

小学生のプログラミング的思考を要素ごとに測定する方法の検討

堀部瑞穂¹ 加藤直樹²

概要：本稿では、児童のプログラミング的思考の習熟度を測定し、詳細に分析できるようにすることを目標に、プログラミング的思考を要素ごとに測定するテスト問題を提案する。また、作成したテスト問題の論理的妥当性と表面的妥当性を検証するために行った評価実験について述べる。検証の結果、提案したテスト問題は文部科学省の指針に沿った4つの要素から児童のプログラミング的思考を客観的に測定できるテスト問題として妥当である可能性が示唆された。一方で、要素ごとに分けることで測定しにくくなる要素が生まれてしまうこと、テスト問題全体の分量や難易度の差、必要とする前提知識の影響などが改善点としてあげられた。

キーワード：プログラミング的思考、テスト

1. はじめに

1.1 研究背景

令和2年度から小学校ではプログラミング教育が全面的に実施され始めた。平成29年度に告示された小学校学習指導要領解説【総則編】[1]では、『子供たちが将来どのような職業に就くとしても時代を越えて普遍的に求められる「プログラミング的思考」を育むため、小学校においては、児童が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としている。』と記されており、学習活動から論理的思考力を身に付け、プログラミング的思考を育むことが目標とされている。ここで述べられているプログラミング的思考は、同文書内で『自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力』と定義されている。また、小学校プログラミング教育の手引(第三版)[2]では、プログラミング的思考を働かせるイメージを図1のように示している。

小学生のプログラミング的思考の育成を目指して様々

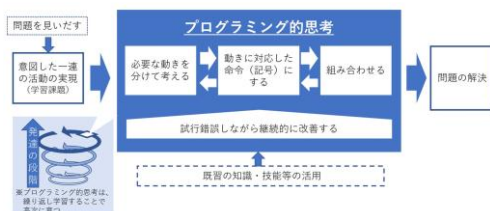


図1 プログラミング的思考のイメージ

な資料が文部科学省から提示されているが、児童がプログラミング的思考をどれほど習得しているか、その測定に関することは言及されていない。矢野らは、小学生のプログラミング能力や論理的思考力の測定について、アンケートによる評価は受講者の主観によるものにすぎないため、より客観的に教育効果を測定する手法が求められると述べており、その一つとして、テストを用いた評価を挙げている[3]。しかし、矢野らが調査した事例の中でプログラミング的思考をテストによって評価した事例はわずか1件しか存在しない。また、小学生のプログラミング的思考を測定するテスト問題として、ベネッセが提供しているPプラスジュニア[4]がある。しかしこのテストは情報活用能力全般を測るものとして設計されており、プログラミング的思考に関しては、文部科学省が想定しているプログラミング的思考の測定が十分に行えていない可能性がある。そのほかに、CTDI: Computational Thinking Diagnostic Instrument や CTt: Computational Thinking Test などの計算論的思考を測定するようなテストもある。教育の情報化の手引き-追補版-[5]では、プログラミング的思考はいわゆるコンピューショナル・シンキング (計算論的思考) の考え方を踏まえつつ、プログラミングと論理的思考との関係を整理しながら提言された定義であると記されていることから、プログラミング的思考を測る指標としてこれらを活用することも考えられる。しかし、これらはそもそも計算論的思考を測るものであるため、プログラミング的思考と粒度が異なり、対照年齢層も一貫性が低い。そのため、Pプラスジュニアと同様にプログラミング的思考の測定が十分に行えない可能性がある。これらのことから、文部科学省の指針に沿って客観的にプログラミング的思考を測定するようなテスト問題の存在が必要である。

1 東京学芸大学教育学部教育支援課程
Faculty of Education, Tokyo Gakugei University
2 東京学芸大学
Tokyo Gakugei University

併せて、テスト問題を作成するにあたっては、プログラミング的思考とはどのようなものかをより明確にしなければならない。そのためには、プログラミング的思考の構成要素とその関係性を考える必要がある。ここで、図 1 ではプログラミング的思考が「必要な動きを分けて考える」、「動きに対応した命令（記号）にする」、「組み合わせる」、「試行錯誤しながら継続的に改善する」という 4 つの要素で構成されていると図示している。そこで、これらの要素を詳細に定義し、その関係性を示すことができれば、プログラミング的思考を要素ごとに測定するテスト問題が作成できるため、それを解くことにより文部科学省の指針に沿ったプログラミング的思考の詳細な測定の実現が期待される。

1.2 本研究の目的

本研究では、児童のプログラミング的思考の習熟度を測定し、詳細に分析できるようにすることを目標に、プログラミング的思考を要素ごとに測定するテスト問題の作成方針を提案し、その方針に基づいて作成した問題の実際に回答した者による評価を通して妥当性を検証する。

2. 先行研究

2.1 検定試験

小学生のプログラミング的思考を測定する問題として、ベネッセが提供している P プラスジュニア[4]がある。これはデジタル・情報活用能力を総合的に測定する小学生向けの検定試験で、文部科学省による「情報活用能力の体系表」の内容をもとにしている。「A 知識及び技能」、「B 思考力・判断力・表現力等」を意識した問題解決型の出題を含んでおり、実社会でも通用する力の育成を目指している試験問題となっている。プログラミング的思考の測定については設定された細かい項目の中からランダムに出題する形式になっているため、文部科学省の想定するプログラミング的思考の測定が十分に行えない可能性がある。

本研究ではこの検定試験も参考にしつつ、図 1 に即してプログラミング的思考を要素ごとに定義し、その関係性を示したうえで問題作成を行う。

2.2 プログラミング的思考の定量的測定

小学生のプログラミング的思考力を測定するために実際に問題を作成し、結果を分析した研究がある。矢野らは、プログラミング教育の教育効果を客観的・定量的に測定するため、プログラミング的思考力を測定する問題を開発し、徳島県内のプログラミングワークショップにて、参加した小学生のプログラミング的思考力を測定した。その測定結果から学年やプログラミングの経験、ワークショップの前後での結果の差などに注目して考察を行い、開発した問題の有効性を評価した。疑似的なロボットプログラミングを行う問題を用いた手法を提案したこの研究では、全 5 問のうち、第 4 問と第 5 問は正答率を用いてプログラミング的

思考力を評価できる可能性が示されている。

本研究ではこの研究で作成された問題の形式も参考にし、プログラミング的思考をより詳細に測定する問題を提案する。

3. プログラミング的思考の構成要素の検討

プログラミング的思考を測定するテスト問題を作成するために、まずはプログラミング的思考の構成要素を定義する。

ここでは、構成要素について、第 1 章で示した図 1 を元に検討していく。また、図 1 の出典 [2]の中に、コンピュータを動作させるための手順例も載せられている（図 2）。これらも踏まえ、各要素を FizzBuzz 問題の例を挙げながら詳細に定義していく。FizzBuzz 問題とは、1 から 100 までの数字を画面に表示していき、3 の倍数のときは数字の代わりに Fizz、5 の倍数のときは数字の代わりに Buzz、15 の倍数のときは数字の代わりに FizzBuzz と表示する、というプログラミングの基本的な問題である。

3.1 要素 1: 「必要な動きを考える」

一つ目の要素（要素 1）は、どのような動きが必要なのか、動きをどのように分ければよいのかを考える力とする。図 1 では「必要な動きを分けて考える」、図 2 では②に該当する。

FizzBuzz 問題では、

- ・「1 から 100 までの数字を画面に表示する」、
- ・「3 の倍数のときは数字の代わりに Fizz と表示する」、
- ・「5 の倍数のときは数字の代わりに Buzz と表示する」、
- ・「15 の倍数のときは数字の代わりに FizzBuzz と表示する」

というように、文章の中から“必要な動き”を読み取り、それらを“分けて”考えることである。さらに、「1 から 100 までの数字を画面に表示する」は、

- ・「最初に 1 を表示する」
- ・「表示している数字に 1 を足す」
- ・「その数を表示する」
- ・「これを 100 まで繰り返す」

と、さらに細かい動きからなることを考えるところも、ここに当てはまる。

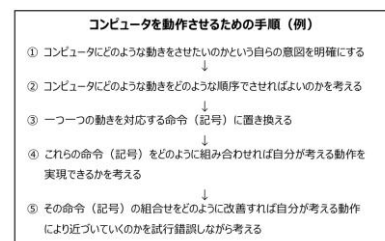


図 2 コンピュータを動作させるための手順例

3.2 要素2:「動きに対応した命令を考える」

二つ目の要素(要素2)は、考えた動きに対応した命令を考える力(当てはめる力)とする。図1では「動きに対応した命令(記号)にする」、図2では③に該当する。

FizzBuzz問題において、どのような動きが必要なのか、動きをどのように分ければよいのかを考えた後は、一つ一つの動きに対してコンピュータ上で動作する命令に置き換えることがこの要素にあたる。そして、変数を定義する、関数を呼び出す、といった単純な変換だけでなく、それらを用いるためにはどのような命令にすべきか、といった記号に置き換えるために必要となる構造を見つけることも、この要素に当たる。

具体的には、「1から100までの数字を画面に表示する」であれば、1から100までの数を順番に表示していくので、今使える記号から考えると、繰り返しの構造が適切であることをを見つける。そして、要素1に戻り、繰り返し構造を用いたより細かい動きを考えて、それに合わせて一つ一つの記号に置き換えていく。「3の倍数のときは数字の代わりにFizzと表示すること」であれば今使える記号からは「3で割れるかを判定する条件分岐の構造が適切と考えることである。なお、「3で割り切れる」は剰余演算を使える場合と使えない場合で動きも変わるためこの点は使用する言語などにより対応を変える必要がある。対応する命令のなかで動きを実現するためにはどれが適切なのか、その命令の実現のためにどういった細かい動きが必要なのかを考えなければいけないことから、要素1と要素2が非常に絡み合う部分とも言える。

3.3 要素3:「命令の組み合わせを考える」

三つ目の要素(要素3)は、命令の組み合わせを考える力であり、図1では「組み合わせる」、図2では④に該当する。

FizzBuzz問題において、一つ一つの動きをコンピュータ上で動作する命令に置き換えた後は、それらをうまく組み合わせ問題の解決を目指すことがこの要素にあたる。具体的には、「変数を用意し、その中に1から100までの数を順番に代入し、表示していく」や「変数に代入されている数字が3で割れるかを判定し、3で割れるときはFizzと表示する」などの要素2までで考えたことをどのように組み合わせたら目的とする動きになるかを考えることである。

命令の順番を入れ替えるような単純な組み合わせから、組み合わせた命令同士をまた組み合わせたりするなど、複雑になっていくほど全体を見通す力が必要となる要素である。

3.4 要素4:「動きの改善を考える」

四つ目の要素は、プログラミング的思考全体を通して、自分が考えた内容を目的に沿うように改善する力であり、

図1では「試行錯誤しながら継続的に改善する」、図2では⑤が該当する。

FizzBuzz問題では、例えば数字が100以上続いて出力されてしまったり、分岐させる条件設定を間違えてしまったり、組み合わせを間違えたことでただの3の倍数のときにもFizzBuzzと出力されてしまったりなどで、正解にたどり着くまでに様々な誤りをしてしまうことがある。そこで、理想の動きと現在の動きのイメージを比較し、差異を見つけ、修正箇所を考えていくことである。

課題を解決するとき、最初から解決策を正しく詳細に考え、そのイメージを実際の動きにすることは容易ではないため、自分が当初想像していた考えを解答の途中で適宜改善、修正していくが必要になる。問題の意図に沿うようなものになっているか、正解のイメージと実際の動きを比較し、違いを見つけ、正しい動きに近づけるための手段や方法を考えることができなければ、自身の成長につながらず、問題が複雑化していったときに対応が難しくなってしまう。

3.5 要素の関係性

ここまで具体例を用いてプログラミング的思考の流れを整理してきた。これを参考にしつつ、先に定義した四つの要素の関係性を整理していく。

(1) 要素1,2,3の往還

要素1、要素2、要素3、一見順序性があるが、考えた動きに対応する命令を考える際、対応する命令がない場合は、別の動きに変える必要が生じる。そのため、要素2の段階から要素1へ戻ることが頻繁に生じる。

同様に、組み合わせる段階で命令を変える、動きを変える必要が生じることがある。そのため、要素3の段階から要素2や要素1に戻ることも頻繁に生じる。

図1では要素1と要素2、要素2と要素3がそれぞれ左右の矢印で結ばれており、要素1と要素3は直接行き来することがないかのように描写されている。上記の通り、組み合わせを考えた後に必要な動きを分けて考える段階へと直接遷移することも多々ある。反対に、必要な動きを考えてから直接組み合わせを考えるようなことは起こりえない。動きを考えた後は必ずコンピュータ上で動作する命令に置き換える必要があるため、要素1→要素3という関係性は成り立つことはない。必ず要素2を経由する要素1→要素2→要素3という関係性になる。

(2) 要素4の作用

要素4はどの要素のどの段階でも作用する。

誤った動きを考えていた場合、考えた命令が動きに対応しない場合、組み合わせ方を間違えた場合など、要素4は常に必要な力となってくる。そのため、要素4は独立したものと捉えず、要素1に含まれる要素4、要素2に含まれ

る要素4, 要素3に含まれる要素4, のように各要素に付随する形で考える。

4. テスト問題の作成方針

4.1 原則的な方針

第一に, プログラミング教育の対象者が小学生であることから, 小学生が取り組める問題とすることが必要である。そこで, 問題は原則として, 抽象的な問題ではなく, 小学校におけるプログラミング教育で行われている現実的な活動を想定したものとする。

第二に, 要素ごとにプログラミング的思考を測定できるようにする。第3.5項で述べたように, プログラミング的思考による活動は各要素が独立していない。そこで単純に具体的なプログラミング活動を行う問題とすると, たとえば間違った場合でも, どの要素が欠けていることなのかを判別できない。そこで, ある要素以外に必要な要素にかかる部分を考える必要のないように手立てをする。

ただし, 要素4に関しては独立できないため, 問題作成時にこの原則は適用しない。

4.2 習熟度とテスト適応性

本テストでは, それぞれの要素の習熟度には, 自力で答えを考えられる(Aレベル), 問題文や選択肢で示されている各要素の内容を理解できる(Bレベル), 問題文や選択肢で示されていても各要素の内容を理解できない(Cレベル)の三段階があるものとする。

要素1の必要な動きを分けて考えるためには, 文章の読解力や一般常識といったプログラミング的思考ではない能力や知識が必要である。要素2の動きに対応した命令を考えるためには, 必要な動きを分けて考えた結果を利用することになるため, 要素1の習熟度がBレベルに達している必要がある。要素3の組み合わせを考えるためには, 必要な動きを分けて考えた結果と, 動きに対応した命令を考える結果を利用することになるため, 要素1, 要素2, それぞれの習熟度がBレベルに達している必要がある。

これらを踏まえると, 各要素の習熟度に関して, 表1のように4つのグループに分けられる。

第一グループは, 要素1がCレベルである人のグループである。このグループは要素2以降のテスト結果が正確に

表1. 回答者のグループ分け

	要素1	要素2	要素3
第一グループ	C		
第二グループ	A or B	C	
第三グループ	A or B	A or B	C
第四グループ	A or B	A or B	A or B

表れないグループとなる。第二グループは要素1がA又はBであるが要素2がCの人のグループである。このグループは要素3以降のテスト結果が正確に表れないグループとなる。第三グループは要素1, 要素2ともにAまたはBであるが, 要素3がCの人のグループである。このグループは, 要素4のテスト結果が正確に表れないグループとなる。第四グループは要素1, 要素2, 要素3すべてがAまたはBのグループである。このグループのみ, すべてのテスト結果が正確に表れることを想定している。

本テストを実施する際には, この点を考慮し, 段階的な実施をする必要がある。これは, そもそもプログラミング教育がプログラミング的思考の習熟度を考慮することなく行われていることの問題にも関連しており, 本テストなどを用いて, プログラミング教育自体を段階的に実施する必要があることの示唆にもなる。

5. 問題作成

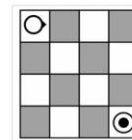
5.1 第一原則からの具体的な方針

第一原則に則り, 小学校のプログラミング教育でもよく取り上げられる, マス目上において対象物を目的地まで移動させる動きを, ブロックを用いて考える活動を取り上げる。これは矢野ら[3]の論文でも取り上げていたものである。矢野らの問題例を図3に示す。

5.2 第二原則からの具体的な方針

今回作成するテストは特定の教科の内容(例えば算数や理科など)にあえて寄せることはせず, あくまでもプログラミング的思考をもとにした問題にする。プログラミング教育の手引き一追補版一第3章において, 小学校プログラミング教育のねらい(図4)では, 小学校プログラミング教育のねらいを3つ述べている。今回作成するテストは“プログラミング的思考を測定する”ことが目的のため, この中でも特に①に焦点を当てた問題を作る必要がある。そのため, 前章で述べた第二の原則にも関連して, プログラミング的思考に含まれる要素以外の能力を極力除くため, 各教科で取り扱う内容に寄せることはしない。

○を●のところまでうごかしましよう!
つかえるブロックは5しゆるいです。
ブロックはなんこつかってもかまいません。



ア: 1マスすすむ

イ: みぎにむく

ウ: ●回くりかえす

エ: もし のうえにいたら

オ: もし のうえにいたら

こたえ

図3 矢野らの問題例

1. 小学校プログラミング教育で育成する資質・能力

(1) 小学校プログラミング教育のねらい

小学校におけるプログラミング教育のねらいは、大まかに言えば次の3つということができる。

- ①「プログラミング的思考」を育むこと
- ②プログラムの働きやよさ、情報社会がコンピュータ等の情報技術によって支えられていることなどに気付くことができるようにするとともに、コンピュータ等を上手に活用して身近な問題を解決したり、よりよい社会を築いたりしようとする態度を育むこと
- ③各教科等の内容を指導する中で実施する場合には、各教科等での学びをより確実なものとする

図 4 小学校プログラミング教育のねらい

また、要素 4 は図 1 で各要素にそれぞれ付随する形で示されている。つまり、要素 1 を改善する力、要素 2 を改善する力、要素 3 を改善する力がそれぞれ存在する。しかし、第二の原則でも述べたように、要素 4 をこのように分けて考えてしまうと、たとえば要素 3 で組み合わせを考えた時、答えを出すときに要素 3 と要素 4 の力をどれぐらい使ったのか、判断が難しくなってしまう。そこで、要素 4 についても、各要素にそれぞれ付随する形として捉えずに独立させる。

先に示した矢野らの問題では、動きのブロックが細かく分けられた状態であらかじめ提示されている。ただし必要なブロックもあるので、必要なブロックは解答者がピックアップしなければいけない。また、条件分岐もブロックの動きに含まれており、これらを組み合わせて問題を解決するため、この問題には要素 1, 3 の力が含まれている。そのため、例えば要素 1 のみの力を測定する場合は、条件分岐のような関数は既に完成している又は問題自体に存在しないこと、動きは既に組み合わせられている状態で示されていることが必要となる。このように、測定したい要素で思考する部分以外に考えなければいけない内容は問題文中ですべて説明し、あらかじめ示しておく。そして第 4.2 項で定義した各要素の習熟度が B 以上ある場合、測定したい要素以外の内容は問題文中から読み取れているため、測定したい要素のみを測定できているものとする。

5.3 全体構造

テストの構成は、第二の原則に基づき、要素 1 の力（課題の意図に沿った必要な動きを考えることができるか、その動きを適切に分解することができるかといった力）を測定する問題（大問 1）、要素 2 の力（考えた動きを課題の意図に沿った命令に置き換えることができるか、特に条件分岐や繰り返しのようプログラミングの基礎が理解できているか、といった力）を測定する問題（大問 2）、要素 3 の力（課題の意図に沿って命令を組み合わせられるのか、といった力）を測定する問題（大問 3）、要素 4 の力（間違いのある既に出来上がった考えを見て、理由を述べて間違いを指摘できるか、といった全体を見て間違いを見つけられる力）を測定する問題（大問 4）から構成する。

5.4 大問 1：必要な動きを分けて考える力の測定問題

大問 1 は要素 1 の「必要な動きを分けて考える」力を測定する問題である。東西南北にそれぞれ 1 マス進むカードを用いて、マス目の中心にいる“私”を指定された目的地へ移動する動きを考える内容とした。この大問では要素 1 の必要な動きを考えること以外のプログラミング的思考が極力働かないようにするために、マス目を移動すること、カードを活用することは矢野らの問題と同じであるが、条件分岐のカードを無くし、要素 3（カードの組み合わせを考えること）が必要ない問題とした。

5.4.1 問 1～問 3

問 1 はマス目上の“お店”に行くまでに必要なカードを、問 2 は不要なカードを答える問題である（図 5）。問 3 は、問 1, 2 とは逆に、用意されたカードからどこに着くのかを答える問題である。カードが表す地図上での動きがイメージできれば正答でき、要素 3 の動きの順番（組み合わせ）を考える必要をなくしている。

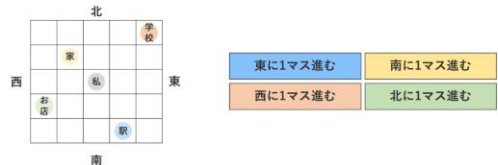
必要なカード（動き）が与えられたとき、それらを用いることでどんな動きになるのかを考えられているかを確認する問題であり、B レベルに該当する。

5.4.2 問 4～問 6

問 4 はある場所を経由してある場所に行くまでに必要なカードの種類とその枚数を答える問題（図 6）、問 5 はある場所を経由してある場所に行くまでに必要なカードのうち、

第1問

下の地図上で「私」はマス目の中心にいます。周りには「学校」、「家」、「お店」、「駅」があります。私ができる動きは下のカードの通りです。



次の問題に答えましょう。ただし、どの問題でもスタート地点から目的地までは最短の道を選びます。

問1

「お店」に行くまでに必要なカードの種類を下からすべて選び、記号で答えましょう。

- (ア) 東に1マス進む
- (イ) 西に1マス進む
- (ウ) 南に1マス進む
- (エ) 北に1マス進む

答え、_____

問2

「学校」に行くまでに不要なカードの種類を下からすべて選び、記号で答えましょう。

- (ア) 東に1マス進む
- (イ) 西に1マス進む
- (ウ) 南に1マス進む
- (エ) 北に1マス進む

答え、_____

問3

東に1マス進む を1枚、南に1マス進む を2枚使うと、どこに着きますか、下から選んで記号で答えましょう。

- (ア) 学校
- (イ) 駅
- (ウ) 家
- (エ) お店

答え、_____

図 5 大問 1：問 1～問 3

使用する枚数が最も少ないカードを答える問題、問6はどこかを經由して目的地に移動するとき、使うカードの合計枚数が最も多くなるのは、どこを經由してどこに行くときかを答える問題である。

經由する目的地が一か所増えているが、前半の動きと後半の動きにそれぞれ分けてイメージできればよく、答えもその順序性を問わないため、問1～問3と同様に組み合わせを考える必要がない。

自分で動きを分けて考えられているか確認する問題であるため、Aレベルに該当する。

5.5 大問2：動きに対応した命令にする力の測定問題

大問2は要素2の「動きに対応した命令を考える」(条件分岐や繰り返しを考える)力を測定する問題である。

大問1では要素1のみの力を測定できるように、条件分岐や繰り返しは存在しない、動きを組み合わせる必要のない問題であった。大問2も同様に、必要な動きとその動きの組み合わせは既に問題で示された状態にし、条件分岐と繰り返しに関することのみを考える問題とした。

また、問題を作成するにあたってフローチャートは用いないこととした。フローチャートはアルゴリズムの表現方法であるが、これを理解できていないと問題の意味が分からなくなることを防ぐためである。小学5,6年生にとってフローチャートは見慣れておらず、学習指導要領に含まれているものでもないため、今回は文章のみで問題内のアルゴリズムを説明した。

問4

「お店」に寄ってから「家」に行くまでに必要なカードの枚数はそれぞれ何枚ですか。(ア)、(イ)、(ウ)、(エ)にあてはまる数字を書きましよう。使わないときは0と書きましよう。

東に1マス進む	×	(あ)枚
西に1マス進む	×	(い)枚
南に1マス進む	×	(う)枚
北に1マス進む	×	(え)枚

答え. (あ) (い) (う) (え)

問5

「学校」に寄ってから「駅」に行くまでに必要なカードの中で、使う枚数が最も少ないカードを下から選び、記号で答えましよう。

- (ア) 東に1マス進む
- (イ) 西に1マス進む
- (ウ) 南に1マス進む
- (エ) 北に1マス進む

答え. _____

問6

問4や問5のようにどこかに寄ってからどこかに行く方法で移動するとき、使うカードの合計枚数が最も多くなるのはどこに寄ってからどこに行くまででしょうか。(あ)、(い)にあてはまるものを下の記号からそれぞれ選んで答えましよう。

使うカードの合計枚数が最も多くなるのは、(あ)に寄ってから(い)に移動するとき。

- (ア) 学校
- (イ) 駅
- (ウ) 家
- (エ) お店

答え. (あ) (い)

図6 大問1：問4～問6

問1と問3は条件分岐、問2は繰り返し(条件)の理解を測る問題である。ここでは、問1のみ詳細を記す。

問1は硬貨を入れると特定の動作をするロボットの仕組みを考える問題である(図7)。条件分岐の理解を確認するために、あきらくんの解説の中で動きの分岐条件が空欄になっている。必要な動きやそれらを組み合わせた全体の流れは解説ですでに述べているため、要素2以外の力は必要ない。問題文で示されている条件がそのまま分岐点になっているため、問題と解説を照らし合わせて条件分岐が理解できていれば正答できる。

5.6 大問3：命令の組み合わせを考える力の測定問題

大問3は要素3の「組み合わせる」力を測定する問題である。大問3も大問2と同様に、必要な動きと条件分岐は既に示された状態にし、組み合わせに関することのみを問題で答える形式にするようにした。

問1と問2は問題文に示された動きについて、4つの手順説明のうち正しいものを選ぶ問題である。問3は与えられた選択肢から組み合わせを自力で考える問題である。ここでは問3のみ詳細を説明する。

問3では、読み取った温度によって電光掲示板に特定の表示をするプログラムの仕組みを考える問題である(図8)。動きの流れは解説で示されており、必要な動きと分岐条件は選択肢で示されているため、解説内の空欄に合うように記号を組み合わせる形式となっている。問題文からプログラムの仕組みを読み取り、選択肢内で示されている必要な動きと動きの分岐点を理解し、最終的な流れをイメージできれば正答できる。

5.6.1 大問4：動きの改善を考える力の測定問題

大問4は4つ目の要素である「動きの改善を考える」力を測定する問題である(図9)。

問1

あきらくんの家には硬貨をいれると手を挙げる貯金ロボットがあります。このロボットは下の条件に従って動きましよう。

- 入れることができる硬貨は、1円5円10円50円100円500円です。
- 100円以上の硬貨を入れると右手を挙げましよう。
- 穴の開いた硬貨を入れると左手を挙げましよう。

あきらくんはみんなの前でロボットの仕組みについて説明してましよう。(1)～(3)に当てはまる数をア～カから選んで答えましよう。ただし、同じ記号は何度選んでもよいです。



あきらくん

「このロボットはお金を入れると、まずそのお金が(1)円以上かどうかを判断してましよう。もし(1)円以上なら右手を挙げましよう。違ったら、次にそのお金が(2)円もしくは(3)円かを判断してましよう。もし(2)円か(3)円のどちらかなら左手を挙げましよう。違ったら何もしてましよう。」

- (ア) 1
- (イ) 5
- (ウ) 10
- (エ) 50
- (オ) 100
- (カ) 500

答え. (1) (2) (3)

図7 大問2：問1

問3

温度計で気温を読み取り、気温によって電光掲示板の表示を下のように変えるプログラムを考えます。

- 計測した温度が、
- 30℃より高いとき、「暑い日です>.<」と表示します。
 - 20～30℃のとき、「ちょうどよいです^.^」と表示します。
 - 20℃より低いとき、「少し寒いです-.-」と表示します。

このプログラムの内容をゆうじくんは次のように考えた。空欄の(1)、(2)に当てはまるものをア～オから、(A)～(C)に当てはまるものをカ～クからそれぞれ選びなさい。

ゆうじくん 最初に温度を読み取って、まず(1)かどうかを調べます。もし(1)なら、(A)と表示します。そうじゃないなら、次に(2)かどうかを調べます。もし(2)なら、(B)と表示します。そうじゃないなら(C)と表示します。

- (ア) 温度は 30℃より高い
(イ) 温度は 30℃より低い
(ウ) 温度は 20～30℃の間にある
(エ) 温度は 20℃より高い
(オ) 温度は 20℃より低い

- (カ) 「暑い日です>.<」
(キ) 「ちょうどよいです^.^」
(ク) 「少し寒いです-.-」

答え、(1) _____ (2) _____
(A) _____ (B) _____ (C) _____

図 8 大問 3：問 3

第 5.2 項でも述べたように、今回のテストでは要素 4 を、全体を見て間違いを見つけられるかで測定する。そのため、問題を解く際には、必要な動き、分岐や繰り返しの条件、組み合わせのすべてに注目しながら、示された動きと正しい動きを比較し、間違っている箇所を指摘する形式で問題を作成した。

5.6.2 問 1

問 1 は間違った動きをする箇所を理由付きで説明する問題である。今までの問題は空欄に当てはまるものを選択肢で選んできたが、この問題は空欄の無い説明でどこが間違っているかを指摘し、理由を説明する形式となっている。正しい動きと説明の動きをそれぞれイメージして比較し、間違った箇所を見つけ、言葉で説明できれば正答できる。

5.6.3 問 2

問 2 も同様に、説明された動きが正しいかどうかを理由付きで説明する問題である。形式は問 1 とほとんど同じであるが、今回は二人分の説明があり、間違っているかどうかをそれぞれ判断しなければいけない問題となっている。その分問 1 に比べると難易度は上がるが、正しい動きと説明の動きをそれぞれイメージして比較し、間違った箇所があるかどうかを探し、もし間違いがある場合はそれを言葉で説明できれば正答できる点は同じである。

6. 評価実験

6.1 評価に用いたテスト理論

評価を行うにあたっては、古典的テスト理論と項目反応理論の 2 つのテスト理論体系のうち、作成したテストが小

規模であることや導入コストが低いことから、古典的テスト理論を採用する。

名古屋大学大学院教育発達科学研究科心理発達科学専攻石井研究室の Web サイト[6]より、古典テスト理論ではテストの妥当性と信頼性を測定し、その結果からテストを評価するが、信頼性の測定までたどり着けなかったため、本研究ではテストの妥当性評価のみを行う。妥当性を確認する区分として主に内容の妥当性、基準的妥当性、構成概念妥当性の 3 つがあるが、作成したテストと似たようなテス

第4問

問 1

さくらさんはボタンを押すと時間によって挨拶を変える時計ロボットを買いました。この時計ロボットは下の条件に従って動きます。

- ボタンを押したとき、
- 5:00～10:59 の間は「おはよう」としゃべる
 - 11:00～17:59 の間は「こんにちは」としゃべる
 - 18:00～21:59 の間は「こんばんは」としゃべる
 - それ以外の間は「おやすみなさい」としゃべる

さくらさんはこのロボットの仕組みについて下のように説明を考えていました。しかし、さくらさんはこの説明を考えている途中で自分の間違いに気がきました。さくらさんの説明はどこが間違っているでしょうか。間違っている理由を説明しましょう。

さくらさん 「このロボットはボタンが押されたときに今何時かを調べます。もし時間の数が 5 以上 11 以下なら「おはよう」と言います。11 以上 18 以下なら「こんにちは」と言います。③は時間が 18 以上 22 以下なら「こんばんは」と言います。それ以外の時は「おやすみ」と言います。」

(間違っている理由)

問 2

はるかさんの家にはプリントを仕分けるロボットがあります。このロボットはプリントをセットするとフロントに書かれた教科名と日付を読み取り、設定した条件に従ってボックスに仕分けていきます。

今回、はるかさんは下の条件に従ってプリントを仕分けるつもりです。

- 国語と社会のプリントは残す。それ以外のプリントは捨てる。
- ただし、11 月の算数のプリントはテスト勉強で使用するため、残す。

はるかさんはこの条件で仕分けロボットを使うときの設定を下のように考えました。

ゆうじくん まずはプリントが国語か社会かを調べます。もし国語か社会なら残します。国語と社会以外なら、次に 11 月かどうかを調べます。もし 11 月以外なら捨てます。11 月なら、そのプリントが算数かどうかを調べます。算数なら残します。算数以外なら捨てます。

ここで、弟のじゅんくんは次のように考えました。

ゆうじくん まずはプリントが 11 月かどうかを調べます。もし 11 月なら、次に算数かどうかを調べます。もし算数なら残します。算数以外なら捨てます。最初が 11 月以外なら、次に国語か社会かどうかを調べます。もし国語か社会なら残します。国語と社会以外なら捨てます。

はるかさんとじゅんくんの考えで、適切なものを下のア～エから 1 つ選び、記号で答えましょう。また、選んだ理由も書きましょう。

- (ア) 二人とも正しい
(イ) はるかさんが正しく、じゅんくんが間違っている
(ウ) じゅんくんが正しく、はるかさんが間違っている
(エ) 二人とも間違っている

答え、_____

(間違っている理由)

図 9 大問 4：問 1～問 2

トが未だ存在せず他のテストとの比較が難しいことから、今回は内容的妥当性に絞って、専門家から見た適切さを考える論理的妥当性の評価と、回答者の目から見た適切さを考える表面的妥当性の評価を行った。

6.2 評価方法

内容的妥当性の評価に絞ってアンケート形式による評価実験を行った。対象は情報教育を専門に学ぶ学部生(3年生以上)と大学院生の計12名である。被験者には、まず作成したテスト問題に目を通してもらい、次に本研究で定義した4つの要素について説明する文章を読んでもらう。その後、テスト問題に対して論理的妥当性、表面的妥当性の側面から用意したアンケートに回答してもらい、「非常に妥当」を2点、「全く妥当でない」を-2点として、2点から-2点の5段階の点数付けによる評価をしてもらった。それぞれの質問項目と回答結果を、図10と図11に示す。

6.2.1 考察

評価実験の結果から本研究で作成したテスト問題はプログラミング的思考を要素ごとに測定する上で一定の妥当性を示したが、論理的妥当性における自由記述の内容から、いくつかの課題も挙げられた。

二つ目の要素(動きに対応した命令を考える)については、その力を使う上でプログラミング言語やツールの知識が前提条件として必要になる。そのため、前提条件に極力依存しないように問題を作成することは非常に困難である。本研究ではこの課題を解消するために、対応した命令を考える力の中でも、条件分岐や繰り返しのような一般化しやすい構造(命令)に着目した問題を作成したが、この内容だけでは二つ目の要素を測っているとは言えないという指摘が寄せられた。

前提条件や知識を必要とせずに問題を作成するために、テスト問題オリジナルのプログラミング言語を用いるとい

った手段も考えられるが、結局は説明量が膨大になる、説明の理解力が問題の正答率に関わってくるといった別の問題が生じるため、様々な点を考慮しながらもう一度考え直す必要がある。

また、大問ごとに難易度のばらつきがあることや、文章量が多いこと、問題の分かりにくさなども指摘された。難易度のばらつきに関しては各大問の問題数の少なさが要因として考えられるが、そもそも全体の文章量が多いため、過度に時間がかからないよう調整するために問題数を減らした側面もある。そしてその文章量の多さが問題の分かりにくさにもつながっているため、図表などを増やして直感的にわかりやすくすることで文章量を減らしたり、表現を変えたりすることで改善される可能性があると考えている。

7. おわりに

本稿では、児童のプログラミング的思考の習熟度を測定し、詳細に分析できるようにすることを目標に、プログラミング的思考を要素ごとに測定するテスト問題作成方針の提案と、それに基づいて作成した問題を実際に回答した者による評価を通じた妥当性検証について述べた。

検証からは、文部科学省の指針に沿った4つの要素から児童のプログラミング的思考を客観的に測定できるテスト問題として妥当である可能性が示唆された。一方で、問題の曖昧性、難易度や分量、必要とする前提知識の影響など、改善が必要な部分も散見された。これらの点の解消していくことが今後の課題である。

また、今回は一つのテストからのみの評価であり、大学生・大学院生による評価しか行っていないことから、その信頼性は確保されておらず、そもそも方針自体の検証まで至っていない。今後、この観点から信頼性を向上させる取り組みを重ねると共に、実際の小学生に対してテストを通しての妥当性の検証、試験時間や難易度の調整を行なっていくことが必要である。

参考文献

- [1] 【総則編】小学校学習指導要領(平成29年告示)解説
- [2] 小学校プログラミング教育の手引(第三版)(最終確認:2025年3月)
- [3] 矢野里奈, 谷岡広樹, 松浦健二, 佐野雅彦, 上田哲史: 小学生のプログラミング的思考力の定量的測定とその分析, 研究報告教育学習支援情報システム(CLE), Vol. 2020-CLE-30, No. 22, pp. 1-4 (2020)
- [4] Pスタディ・P検・Pプラス | ベネッセPシリーズ公式サイト(最終確認:2025年3月)
- [5] 教育の情報化の手引き-追補版-(令和2年6月)第3章(最終確認:2025年9月)
- [6] テスト研究: 領域 - 石井研究室: 名古屋大学 教育発達科学研究科 心理発達科学専攻(最終確認:2025年7月)

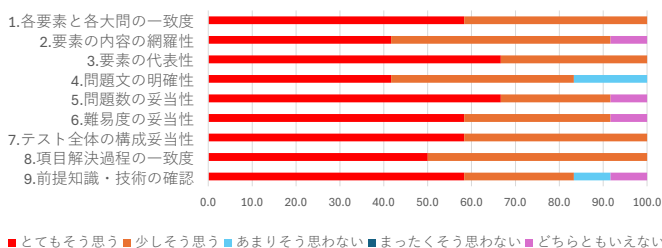


図10 論理的妥当性の質問項目と回答結果

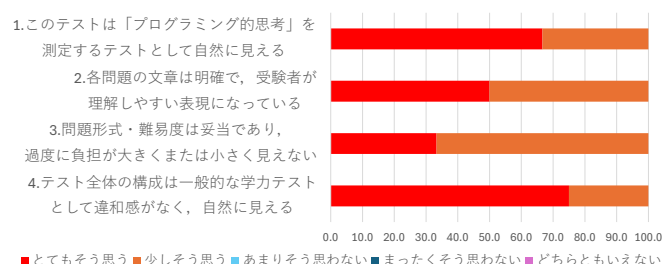


図11 表面的妥当性の質問項目と回答結果