材に対する IT 教育の場が増えており、これらの現状を踏まえると今回のカリキュラムの転用は IT 人材確保に向けた対策として利用することも可能であると考えられる。

### 2. 今後の研究スケジュール

以上の考察を踏まえ、今後実際にロボットプログラミング教材の開発と評価を以下のスケジュールで実施する予定である。

#### 【近畿大学】

2021年度：ロボットプログラミング用教材の試作と評価

2022年度上期：2年次配当のプログラム設計論1へ試験導入

2022年度下期：春semesterの実施内容を元に教育効果の評価、教材の改修・再開発

2023年度上期：ロボットプログラミングを用いたリメディアル的教育本格開始

2025年度 情報Iを必修科目として履修した学生が大学へ入学

【TDC ソフト】

2022年度上期：未経験者キャリア採用向けカリキュラム設計

2022年度下期：TDC ソフト社内管理部門の未経験者を対象に検証

2023年度以降：未経験者採用者向け教育の要望がある自社、顧客、パートナーを対象に適用予定

なお教材の開発と検証は本学と TDC ソフト株式会社の共同研究によって行い、効果検証についても学生向けのリメディアル教育と社会人向けの研修教材の両面から行う事を予定している。

引用文献

- 近畿大学. (2021年9月1日). 卒業生の進路・就職先. 参照日：2022年1月2日, 参照先：近畿大学経営学部：<https://www.kindai.ac.jp/business/career/future/after-graduation/>
- 文部科学省. (2018). 高等学校学習指導要領 (平成30年告示). 参照日：2021年12月25日, 参照先：[https://www.mext.go.jp/content/1384661\\_6\\_1\\_3.pdf](https://www.mext.go.jp/content/1384661_6_1_3.pdf)
- 文部科学省. (2018). 高等学校情報科「情報 I」教員研修用教材. 参照日：2021年12月25日, 参照先：[https://www.mext.go.jp/a\\_menu/shotou/zyouhou/detail/1416756.htm](https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416756.htm)
- 文部科学省. (2021). Scratch. 参照日：2021年12月25日, 参照先：未来の学びコンソーシアム：<https://miraino-manabi.mext.go.jp/content/285>

参照文献

- 近畿大学. (2021年9月1日). 卒業生の進路・就職先. 参照日：2022年1月2日, 参照先：近畿大学経営学部：<https://www.kindai.ac.jp/business/career/future/after-graduation/>
- 文部科学省. (2018). 高等学校学習指導要領 (平成30年告示). 参照日：2021年12月25日, 参照先：[https://www.mext.go.jp/content/1384661\\_6\\_1\\_3.pdf](https://www.mext.go.jp/content/1384661_6_1_3.pdf)
- 文部科学省. (2018). 高等学校情報科「情報 I」教員研修用教材. 参照日：2021年12月25日, 参照先：[https://www.mext.go.jp/a\\_menu/shotou/zyouhou/detail/1416756.htm](https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416756.htm)
- 文部科学省. (2021). Scratch. 参照日：2021年12月25日, 参照先：未来の学びコンソーシアム：<https://miraino-manabi.mext.go.jp/content/285>
- 文部科学省. (2021年3月31日). 高等学校情報科担当教員に関する現状及び文部科学省の今後の取組について. 参照日：2021年12月25日, 参照先：[https://www.mext.go.jp/content/20210330-mxt\\_jogai01-100013301\\_007.pdf](https://www.mext.go.jp/content/20210330-mxt_jogai01-100013301_007.pdf)
- 鈴木和也, 荒平高章. (2021). リメディアル教育のための情報リテラシー. 八千代出版.