

プログラミング教育と教育コンテンツの可能性

第2報

栗田 るみ子

要約

意思決定は認知心理学では、「ある複数の選択肢 (alternative) の中から、1つあるいはいくつかの選択肢を選択すること」(竹村, 1996, p81) と定義されている。さらに、小橋 (1988) は意思決定は「選択を正当化する理由づけをさがすこと」(p40), 「なぜある選択肢を選んだのかその理由が自他に対して容易に説明できて、その正しさが弁護できることが人間には重要」(p49) であるとしている。これらの指摘をみると、私たちは意思決定の連続の中で行動しているが、熟考を要する場面や合意形成を要する意思決定の場面では、選択の根拠が必要であり、その前提として質の高い「選択肢」の必要性が確認できる。このような合意形成を要する意思決定の場面は、一意解のないもの、すなわち「目標関数や制約条件、問題を構成する変数の相互関係に解釈の任意性や複数の可能性 (曖昧さ) があるもの (松田, 2014)」が考えられ、その中から適切な選択をすることが重要である。こうした意思決定の過程では、複雑な問題をシンプルで抽象度の高いモデルにする能力や作成したモデルから得られる結果を解釈したり、妥当性を評価する能力が求められる。さらには、合意性を得るために、それらの過程や妥当性を説得力のある形で他者に伝える表現力も重要となる。

高等学校の共通教科「情報」の単元「モデル化とシミュレーション」は、こうした意思決定に要する能力の育成を図る機会となりうると思われる。なぜなら、そこでは「問題を抽象化してモデルを作るというモデル化の手法により、問題の分析がしやすくなり、シミュレーションなどの手法を適用できるようになったり、問題解決が行いやすくなったりすることを理解させる」ことが位置付けられているからである (文部科学省, 2010, p29)。

キーワード：シミュレーション、意思決定、プログラミング教育

目的方法

本研究では、高校の共通教科「情報」において、意思決定力の変容を意図する授業デザインを提案し、その有効性を実証的に例証することを目的とする。

そのために、山口・西村 (2015) に提案している数理科学的意思決定力に関する授業デザインのためのフレームワークを参照し、高校「情報」の単元「モデル化とシミュレーション」を対象とする授業デザインのためのフレームワークを作成する。次に、それに基づいて、問題及び授業を設計する。そして、授業を行い、生徒の提案及びその提案の裏付けとなるシミュレーションなどの成果物、自己評価および相互評価の記述、個人レポートにおける振り返りの記述をもとに質的な考察を行い、モデル化とシミュレーションを用い意思決定の活動によって育成される能力の変容を明らかにする。

特に、既習のモデルやシミュレーションの知識や技能を用いる初期段階から、評価活動とその結果に基

づく改善活動，最終提案に至るまでの生徒の変容を捉えることで，本授業デザインの有効性を判断する。

1 普通教科「情報」で学ぶモデル化とシミュレーション

情報科が学習指導要領上の教科となり，2020年で学習指導要領は3回目の改訂である。モデル化とシミュレーションは当初から組み込まれており大きな変化はないものの，各教科書におけるページ分量はかなり多くなっている。この3回の改定において，モデル化とシミュレーションがどのように組み込まれているのか以下にまとめる。

1-1 平成12年（2000年）「情報A」「情報B」「情報C」

2000年3月学習指導要領改訂により情報科が新設され，「情報A」，「情報B」，「情報C」の3科目が新設された。

モデル化とシミュレーション項目

情報A，情報Cには取り扱いがない，「情報B」の(3)「問題のモデル化とコンピュータを活用した解決」で取り扱っている。以下，学習指導要領解説の一文である。

(3) 問題のモデル化とコンピュータを活用した解決

ア モデル化とシミュレーション

身のまわりの現象や社会現象などを通して，モデル化とシミュレーションの考え方や方法を理解させ，実際の問題解決に活用できるようにする。

(内容の取扱い)

内容の(3)については，ソフトウェアやプログラミング言語を用い，実習を中心に扱うようにする。その際，ソフトウェアの利用技術やプログラミング言語の習得が目的とならないようにする。ア及びイについては，基本的な考え方は必ず扱うが，実習については，生徒の実態等に応じ，いずれかを選択して扱うことができる。アについては，内容の(2)のイ，ウ及び(4)のアと関連付けた題材や，時間経過や偶然性に伴って変化する現象などのうち，簡単にモデル化できる題材を扱い，数理的，技術的な内容に深入りしないようにする。

上記において，コンピュータの仕組みと働きの項目とのつながりも述べている。コンピュータにおける情報の処理やコンピュータの仕組み，コンピュータ内部での基本的な処理の仕組み及び簡単なアルゴリズムを理解させる点でシミュレーションの学習へと導く教材となる。また，コンピュータを活用した問題解決では，問題をモデル化してコンピュータ上で扱えるようにすることが可能である。モデル化は，問題を構成している要因とその関係を明確にし，さらに，解決のためには，どの要因を操作して，どの要因がどのような基準を満たすようにすればよいのかを明らかにすることである。

また，コンピュータの活用という観点からは，どのような場面でどのように行えばコンピュータを用いたシミュレーションが有効なのか，また，コンピュータを用いたシミュレーションの特性や活用上の留意点の理解などについて学ぶ必要がある。特に，モデル化の仕方が異なるとシミュレーションの結果が異なることについて認識させることは必要である。同じ課題に異なるモデル化を行い，生徒の分担に

よるシミュレーションを行わせ、その結果を比較させるなどの活動が考えられる。

「情報 B」では、身近な生活に役立つ問題解決能力を育成するのがねらいであり、扱う教材として、例えば、街中の行列のように偶然性によって状態が時系列的に変化していくような現象が想定されるが、モデル化の方法を学ぶことが目的であり、生徒が各教科等で学習済みの知識を生かして簡単にモデル化できる程度の題材を取り上げることが望ましく、他の内容と関係するもの以外として、サイコロやじゃんけんを用いた簡単なゲームにモデル化できる程度の題材や人口変動などの必要なデータが簡単に収集できる程度の題材を扱うことが考えられている。

特徴としては、このときの学習指導要領には「深入りしない」ことが挙げられている。「モデル化とシミュレーション」以外でも同じような考え方であり、教科の新設において、極端に深入りしてしまうことがないようにという注意だったと推測される。

1-2 平成 21 年（2009 年）3 月 「社会と情報」「情報の科学」

3 科目から 2 科目、「社会と情報」と「情報の科学」となった。「社会と情報」は「情報の特徴と情報化が社会に及ぼす影響を理解させ、情報機器や情報通信ネットワークなどを適切に活用して情報を収集、処理、表現するとともに効果的にコミュニケーションを行う能力を養い、情報社会に積極的に参画する態度を育てること」、「情報の科学」は「情報社会を支える情報技術の役割や影響を理解させるとともに、情報と情報技術を問題の発見と解決に効果的に活用するための科学的な考え方を習得させ、情報社会の発展に主体的に寄与する能力と態度を育てること」となっている。

「社会と情報」…記述なし

「情報の科学」… (2)「問題解決とコンピュータの活用」で取り扱う

「情報の科学」の科目のねらいとなっている「問題解決」という内容にはア～ウの 3 項目があり、そのうちのひとつとして扱っている。

具体的なことが書かれている学習指導要領解説について確認するが、問題解決の基本的な考え方、問題の解決と処理手順の自動化、モデル化とシミュレーションに関する基礎的な知識と技能を習得させることをねらいとしている。(ウ)のモデル化とシミュレーションでは、問題を効果的に解決するための方法として、モデル化やシミュレーションに必要な基礎的な知識と技能を習得させる。その際、問題を抽象化してモデルを作るというモデル化の手法により、問題の分析がしやすくなり、シミュレーションなどの手法が適用できるようになったり、問題解決が行いやすくなったりすることを理解させる。そのためには、紙に図示したり、カードの集まりを用いたりしてモデルを表現し、それを手で操作（シミュレーション）させることにより、モデル化の意味と有用性を理解させることが考えられる。また、問題をコンピュータで解く場合には、問題を抽象化して作成したモデルに対してアルゴリズムやシミュレーションを適用することで解決方法を求めることができること、そのモデル化の過程において省略した部分の影響などで解が不正確になる場合もあることなどを理解させ、最終的にはコンピュータによる問題解決とモデル化やシミュレーションとの関係を理解させるとともに、問題解決においてモデル化とシミュレーションの考え方が活用できるようにさせる。そのため、モデル化とシミュレーションに基づいてコンピュータで問題を解決する具体例を体験させるようにする。

モデル化については、例えば鉄道の路線図は、線路が実際にどのように敷設されているかなどについては省略して、駅と駅とを直接つなぐなどして路線を簡略化して示すことで、行き先までの乗り換えを含む経路を見だしやすくしていること、家具の配置図では、個々の家具の高さや質感などは省略して

床面のふさがり具合を見だしやすくしているなどの例を挙げて、その利点を理解させる。その際、既に確立されている定型的なモデルを知識として理解させることだけで終わらないように留意する。

シミュレーションについては、コンピュータを用いたシミュレーションの特性や活用上の留意点について学ぶ。その際、問題解決のどのような場面でどのように活用すれば、コンピュータを用いたシミュレーションが有効かについても理解させる。

モデル化とシミュレーションについては、単にアプリケーションソフトウェアやプログラム言語を使ってモデル化とシミュレーションを行うことに主眼を置くのではなく、問題解決を適切に行うための有効な手段としてモデル化とシミュレーションを取り扱うことが大切である。

「情報B」との比較を見ると、「深入りしない」という言葉が消えている。また、コンピュータの活用が前提の「情報B」と異なり、モデル化の手法とシミュレーションの手法の2つの手法の有用性に重きが置かれた。

1-3 平成30年(2018年)3月 「情報Ⅰ」「情報Ⅱ」

過去の「情報A」「情報B」「情報C」,「社会と情報」「情報の科学」では、いずれかの科目を履修するため、「モデル化とシミュレーション」をすべての高校生が学習しない状況であったが、3回目の改定の大きな特徴として「情報Ⅰ」を共通必修科目として設置した(図表1)。

「情報Ⅰ」では、(3)「コンピュータとプログラミング」の(ウ)に記述がある。

(3) コンピュータとプログラミング

コンピュータで情報が処理される仕組みに着目し、プログラミングやシミュレーションによって問題を発見・解決する活動を通して、次の事項を身に付けることができるよう指導する。

ア 次のような知識及び技能を身に付けること。

(ア)(イ) 略

(ウ) 社会や自然などにおける事象をモデル化する方法、シミュレーションを通してモデルを評価し改善する方法について理解すること。

イ 次のような思考力、判断力、表現力等を身に付けること。

(ア)(イ) 略

(ウ) 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こと。

(内容の取扱い)

(4) 内容の(3)のアの(イ)及びイの(イ)については、(略)アの(ウ)及びイの(ウ)については、コンピュータを使う場合と使わない場合の双方を体験させるとともに、モデルの違いによって結果に違いが出るということについても触れるようにする。

さらに、学習指導要領解説では、「問題解決にコンピュータや外部装置を活用する活動を通して情報の科学的な見方・考え方を働かせて、コンピュータの仕組みとコンピュータでの情報の内部表現、計算に関する限界などを理解し、アルゴリズムを表現し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の機能を使う方法や技能を身に付けるようにし、モデル化やシミュレーションなどの目的

図表 1 情報科目の変遷

改訂前と改訂後の共通教科情報科

- 「情報Ⅰ」は、問題の発見・解決に向けて、事象を情報とその結び付きの視点から捉え、情報技術を適切かつ効果的に活用する力を全ての生徒に育む共通必修履修科目。
- 「情報Ⅱ」は、「情報Ⅰ」の基礎の上に、情報システムや多様なデータを適切かつ効果的に活用する力や、コンテンツを創造する力を育む選択科目。

社会と情報（改訂前）

情報の特徴と情報化が社会に及ぼす影響を理解させ、情報機器や情報通信ネットワークなどを適切に活用して情報を収集、処理、表現するとともに効果的にコミュニケーションを行う能力を養い、情報社会に積極的に参画する態度を育てる。

- (1) 情報の活用と表現
- (2) 情報通信ネットワークとコミュニケーション
- (3) 情報社会の課題と情報モラル
- (4) 望ましい情報社会の構築

情報の科学（改訂前）

情報社会を支える情報技術の役割や影響を理解させるとともに、情報と情報技術を問題の発見と解決に効果的に活用するための科学的な考え方を習得させ、情報社会の発展に主体的に寄与する能力と態度を育てる。

- (1) コンピュータと情報通信ネットワーク
- (2) 問題解決とコンピュータの活用
- (3) 情報の管理と問題解決
- (4) 情報技術の進展と情報モラル

情報Ⅱ（改訂後）

情報に関する科学的な見方・考え方を働かせ、情報技術を活用して問題の発見・解決を行う学習活動を通して、問題の発見・解決に向けて情報と情報技術を適切かつ効果的、創造的に活用し、情報社会に主体的に参画し、その発展に寄与するための資質・能力を次のとおり育成することを目指す。

- (1) 多様なコミュニケーションの実現、情報システムや多様なデータの活用について理解を深め技能を習得するとともに、情報技術の発展と社会の変化について理解を深めるようにする。
- (2) 様々な事象を情報とその結び付きとして捉え、問題の発見・解決に向けて情報と情報技術を適切かつ効果的、創造的に活用する力を養う。
- (3) 情報と情報技術を適切に活用するとともに、新たな価値の創造を目指し、情報社会に主体的に参画し、その発展に寄与する態度を養う。

- (1) 情報社会の進展と情報技術
- (2) コミュニケーションとコンテンツ
- (3) 情報とデータサイエンス
- (4) 情報システムとプログラミング
- (5) 情報と情報技術を活用した問題発見・解決の探究

情報Ⅰ（改訂後）

情報に関する科学的な見方・考え方を働かせ、情報技術を活用して問題の発見・解決を行う学習活動を通して、問題の発見・解決に向けて情報と情報技術を適切かつ効果的に活用し、情報社会に主体的に参画するための資質・能力を次のとおり育成することを目指す。

- (1) 効果的なコミュニケーションの実現、コンピュータやデータの活用について理解を深め技能を習得するとともに、情報社会と人との関わりについて理解を深めるようにする。
- (2) 様々な事象を情報とその結び付きとして捉え、問題の発見・解決に向けて情報と情報技術を適切かつ効果的に活用する力を養う。
- (3) 情報と情報技術を適切に活用するとともに、情報社会に主体的に参画する態度を養う。

- (1) 情報社会の問題解決
- (2) コミュニケーションと情報デザイン
- (3) コンピュータとプログラミング
- (4) 情報通信ネットワークとデータの活用

引用：文部科学省 学習指導要領解説情報 P6 より

に応じてコンピュータの能力を引き出す力を養う。」とある。このような活動を通して、問題解決にコンピュータを積極的に活用しようとする態度、結果を振り返って改善しようとする態度、生活の中で使われているプログラムを見いだして改善しようとするなどを通じて情報社会に主体的に参画しようとする態度を養うことが考えられる。

ここでは、中学校技術・家庭科技術分野の内容「D 情報の技術」の学習を踏まえたプログラミングを扱う。コンピュータでの情報の内部表現や情報の抽象化、情報デザインについては、共通教科情報科の「情報I」の2の(2)「コミュニケーションと情報デザイン」の内容と関連付けて扱う。

次に、ア(ウ)社会や自然などにおける事象をモデル化する方法、シミュレーションを通してモデルを評価し改善する方法について理解することでは、モデル化とシミュレーションを身近な問題を発見し解決する手段として活用するために、実際の事象を図や数式などにモデル化して表現する方法、モデル化した事象をシミュレーションできるように表現し条件を変えるなどしてシミュレーションする方法、作成したモデルのシミュレーションを通じてモデルを改善する方法を理解するようにする。その際、モデルの違いによってシミュレーションの結果や精度が異なる場合があることを理解するようにする。

また、イ(ウ)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ことでは、モデル化とシミュレーションの考え方を様々な場面で活用するために、モデル化とシミュレーションを問題の発見や解決に役立てたり、その結果から問題の適切な解決方法を考えたり選択したりする力を養う。

その際、学校や地域の実態及び生徒の状況に応じて、数学科と連携し、不規則な現象を含む確率的モデルを扱うことも考えられる。

2 モデル化とシミュレーション指導

現実の事象をモデル化してシミュレーションする活動を取り上げ、現実の事象を抽象化することでコンピュータが扱える形に表現するモデル化のメリットや抽象化に起因するモデル化の限界、シミュレーション結果から予測を行ったり最適な解決方法を検討したりすることなどを扱う。その際、学校や地域の実態及び生徒の状況に応じて、プログラミング、シミュレーション専用ソフトウェア、表計算ソフトウェアの利用などシミュレーションを行う方法について配慮する。また、数式を利用したモデル化とシミュレーションを取り上げ、金利計算、人口の増減、インフルエンザの流行、数学や物理などの事象を扱うことなどが考えられる。

学習活動の例としては、コンピュータや外部装置についての仕組みや特徴、モデル化とシミュレーションの考え方などを学んだ後に、生徒の希望する問題についての学習を深める中で、アルゴリズムやプログラミングなどについて自ら学び、問題の発見・解決に必要な資質・能力を獲得することなどが考えられる。

更に、モデル化とシミュレーションに関する学習活動としては、平面図等を利用した家具の配置等の単純なモデルによるシミュレーションやシミュレーションソフトウェアを利用した体験を通して、事象をどのようにモデル化しているのかを調べたり、生徒自らがモデル化を行ったり、モデル化の長所と短所を調べたりする学習活動などが考えられる。

学校や地域の実態及び生徒の状況に応じて乱数を用いたシミュレーションなどを題材とするとともに、インフルエンザが爆発的に増える理由、感染を抑えるための方法について考えるような題材を基にモデル化とシミュレーションを行う学習活動などが考えられる。

学習指導要領が改訂されるにあたり、育成すべき資質・能力についての議論があり、3つの柱として「何を知っているか、何ができるか（個別の知識・技能）」、「知っていること・できることをどう使うか（思考力・判断力・表現力等）」、「どのように社会・世界と関わり、よりよい人生を送るか（学びに向かう力、人間性等）」に整理された。学習指導要領自体では、「知識・技能」と「思考力・判断力・表現力」それぞれについて示され、「科学的な理解に裏打ちされた情報活用能力の育成」が情報科の柱になっているといえる。

上記から、モデル化とシミュレーションについてまとめる。

学習活動…【問題解決にコンピュータなどを活用する活動】

知識・技能…【社会や自然などにおける事象をモデル化する方法、シミュレーションを通してモデルを評価し改善する方法】

具体的には【実際の事象を図や数式などにモデル化して表現する方法、モデル化した事象をシミュレーションできるように表現し条件を変えるなどしてシミュレーションする方法、作成したモデルのシミュレーションを通じてモデルを改善する方法についての理解】

思考力・判断力・表現力…【目的に応じたモデル化やシミュレーションを適切に行い、その結果を踏まえて問題の適切な解決方法を考える】

学びに向かう力・人間性等…【問題解決にコンピュータを積極的に活用しようとする態度、結果を振り返って改善しようとする態度、生活の中で使われているプログラムを見いだして改善しようとする事などを通じて情報社会に主体的に参画しようとする態度】

となる。コンピュータとの関わりについては【モデル化やシミュレーションなどの目的に応じてコンピュータの能力を引き出す力を養う】と書かれており、コンピュータを活用することが求められる。はじめからコンピュータを活用するだけではなく、【コンピュータを使う場合と使わない場合の双方を体験させる】というようにコンピュータを使わない場合について体験を通して理解することも求められる点で、「情報の科学」と同等と考えられる。図表2に示すように、このような活動から各教科書では、様々なプログラム言語を取り入れている。

図表2 情報1の教科書で使われているプログラミング言語

	VBA	JavaScript	Physon	Scratch
実教出版	○	○	○	
東京書籍		○	○	○
日本文教出版		○	○	○
数研出版	○	○	○	○
第一学習社	○			○

3 モデル化とシミュレーション実習例

本研究の目的の1つであるモデル化とシミュレーション教育が問題解決思考の育成につながる点に着目しているため、授業モデルの演習課題を述べる。以下2点は学部における情報科目での課題例である。

3-1 例題 1

乱数によるさいころの目の発生プログラム

図表 3 例題 1 さいころプログラム解説

<p>JavaScript で乱数を発生させる方法 JavaScript では Math.random () 関数で乱数を発生させる。</p>	<p><i>Math.random ()</i></p>																											
<p>Math.random () は 0 ~ 0.999... の範囲でランダム値を返す、一様乱数を使う。サイコロ (6 面ダイス) のように 1 ~ 6 の範囲で乱数を得たい場合、ランダム関数で実現可能となる。</p>	<pre>let min = 1; let max = 6; let length = max - min + 1; let value = Math.floor (Math.random () * length) + min; console.log (value);</pre>																											
<p>最小値を 1, 最大値を 6 で、6 パターンの結果なので、(6 - 1 + 1) で length を 6 Math.random () の値と length をかけ算。 小数点を切り捨て、最小値の値を足す (今回は 1)。</p>	<p>サイコロの値を再現</p> <table border="1" data-bbox="730 860 1396 1099"> <thead> <tr> <th>Math.random()の値</th> <th>6 を掛けた値</th> <th>小数点を切り捨てた値</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0.1</td> <td>0.6</td> <td>0</td> <td>1</td> </tr> <tr> <td>0.2</td> <td>1.2</td> <td>1</td> <td>2</td> </tr> <tr> <td>0.5</td> <td>3</td> <td>3</td> <td>4</td> </tr> <tr> <td>0.9</td> <td>5.4</td> <td>5</td> <td>6</td> </tr> </tbody> </table>				Math.random()の値	6 を掛けた値	小数点を切り捨てた値	1	0	0	0	1	0.1	0.6	0	1	0.2	1.2	1	2	0.5	3	3	4	0.9	5.4	5	6
Math.random()の値	6 を掛けた値	小数点を切り捨てた値	1																									
0	0	0	1																									
0.1	0.6	0	1																									
0.2	1.2	1	2																									
0.5	3	3	4																									
0.9	5.4	5	6																									
<p>サイコロを再現する関数の作成 折角なのでサイコロを再現する関数を作ってみます。汎用性を持たせるために引数を指定すれば 10 面ダイスなどにも対応できるようにしておきます。</p>	<pre>function random (min = 1, max = 6) { let length = max - min + 1; let value = Math.floor (Math.random () * length) + min; return value; }</pre>																											
<p>サイコロを再現する関数の作成 (配列版) 作成した関数をループ処理で複数回呼び出す</p>	<pre>function random_values (min = 1, max = 6, number = 1) { let result = []; for (let i = 0; i < number; i++) { result.push (random (min, max)); } return result; }</pre>																											
<p>サイコロ関数の動作確認 (100 回)</p>	<pre>console.log (random_values (1, 6, 100));</pre>																											
<p>結果を度数分布表で確認 (サイコロを振る回数を変更し実験を繰り返す)</p>	<pre><script src="https://cdn.plot.ly/plotly-latest.min.js"></script> <script> function random(min = 1, max = 6) { let length = max - min + 1; let value = Math.floor(Math.random() * length) + min; return value; } function random_values(min = 1, max = 6, number = 1) { let result = []; for (let i = 0; i < number; i++) { result.push(random(min, max)); } return result; } </script></pre>																											

```

function plot() {
  let deme = random_values(1, 6, 10000);
  let result = {
    1:0,
    2:0,
    3:0,
    4:0,
    5:0,
    6:0
  }

  for (let i = 0; i < deme.length; i++) {
    result[deme[i]]++;
  }

  // グラフ
  let graph = "myDiv";
  let layout = {
    height: 400,
    width: 400,
    showlegend:false,
    title:"シミュレーションの結果",
    xaxis: {
      title: "出る目の数"
    },
    yaxis: {
      title: "回数"
    },
  };
  let x_memori = ["1","2","3","4","5","6"];
  let trace = {
    x: x_memori,
    y: Object.values(result),
    type: "bar"
  };
  let data = [trace];

  Plotly.newPlot(graph, data, layout);
}
</script>
</head>
<body onload="plot()">
  <div id="myDiv"></div>
</body>

```

変数定義
100回振った結果を deme 配列に格納
値が何回ずつ出たのかを集計（連想配列 result
を使う）
全てを 0 回として初期化

```

let deme = random_values (1, 6, 100);
let result = {
  1:0,
  2:0,
  3:0,
  4:0,
  5:0,
  6:0
}

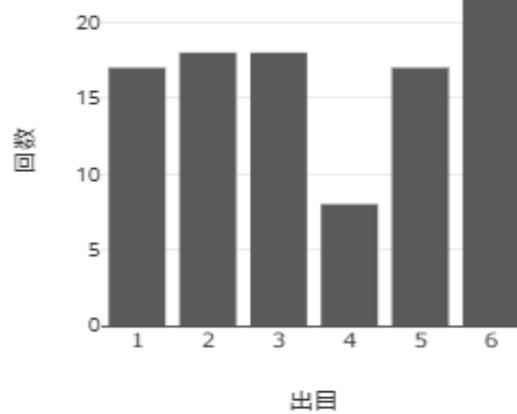
```

ループ処理 出目配列を元にループを行う 1～6の値の出た数の集計 (result 連想配列に集計)	<pre>for (let i = 0; i < deme.length; i++) { result[deme[i]]++; }</pre>
グラフ作成 配列は x 軸 (トレース用の連想配列を用意)	<pre>let x_memori = ["1", "2", "3", "4", "5", "6"]; let trace = { x: x_memori, y: Object.values (result), type: "bar" }; let data = [trace];</pre>

実行結果は、ランダム関数の結果であるため、何回でも実行し変化を確認できる。

図表 4 実行結果グラフ

サイコロ・シミュレーション



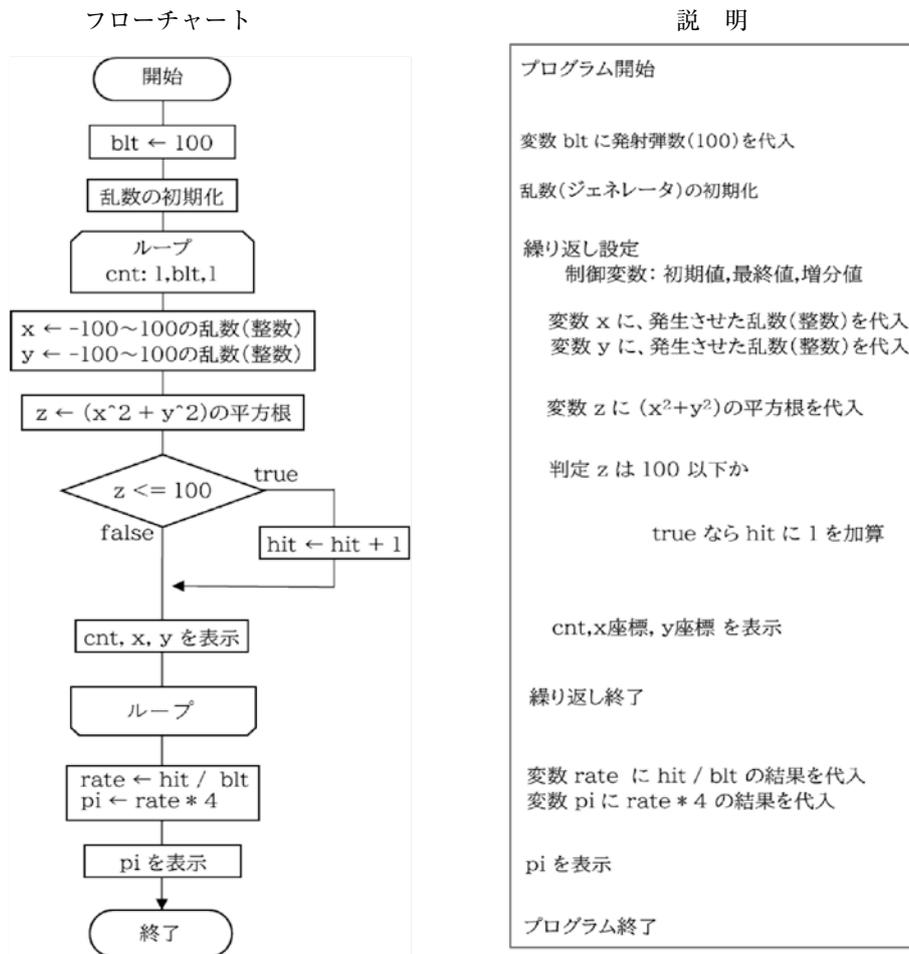
3-2 例題 2

本実習課題は、乱数を使った 250 発の弾を打ち込みプログラムである。

図表 5 モンテカルロ法円周率プログラム実行結果

250 発すべてが正方形の中に入るとして、そのうち円内に命中した弾の数を T とすると、その比率は、正方形と円の「面積比」と同じはずということを前提としている。 $T : 250 = \pi r^2 : 4r^2$ $250\pi r^2 = 4r^2 T$ $\pi = 4r^2 T / 250r^2$ $\pi = T / 250 \times 4$

図表 6 フローチャート



モンテカルロ法で円周率 VBA プログラムソースコード

Rem ■モンテカルロ法で円周率

' 試行回数 100 とし, 試行Noと円周率を表示する
' DoEvents 関数を利用

Sub piMontecarlo_2 ()

```

Dim blt As Long           ' 発射弾数上限
Dim cnt As Long          ' 弾数カウンター
Dim x As Integer         ' X 座標
Dim y As Integer         ' Y 座標
Dim z As Long            ' 斜辺の長さ
Dim hit As Long          ' 命中弾数
Dim rate As Double       ' 命中率
    
```

```

Dim pi As Double          ' 円周率近似値

Dim sTime As Double      ' 開始時刻
Dim eTime As Double      ' 終了時刻

Dim tst As Integer       ' 試行回数
Dim i As Integer         ' 試行回数カウンター
Dim sho As Double        ' 円周率の最小値
Dim dai As Double        ' 円周率の最大値
Dim total As Double      ' 円周率の累積値
Dim heikin As Double     ' 円周率の平均値

sTime = Timer ()
tst = 100
blt = 10000

Randomize                ' 乱数ジェネレータの初期化

For i = 1 To tst
    DoEvents              ' CPU の制御を OS に開放
    ' 弾の発射を繰り返す
    For cnt = 1 To blt
        x = WorksheetFunction.RandBetween (-100, 100) ' x 座標の取得
        y = WorksheetFunction.RandBetween (-100, 100) ' y 座標の取得
        z = Sqr (x ^ 2 + y ^ 2) ' 斜辺の長さを求める
        ' 命中判定
        If z <= 100 Then
            hit = hit + 1 ' 命中数のカウントアップ
        End If

        ' Debug.Print cnt & "(" & x & ", " & y & ";" )

    Next cnt              ' 次の発射へ

    ' 命中率を求めて円周率を計算
    rate = hit / blt
    pi = rate * 4

    ' 試行Noと円周率の表示

```

```

Debug.Print i & vbTab & pi
total = total + pi
heikin = total / i

' 円周率の最小値, 最大値を記録
If i = 1 Then      ' 試行 1 回目なら
    sho = pi        ' sho に pi を代入
    dai = pi        ' dai にも pi を代入
Else              ' それ以外の場合には
    If pi < sho Then sho = pi    ' pi<sho なら sho に pi を代入
    If pi > dai Then dai = pi    ' pi>dai なら dai に pi を代入
End If

hit = 0          ' 命中数を初期化 (次の試行に備えて)

Next i          ' 次の試行へ

' 発射弾数, 円周率の範囲, 円周率の平均を表示
Debug.Print "発射弾数" & blt
Debug.Print "値の範囲" & sho & "<  $\pi$  <" & dai
Debug.Print "円周率平均" & heikin

eTime = Timer    ' 終了時刻の取得

Debug.Print "所要時間" & eTime - sTime & "秒" ' 所要時間の表示

End Sub

```

Rem ■モンテカルロ法で円周率

```

' ワークシートにも反映させる
' ピボットテーブルの更新を可能にする (範囲変更への対応を含む)
' VBA からデータ分析ツール「基本統計量」を利用する
' ワークシートで試行回数, 発射弾数を指定する方式
' エラー処理

```

```
Sub piMontecarlo_3 ()
```

```
    ' 変数宣言
```

```
    Dim tst As Integer          ' 試行回数
```

```

Dim i As Integer      ' 試行回数制御カウンター
Dim blt As Long      ' 発射弾数
Dim cnt As Long      ' 発射弾数制御カウンター
Dim x As Integer     ' X 座標
Dim y As Integer     ' Y 座標
Dim z As Integer     ' 斜辺の長さ
Dim hit As Long      ' 命中弾数
Dim rate As Double   ' 命中率
Dim pi As Double     ' 円周率近似
Dim total As Double  ' 円周率の累積
Dim heikin As Double ' 円周率の平均
Dim sTime As Double  ' 開始時刻
Dim eTime As Double  ' 終了時刻

sTime = Timer ()     ' 開始時刻格納

' シート「VBA 版」をアクティブに
Sheets ("DebugPrint 版").Activate

' 出力データのクリア
Range ("B5:D6").ClearContents
Range ("A9").CurrentRegion.Offset (1).ClearContents
Range ("H2:I17").ClearContents

' エラー処理
On Error GoTo ErrLabel
' 試行回数と発射弾数をワークシートから参照する方式へ
tst = Range ("B3").Value
blt = Range ("B4").Value

Randomize            ' 乱数ジェネレータの初期化

For i = 1 To tst
DoEvents            ' VBA プログラムが占有している制御を OS に開放する関数

' 弾の発射を繰り返す
For cnt = 1 To blt
    x = Int (Rnd () *201-100) ' x 座標の取得
    y = Int (Rnd () *201-100) ' y 座標の取得

```

```

z = Sqr (x ^ 2 + y ^ 2)      ' 斜辺の長さを求める
' 命中判定
If z <= 100 Then
    hit = hit + 1            ' 命中数のカウントアップ
End If
' Debug.Print cnt & "(" & x & ", "& y;"")
Next cnt                    ' 次の発射へ

' 命中率を求めて円周率を計算
rate = hit / blt
pi = rate * 4
' Debug.Print i & vbTab & pi

Cells (60000, 1).End (xlUp).Offset (1).Value = i
Cells (60000, 2).End (xlUp).Offset (1).Value = pi

total = total + pi

If i = 1 Then              ' 試行 1 回目なら無条件に sho に pi を代入
    sho = pi                ' sho に pi を代入
    dai = pi                ' dai に pi を代入
Else                        ' それ以外なら
    If pi < sho Then sho = pi ' pi が sho より小さかったら pi を sho に再代入
    If pi > dai Then dai = pi ' pi が dai より大きかったら pi を dai に再代入
End If

heikin = total / i        ' 円周率の平均

hit = 0                    ' 命中数を初期化
Range ("C3").Value = tst - i ' 残り回数の表示

Next i

' Debug.Print "値の範囲" & sho & "< π <" & dai
' Debug.Print "円周率平均" & heikin

Range ("B5").Select
With Selection
    .Value = sho

```

```
.Offset (, 1).Value = "< π <"
.Offset (, 2).Value = dai
.Offset (1).Value = heikin
End With

' データソースを変更する
On Error GoTo ErrLabel

' ピボットテーブルのデータ範囲が異なる場合への対処
Dim myDB As Range

' データ範囲を再取得
Set myDB = Worksheets ("DebugPrint 版").Range ("A8").CurrentRegion
ActiveSheet.PivotTables (1).ChangePivotCache _
    ActiveWorkbook.PivotCaches.Create (xlDatabase, myDB)

' ボットテーブルを更新する
ActiveSheet.PivotTables (1).PivotCache.Refresh

' VBA から分析ツール「基本統計量」を利用する
' セル B8 から続く円周率のデータ領域の最後の行番号を取得
Dim r As Integer
r = Cells (8, 2).End (xlDown).Row
Dim myDS As Range
Set myDS = Range (Cells (8, 2), Cells (r, 2))

' 記述統計アドインの実行
Application.Run "ATPVBAEN.XLAM!Descr", myDS, Range ("H2"), "C", True, True

eTime = Timer () ' 終了時刻を格納
Range ("D3").Value = eTime - sTime & "秒" ' 所要時間を表示

Exit Sub ' エラーなしの場合ここで終了 (必須)

ErrLabel: ' エラー処理用ラベル
MsgBox "試行回数と発射弾数を「整数」で指定してください"
Range ("B3").Select

End Sub
```

プログラム実験結果

試行回数は100回からスタートし発射弾数を増やしながら実行すると以下のようにイミディエイトウィンドウ⁽¹⁾に表示される。

図表7 実行結果

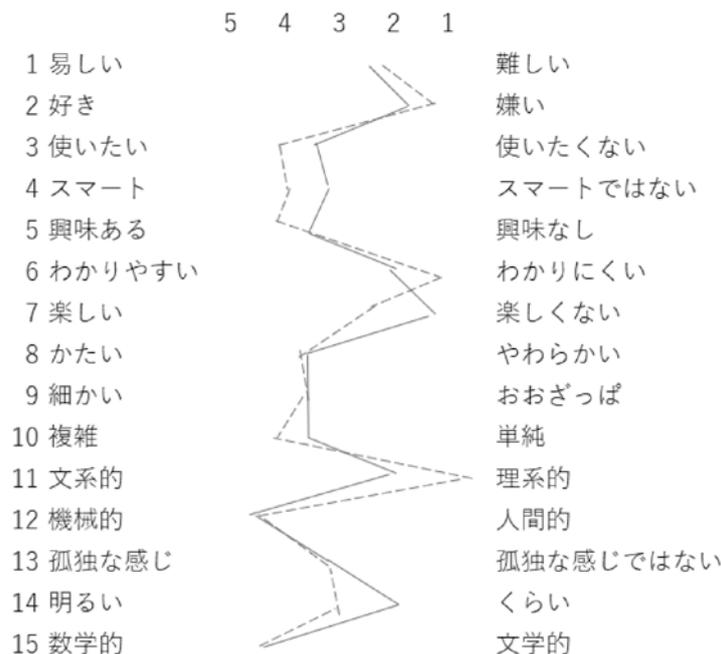
<pre>イミディエイト 97 3.24 98 3.4 99 3.12 100 3 発射弾数 100 値の範囲 2.76<π<3.56 円周率平均 3.1424 所要時間 0.3359375</pre> <p style="text-align: center;">10²=100</p>	<pre>イミディエイト 97 3.14 98 3.244 99 3.128 100 3.132 発射弾数 1000 値の範囲 3.048<π<3.3 円周率平均 3.1538 所要時間 1.6015625</pre> <p style="text-align: center;">10³=1000</p>
<pre>イミディエイト 97 3.1396 98 3.1456 99 3.1396 100 3.1188 発射弾数 10000 値の範囲 3.1036<π<3.1732 円周率平均 3.14156 所要時間 14.0546875</pre> <p style="text-align: center;">10⁴=10000</p>	<pre>イミディエイト 97 3.14164 98 3.14456 99 3.13344 100 3.14796 発射弾数 100000 値の範囲 3.12956<π<3.16008 円周率平均 3.1432988 所要時間 141.515625</pre> <p style="text-align: center;">10⁵=100000</p>
<pre>イミディエイト 97 3.13998 98 3.1412 99 3.144684 100 3.143976 発射弾数 1000000 値の範囲 3.139512<π<3.148556 円周率平均 3.14396332 所要時間 1403.5234375</pre> <p style="text-align: center;">10⁶=1000000</p>	

4 調査

本研究では、情報科の教員免許を取得する学生15名を対象に、プログラミングに対するイメージについて15項目からなる調査を授業の前後に実施した。質問形式は形容詞の対を与え、5反応形式の回答とした。図表8はプログラミングイメージについて、項目別の平均値を求めた結果である。

(1) イミディエイトウィンドウとは、実行経過を即時に確認できる窓で、作業効率を上げる便利なツール。

図表8 プログラミングに対するイメージ



(実線は実習前, 点線は実習後)

プログラミングに対するイメージは全体的にマイナスからプラスの方向へ変化している。特に「使いたい」「スマート」「興味ある」は、プラスイメージへの変化がやや大きい。これはプラスのイメージが強まったというより、新しく始めるプログラミング学習に対する期待感が大きいためと思われる。学習後に「わかりにくい」「複雑な」のイメージが強くなっている。また、プログラミング学習に対する意欲について学習前後で比較してみると、平均値は全体的に下がっているが学習終了後も半数以上の学生が身につけたいと思っている。プログラミングを用いてシミュレーションを学習するための意識付けには、理系的発想を組み込む必要があると思われる。また、質問項目を広く教科横断的なイメージを組み込むことで、人間的因子や機械的因子の抽出などが期待されると推測する。今後、文系科目理系科目の教科を目指す学生に広く意識調査を進める。

おわりに

プログラムを組み学習する点で効果的なことは、条件を整え、要素に分けて解析し、モデルを作成することで、シミュレーションを実装できる形にする能力と共に、減少を可視化できる点である。プログラムは短い簡単なものから取り組むことが重要であるが、基本的なアルゴリズムの理解としては、モデルをシミュレーションに実装できる形に表現することに重点をおくことである。つまり、情報手段を用いた処理や思考の過程を表現することを優先させることで、問題解決としてのPDCA等のサイクルとして捉えることを重視し、解釈、評価に加えて改善も意識させるのである。

参考文献

- [1] 山口武志, 西村圭一, (2015), 授業実践による数理科学的意決定力に関する水準表の記述性および規範性の検証, 第42回数学教育論文発表会
- [2] 竹村和久, (2005), 意決定現象と行動意決定論, ファジィ学会誌 Vol. 17, No6. pp. 646-654
- [3] リンダ・グラットン, アンドリュー・スコット, 池村千秋訳 (2016), 『LIFE SHIFT — 100年時代の人生戦略』, 東洋経済新報社
- [4] アルビン・トフラー, 鈴木健次他訳, (1980), 『第三の波』, 日本放送出版協会
- [5] 後藤貴裕他, (2016), 『高等学校情報科において乱数シミュレーションによるモデル化を通じた数理科学的意決定能力の育成を図る授業実践の事例報告』, 科学教育研究 Vol. 40, pp. 198-208
- [6] 栗田るみ子他共著, (2008), 『情報科教育法』, 学文社, pp. 40-45
- [7] 栗田るみ子, (2019), 『インターネットを利用した新しい教育法と評価を学ぶ』, 城西大学更新制講習テキスト, pp. 4-8
- [8] 栗田るみ子, 白鳥義明, (2021), 『コンピュータと会話しながらプログラミングに挑戦』, 城西大学更新制講習テキスト, pp. 22-27
- [9] 丸山雅貴, 森田裕介, (2020), 「科学教育の研究における STEM/STEAM 教育を指向した取り組みの動向に関する整理」, 日本科学教育学会年会論文集, pp. 289-292
- [10] 未来の学びコンソーシアム, <https://miraino-manabi.jp/> (2022/7access)
- [11] 統計ダッシュボード, <https://dashboard.e-stat.go.jp/> (2022/7access)
- [12] なるほど統計学園, 総務省統計局, <https://www.stat.go.jp/naruhodo/index.html> (2022/7access)

A Study of Programming Education and the Educational Content Potentiality Part 2

Rumiko KURITA

Abstract

In cognitive psychology, decision-making is defined in several ways. For example, Takemura (Takemura, 1996, p. 81) defined decision-making as “selecting one or several alternatives from among several alternatives” (Takemura, 1996, p. 81). Furthermore, Kobashi (1988) defined decision-making as “looking for a reason to justify a choice” (p. 40). That is, we can consider our daily life to be a series of decision-making processes. However, in the case of decision-making that requires deliberation or consensus building, a basis for selection is necessary, and we can confirm that the necessity of high-quality “alternatives” is a prerequisite.

In the explanation of the Course of Study on “Information” in high schools, it is stated that “by creating models by abstracting problems, students can understand that the modeling method makes it easier to analyze problems, apply simulation and other methods, and solve problems” (MEXT, 2010, p. 29).

In this study we propose a class design intended to change the decision-making power and demonstrate its effectiveness on “Information” that is the common subject of the high school. That purpose, based on the framework for class design on mathematical scientific decision-making power proposed by Yamaguchi and Nishimura (2015), we create a framework for class design that targets “modeling and simulation”, the unit of “Information”.

In particular, we judge the effectiveness of this class design by grasping the transformation of students from the initial stage to the final stage. Here, the initial stage is the stage in which students use the knowledge and skills of models and simulations, and the final stage is the stage in which evaluation activities, improvement activities based on the results, and final proposals are made.

Keywords: programming education, simulation techniques, Decision making