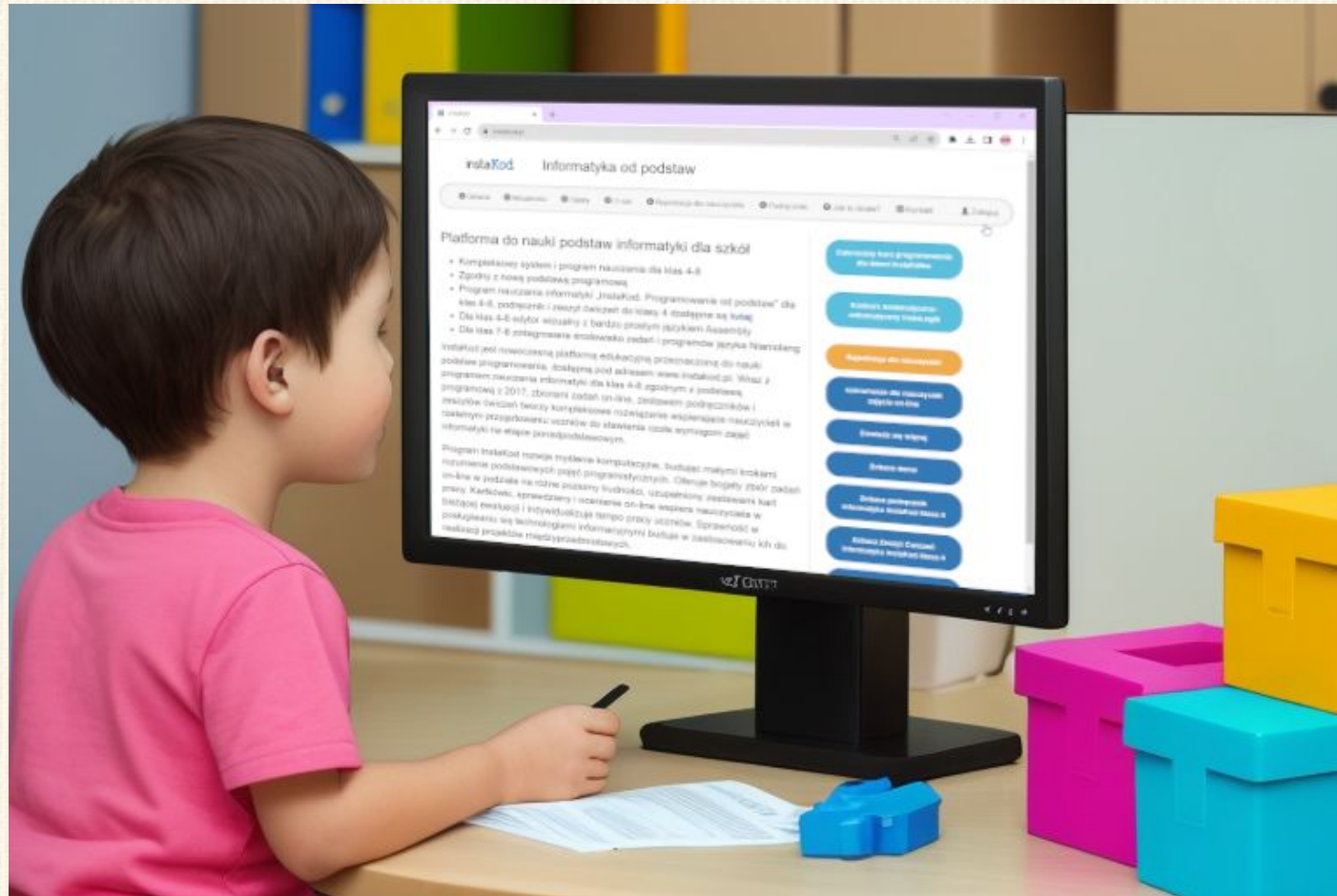


Podstawy programowania dla uczniów klas 4-6

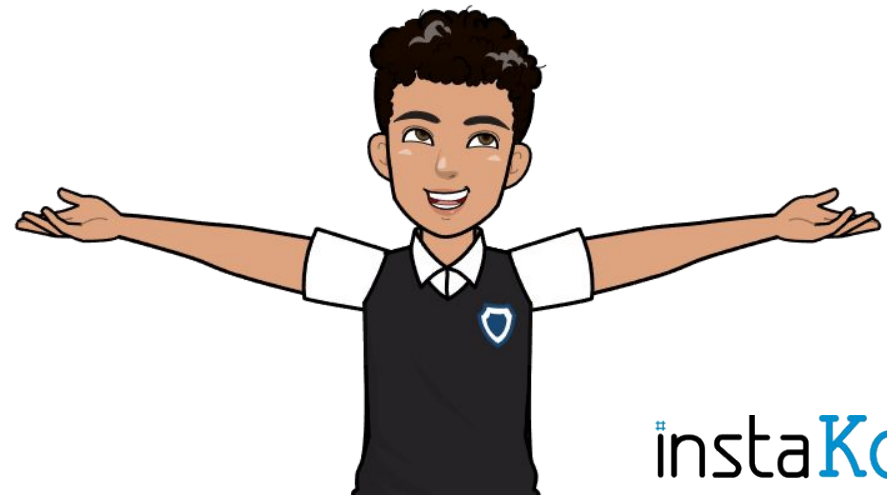
Metodyka



Podstawa programowa SP klasy I-III Informatyka

Na tym etapie edukacyjnym uczniowie poznają podstawowe znaczenie wybranych pojęć związanych z informatyką:

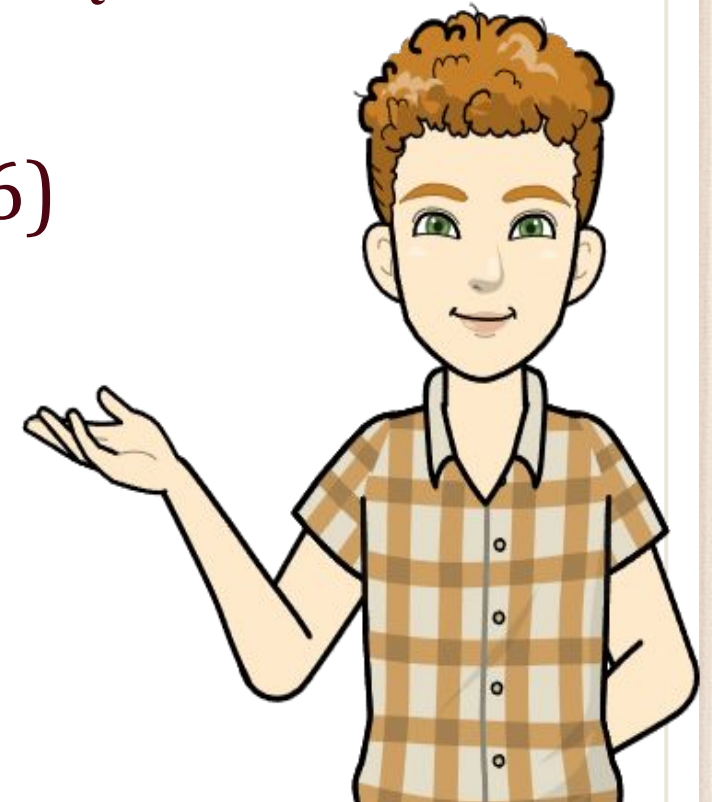
- ❖ liniowa kolejność (sekwencja zdarzeń, logiczny porządek zdarzeń, czynności i wielkości)
- ❖ instrukcja (polecenie)
- ❖ algorytm (plan działania)



Podstawa programowa SP klasy IV-VIII Informatyka

Na drugim etapie edukacyjnym uczniowie poznają wybrane pojęcia związane z informatyką i wybrane algorytmy,

- ❖ pracując w językach blokowych (4-6)
- ❖ tekstowych (7-8)



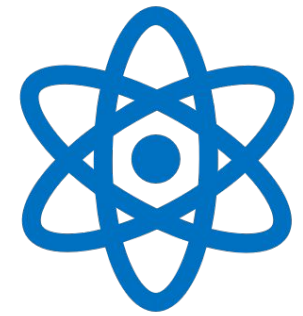
Podstawa programowa SP klasy IV-VIII Informatyka

Klasy 4-6	Klasy 7-8
Język blokowy Assembly	Język tekstowy
Pojęcie zmiennej Wejście i wyjście z programu Stan wartości zmiennych w danej chwili Operacje arytmetyczne Instrukcja warunkowa Pętla	Pojęcie zmiennej Wejście i wyjście z programu Stan wartości zmiennych w danej chwili Operacje arytmetyczne Instrukcja warunkowa Pętla Tablice i napisy (zapoznanie się z pojęciami) Funkcje i wybrane algorytmy (zapoznanie się)

Jacy są uczniowie dziś?

- ❖ Są świetnie zaznajomieni z przeszukiwaniem, chatowaniem, tekstowaniem, graniem, kreowaniem swojego wizerunku w sieci.
- ❖ Z łatwością zaczynają korzystać z nowych aplikacji, szczególnie modnych w środowisku.
- ❖ Mają mnóstwo doświadczenia w interakcji z nowymi technologiami, ale dużo mniej z tworzeniem w nowych technologiach.





Jak uczyć programowania?

Stworzyć uporządkowaną przestrzeń edukacyjną

- ❖ Nie dającą szansy na wejście w rolę biernego „clickera”

Zdefiniować sukces

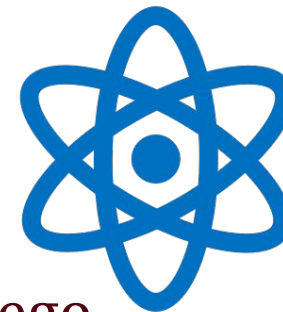
- ❖ Udostępnić zadania, których rozwiązanie można jednoznacznie określić jako poprawne lub niepoprawne

Zdefiniować kreatywność

- ❖ Kreatywność jako umiejętność tworzenia rozwiązań rzeczywistych problemów

Potrzebujemy

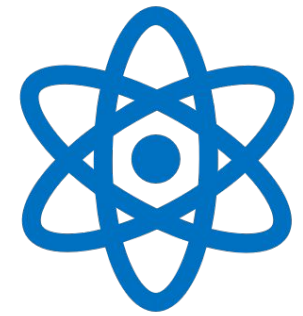
Zadań



- ❖ ... wymagających wymyślenia przepisu na obliczenia i jego implementacji z jednoznacznie zdefiniowaną poprawnością.

Środowiska programistycznego

- ❖ ..skupiającego na istocie wprowadzanych pojęć,
- ❖ umożliwiającego samodzielną naukę we własnym tempie,
- ❖ wykorzystującego gotowość dzieci do eksperymentowania i popełniania błędów,
- ❖ dającego informacje zwrotne o sukcesach i o błędach.

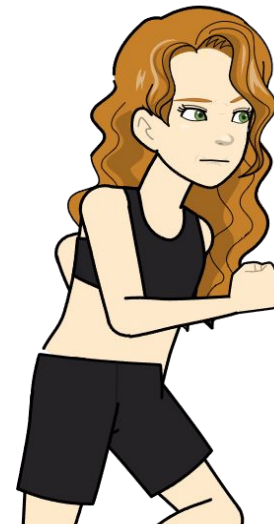


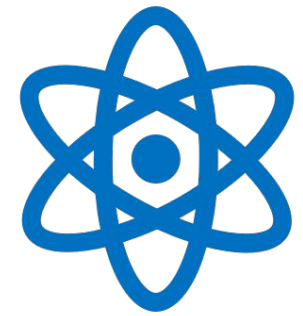
Jak uczyć programowania?

Uczyć pojęć i sposobu myślenia, nie języków

Uczyć na kartce papieru, faza projektowania

- ❖ Intencyjność działań
- ❖ Zapis wymyślnego przepisu na obliczenia
- algorytm, diagram
- ❖ Stan zmiennych

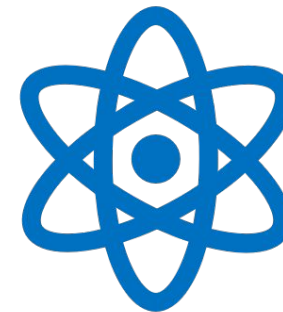




Kolejność wprowadzania pojęć

- ❖ Program
- ❖ Algorytm
- ❖ Jak zapisana jest liczba w pamięci komputera
– system binarny
- ❖ Zmienna (pudełko)
- ❖ Wejście / wyjście - wypisanie zmiennej, wypisanie napisu
- ❖ Operacje arytmetyczne
- ❖ Instrukcja warunkowa
- ❖ Pętla

Wykorzystywane znane uczniom modele matematyczne



Tak proste jak tylko możliwe i nie prostsze

Modele matematyczne:

- Operacje arytmetyczne (dodawanie i odejmowanie)
- Mnożenie (pętle)
- Dzielenie (pętle)
- Wyrażenia arytmetyczne (algebraiczne)
- Wyrażenia logiczne

Pisanie i uruchamianie programu



programista



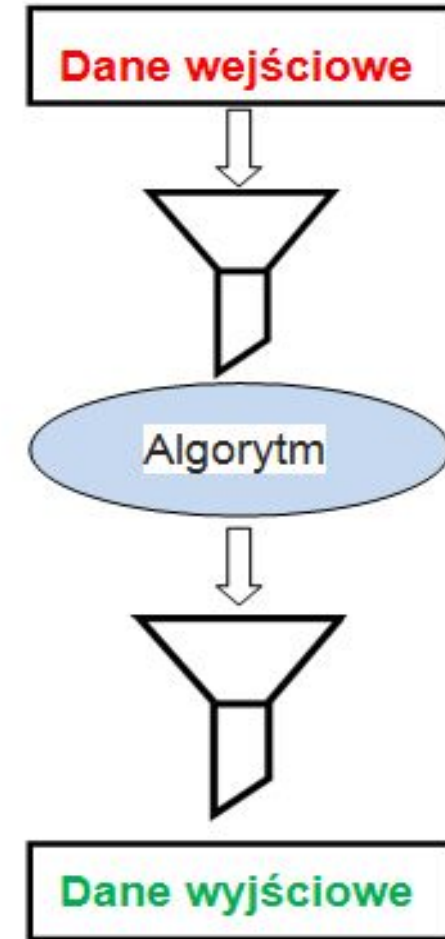
kod programu



użytkownik

Podstawowe pojęcia

- **Zadanie / problem**
- **Algorytm** – lista kroków do wykonania, zadaniem algorytmu jest na podstawie określonych danych osiągnąć oczekiwany wynik.
- **Program** – algorytm zapisany w języku zrozumiałym dla komputera



Rozkład materiału

Na każdy semestr nauki podstaw programowania
10 rozdziałów tematycznych

Rozkład materiału InstaKod

Program pilotażowy 2023/2024

poziom klasy	semestr 1	semestr 2
klasa 4 pilot	program klasy 4 podstawy programowania	technologie informacyjne
klasa 5 pilot	program klasy 4 podstawy programowania	program klasy 5 kontynuacji
klasa 6 pilot	program klasy 5 od podstaw podstawy programowania	program klasy 6 kontynuacji podstawy programowania

Konstrukcja lekcji

Do każdego rozdziału:

- I. Poradnik dla nauczyciela
- II. Podręcznik dla ucznia
- III. Przykłady z wzorcowymi rozwiązaniami
- IV. Karty pracy
- V. Zadania w podziale na 3 poziomy trudności
- VI. Quiz oraz
- VII. Sprawdziany (pierwszy z lekcji 1-4, drugi z lekcji 5-8)

I. Poradnik dla nauczyciela zawiera:

- zadań,
- quizów,
- kart pracy
- sprawdzianów

II. Podręcznik dla ucznia

- wprowadzenie do tematu
- wzorcowe rozwiązania przykładów
- lista dotychczas poznanych instrukcji wraz z ich wyjaśnieniem

Warto jest odsyłać uczniów, którzy mają problem z rozwiązaniem zadań do wzorcowych przykładów, do przepisania podanego rozwiązania i sprawdzenia jak działa.

Uczniowie powinni być na bieżąco zaznajomieni z listą instrukcji z ich wytłumaczeniem, Nauczyciel w trakcie implementacji wymyślonych rozwiązań przykładów powinien do tej listy regularnie się odnosić, pytając zarówno „która instrukcja pozwoli na...” jak i „co robi instrukcja ...”

III. Przykłady

A. Zrozumienie treści zadania

Dobłą praktyką w ćwiczniu czytania ze zrozumieniem jest poza przykładowym pokazaniem technik, narzucenie kosztu na zgłoszenie braku rozumienia nauczycielowi. Nauczyciel nie powinien nigdy dawać „gotowej” interpretacji zadania, tylko serią pytań motywować ucznia do samodzielnego przeczytania, zrozumienia i wytłumaczenia własnymi słowami treści.

Warto pokazać uczniom jak rozkładać treść zadania na krótkie fragmenty i interpretowanie ich kawałek po kawałku.

B. Wymyślenie rozwiązania wraz z przykładowymi danymi wejściowymi i oczekiwanym wynikiem

- przemyślenie przykładowego wejścia i oczekiwanego wyjścia
- wymyślanie rozwiązania moderowane przez nauczyciela poprzez zadawanie pytań nakierowujących
- wychwytywać błędne propozycje rozwiązań podawane przez uczniów, by przejść przez ścieżkę korekty wczesnych błędów - jeśli uczniowie nie wyłapią błędów od razu można nimi pokierować zadając dodatkowe pytania.

C. Zapisanie diagramu rozwiązania na tablicy

Kolejnym krokiem jest zapisanie wymyślonego rozwiązania w formie diagramu.

W przypadku prostych zadań wymyślanie rozwiązania można połączyć z zapisywaniem go fragment po fragmencie.

Diagram rozwiązania może być zapisany w dowolny, zrozumiały dla ucznia sposób z zachowaniem kolejności działań do wykonania.

W diagramie zapisujemy, na których pudełkach wykonujemy operacje, w szczególności, gdzie chcemy zapisać dane wejściowe i gdzie będą znajdować się dane wyjściowe

D. Implementacja

W trakcie pierwszych kilku lekcji można zacząć pisanie kodu od przypomnienia co robią poszczególne poznane już instrukcje.

Ich definicje uczniowie powinni znać na pamięć i być w stanie własnymi słowami wytłumaczyć co dzieje się gdy uruchomiona zostanie konkretna instrukcja.

Warto za na lekcji powtórzyć, które instrukcje są instrukcjami wejścia i wyjścia, w szczególności, która instrukcja „zawsze wypisze to samo niezależnie od podanych danych do programu”, a która wypisze wynik zależny od wczytanych danych i znajdujący się w jednym z pudełek.

E. Uruchomienie programu dla założonych danych i porównanie wyjścia z oczekiwanym

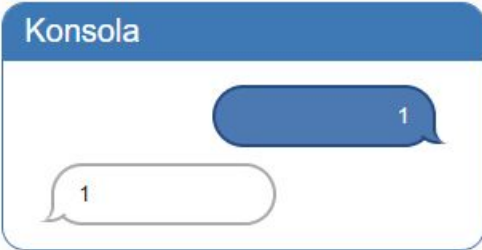
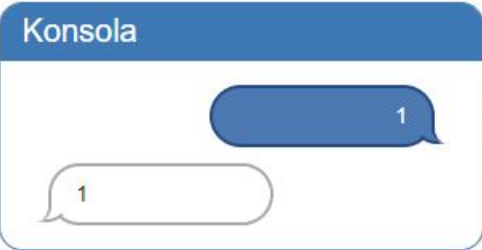
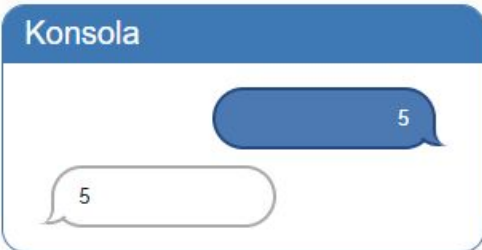
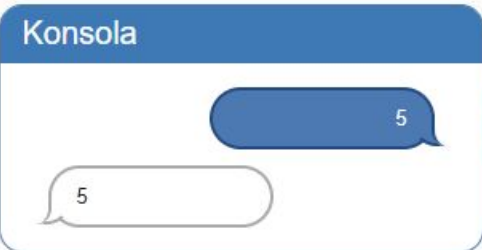
- Przed uruchomieniem programu przypominamy przykładowe dane wejściowe i oczekiwany wynik.
- Program uruchamiamy instrukcja po instrukcji, przy każdym kroku pytając uczniów czego należy się spodziewać po wykonaniu danej instrukcji i skonfrontować to z pojawiającymi się wartościami w pudełkach oraz z zapisami na konsoli.
- Po zakończeniu programu porównujemy otrzymany wynik z oczekiwanym. Jeżeli się różnią, analizujemy różnice i pytaniami nakierowującymi pomagamy uczniom znaleźć błąd.

F. Automatyczne testy i analiza błędów

Opcja automatycznego testowania i zaliczenia zadania dostępna jest po ręcznym uruchomieniu zadania. Każda wprowadzona zmiana do kodu wymaga ręcznego uruchomienia przed automatycznym testowaniem. W testach możemy otrzymać trzy rezultaty:

- Dobrze
- Brak danych
- Zła odpowiedź

Po kliknięciu „Przetestuj” otrzymujemy informację o wynikach testów. Wynik „Dobrze” jest oczywisty.

Testowanie programu					
Rezultat	Twoja odpowiedź	Prawidłowa odpowiedź	Twoje pudełka	Prawidłowe pudełka	Kroków
Dobrze			A: 0 B: 1 C: 0 D: 0	B: 1	2
Dobrze			A: 0 B: 5 C: 0 D: 0	B: 5	2

Zamknij

Wynik „Zła odpowiedź” oznacza, że rozwiązanie daje zły wynik.

Zadanie: V 1.2

Napisz program, który wypisze x.

Przykład 1

Program powinien wypisać:

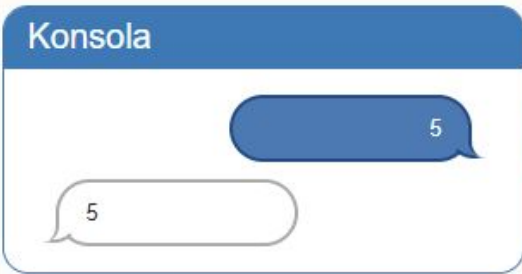
Konsola

x

Testowanie programu

Rezultat	Twoja odpowiedź	Prawidłowa odpowiedź
Zła odpowiedź	Konsola y	Konsola x

Wynik „Brak danych” oznacza, że twój program chce wczytać więcej danych niż jest w rozwiązaniu

Testowanie programu					
Rezultat	Twoja odpowiedź	Prawidłowa odpowiedź	Twoje pudełka	Prawidłowe pudełka	Kroków
Brak danych			A: 0 B: 1 C: 0 D: 0	B: 1	2
Brak danych			A: 0 B: 5 C: 0 D: 0	B: 5	2

Zamknij

G. Wykorzystanie błędów do ćwiczenia ich analizy

Warto jest wykorzystując sugestie uczniów robić w rozwiązaniach przykładów i ich implementacji typowe błędy, by pokazać uczniom ścieżkę analizy błędów i ich poprawiania. Karty pracy „Analiza poprawności rozwiązań” dodatkowo wspierają ćwiczenie analizy błędów.

Przechodząc przez całą ścieżkę rozwiązania problemu informatycznego warto poświęcić energię na zaangażowanie uczniów w wymyślanie i implementowanie rozwiązań oraz krytyczną analizę sugestii kolegów.

Nie warto by uczniowie tracili czas na przepisywanie do swoich zeszytów tworzonych wspólnie rozwiązań przykładów. Wzorcowe rozwiązania przykładów znajdują się w podręczniku dostępnym w danym rozdziale.

W razie problemów z rozwiązaniem zadań z danego rozdziału można odesłać ucznia do przejrzania wzorcowych rozwiązań przykładów w podręczniku. Zaliczenie przykładu nie będzie uwzględnione w sumarycznej liczbie zaliczonych zadań i w ocenach danego rozdziału.

Karty pracy

Ważnym elementem budującym rozumienie co dzieje się w programie są karty pracy. Rozwiązując zadania uczą się tworzyć kod. W kartach pracy uczą się „czytać” kod. Nie mając do dyspozycji komputera uczniowie nie mają możliwości pracy metodą prób i błędów. Skazani są na samodzielną analizę tego co dzieje się w podanych fragmentach kodu. Karty pracy dzielą się na kilka typów:

1. Wartości w pudełkach i zapis na konsoli
2. Wartości początkowe i końcowe
3. Analiza poprawności rozwiązania
4. Diagram rozwiązania

Atmosfera cichej, skupionej pracy w pracowni

Proces precyzyjnego, uporządkowanego myślenia stanowi podstawę nauki programowania. Atmosfera cichej, skupionej pracy i szacunku dla potrzeby skupienia się na swoich myślach wspiera ten proces. Techniki wspierające tworzenie takiej atmosfery:

- Wyciszenie uczniów, ustalenie, że przed wejściem do pracowni muszą zakończyć wszelkie dyskusje i zostawić je za drzwiami.
- Przyzwyczajanie uczniów do wchodzenia do pracowni w ciszy
- Jasne nazwanie i rozdzielenie czasu omawiania wprowadzenia do kolejnych lekcji i przykładów od czasu skupionej indywidualnej pracy nad zadaniami
- Ustalenie sposobu sygnalizowania potrzeby pomocy w czasie skupionej pracy bez odzywania się, np. przez podniesienie ręki.

Zaangażowanie najlepszych uczniów do pomocy

Po kilku pierwszych zajęciach, kiedy uczniowie już dobrze zrozumieli zasady pracy w pracowni, warto zachęcać tych, którzy z sukcesem rozwiązywali wszystkie zadania z danego rozdziału, do udzielania pomocy innym uczniom, którzy o nią poproszą.

Należy jednak podkreślić, że celem jest **pomaganie**, a nie wykonywanie zadań za innych. Chodzi o to, aby pomagać kolegom i koleżankom w zrozumieniu i opanowaniu materiału, wspierając ich w procesie nauki.

Nakierowujące pytania

Metodą usamodzielniania uczniów w rozwiązywaniu problemów informatycznych jest metoda przybliżających pytań. Polega ona na zadawaniu uczniom pytania po pytaniu prowadzących do „wymyślenia” przez nich rozwiązania.

Doskonalenie czytania ze zrozumieniem

Uczniowie często pytają o wyjaśnienie zadań, unikając własnego myślenia.

Skuteczną metodą doskonalenia czytania ze zrozumieniem jest prośba o przeczytanie polecenia na głos, a następnie o powtórzenie go własnymi słowami i zadawanie pytań, które wymagają refleksji nad treścią.

Po pewnym czasie, większość uczniów nabiera umiejętności czytania ze zrozumieniem i przestaje polegać na pytaniach innych osób. Zamiast tego, sami stają się bardziej samodzielni i sami powielają w swoich głowach scenariusz czytania fragment po fragmencie.

Ścieżka dla uczniów zdolnych

- Instakod pozwala na prowadzenie uczniów zdolnych szybszym tempem, ponieważ uczniowie, którzy są w stanie pracować szybciej, powinni być puszczeni dalej. Korzystając samodzielnie z podręcznika mogą rozwiązywać zadania z kolejnych rozdziałów. Jeśli napotkają na zadania lub pojęcia, których nie są w stanie samodzielnie zrozumieć mogą skorzystać z czasu zajęć dodatkowych (np. kółka informatycznego, jeśli takie jest).
- Nauczyciel nie powinien rezerwować czasu indywidualnej pracy z takim uczniem w trakcie lekcji z całą klasą, chyba, że pozostali uczniowie realizujący bieżące zadania nie mają pytań.
- Po zaliczeniu wszystkich rozdziałów z danego poziomu klasy, nauczyciel powinien dopisać ucznia do kolejnego poziomu (dając mu nowy login i hasło), do klasy 8 włącznie.

Dziękujemy

