

主体的・対話的で深い学びを実現する単元構想（商業科「プログラミング」）
 単元名： 2年生 第3章 第2節 配列の利用 「配列の探索」（全4時間）

単元の目標（育成を目指す資質・能力）
知識及び技術 ・配列のアルゴリズムについて理解するとともに、フローチャートやコーディングの技術を身に付けている。
思考力、判断力、表現力等 ・プログラムと情報システムの開発のオブジェクト指向型言語の利用に関する課題を発見し、論理的な根拠に基づいて、情報を処理する手順を見だし、評価、改善することができる。
学びに向かう力、人間性等 ・アルゴリズムを通して、情報システムの開発について自ら学び、主体的かつ協働的に多方面から課題を捉え、考察しようとしている。

既習内容	生徒の実態と抱える課題
（2年1学期） ・フローチャートの作成及び一連の流れ ・フローチャートのトレース （2年2学期） ・Javaによるコーディング	（実態） プログラム全体を一つの流れとしてとらえることよりも、解答を探すことを優先してしまうため、知識の定着につながっていない。コーディングに対しても苦手意識をもつ生徒が多い。 （教師の願い） さまざまな視点でプログラムを捉え、解決策の糸口を見つけ、アルゴリズムを考案させることで、理解を深めるとともに、達成感を味わわせたい。

<p>主体的・対話的で深い学びの視点に立った単元構想（研究仮説と手だて）</p> <p>本単元では、配列の探索を利用したアルゴリズムについて考え、表現することが重要である。データを見つけることだけが目的ではなく、その後の処理方法をいかに正確に読み取り実行できるかが大切であることに気付かせることで、プログラムを一つの流れとして捉えられるようになると考えた。</p> <p>また、配列の探索の考え方について、グループでの発表やまとめによる振り返りをさせることで、学びを確かなものにさせたい。既習内容を生かし、演習問題を活用し、発展的な内容についてグループワークをさせることで、配列の利用についてさまざまな考え方があることや、異なったアプローチの仕方があることなどに気付かせていきたい。</p> <p>【主体的な学び】</p> <ul style="list-style-type: none"> ・授業の内容をプリントにまとめさせる（第1時）。 ・まとめたプリントを基に、振り返りをさせ、処理条件や特徴等を確認させる（第2時）。 ・問題に対しての根拠を考えさせる（第3時）。 <p>【対話的な学び】</p> <ul style="list-style-type: none"> ・自分の考えをまとめ他者に伝え、協力しながらフローチャートを作成させる（第2時）。 ・自分とは違う解答への道筋に気付き、他者の考えを取り入れ、解答の幅を広げられるようにさせる（第3時、第4時）。 <p>【深い学び】</p> <ul style="list-style-type: none"> ・フローチャートからコーディングに書き起こし、プログラムを一つの角度からではなく、多方面から読み取れるように考えさせる（第4時）。

単元計画						
段階	小単元の目標	時	本時の目標	学習活動	主体的・対話的で深い学びの視点に立った授業の工夫	育成を目指す資質・能力や目指す生徒像
基礎基本の学習	・配列の考え方、配列による集計方法を身に付ける。	1	・配列の考え方、配列による集計方法をまとめる。	・配列の考え方や配列による集計方法についての考え方を理解する。 ・配列の必要性、添字の役割を整理し理解する。	㊦ 授業の内容を自分の言葉でプリントにまとめさせる。	・配列の内容について自分の言葉でまとめ、考え方や特徴等を理解することができる。
課題解決	・配列の探索を考え、その他の処理についても、読み解けるようにする。	2	・配列と添字の関係性や特徴を正しく理解する。	・配列の集計方法についてグループで話し合い、フローチャートを考えながら作成する。	㊦ 自分の考えを他者に伝え、協力しながらフローチャートを作成させる。	・自分の考えをもち、他社に伝えることができる。
		3	・探索の処理を読み解く力を身に付け、他者に伝える。	・配列の探索とそれ以外のアルゴリズムを含んだ問題の解答をロイロノート・スクールに提出し、他者の意見と比較する。	㊦ 自力解答する中で解答の根拠を考えさせる。 ㊦ 自分の考えの中に他者の考えを取り入れることで、解答の幅を広げさせる。	・解答の根拠を考えることができる。 ・他者の意見を取り入れ、よりよい解答に導こうとしている。
振り返り・発展学習	・プログラム全体を一つの流れとして捉えられるように考える。	4	・プログラムを多方面から思考することで、全体の流れを捉えられるようにする。	・グループで問題に取り組んだ結果を発表し、互いに評価する。 ・これまでに取り組んだフローチャートを参考に、コーディングを考える。	㊦ 前時で取り組んだ問題の解答を説明させる。 ㊦ フローチャートからコーディングに書き起こすことで、プログラムの流れを整理させる。	・根拠を示して問題の解答を発表できる。 ・フローチャートをコーディングに置き換えることができる。