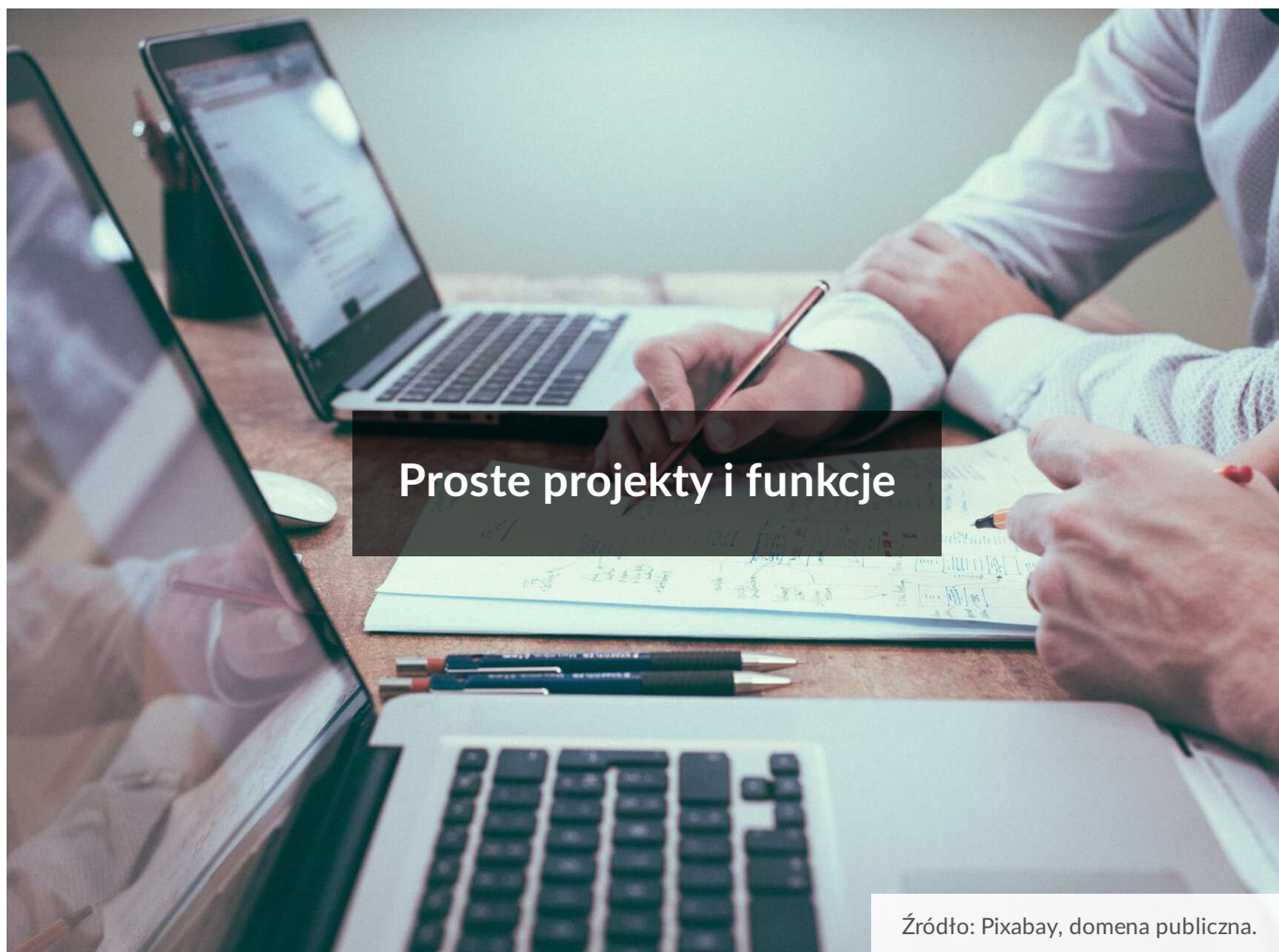




Proste projekty i funkcje

- Wprowadzenie
- Przeczytaj
- Prezentacja multimedialna
- Sprawdź się
- Dla nauczyciela



Źródło: Pixabay, domena publiczna.

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

Funkcje pozwalają rozbić program na bloki logiczne odpowiedzialne za wykonywanie określonych zadań. Zastosowanie funkcji sprawia, że kod programu staje się czytelniejszy, łatwiej jest go modyfikować oraz odnajdować i usuwać pojawiające się w nim błędy. W tym e-materiale zapoznamy się z wykorzystaniem funkcji w prostym projekcie.

Implementację projektów w wybranych językach programowania znajdziesz w e-materiałach:

- [Proste projekty i funkcje w języku C++](#),
- [Proste projekty i funkcje w języku Java](#),
- [Proste projekty i funkcje w języku Python](#).

Więcej zadań? Przejdź do e-materiału [Proste projekty i funkcje – zadania maturalne](#).

Twoje cele

- Wyjaśnisz, czym jest funkcja oraz co odróżnia ją od procedury.
- Rozróżnisz algorytmy stabilne od niestabilnych.
- Obliczysz miejsca zerowe funkcji kwadratowej, korzystając z algorytmu stabilnego.

Przeczytaj

Funkcje – omówienie

Funkcja jest wydzielonym fragmentem kodu programu, odpowiedzialnym za wykonywanie określonego zadania – na przykład obliczenie wyniku działań matematycznych. Raz zdefiniowana funkcja może być używana wielokrotnie: wystarczy ją wywołać w odpowiednim miejscu programu, bez konieczności kopiowania tych samych poleceń.

Funkcje w programowaniu przypominają nieco funkcje matematyczne, które przyporządkowują argumentom określone wartości. W programowaniu jest podobnie: funkcja przyjmuje **argumenty**, przetwarza je i zwraca pewną wartość. Należy jednak podkreślić, że funkcja zdefiniowana w programie nie musi przyjmować żadnych argumentów – i tym różni się od funkcji matematycznej.

W niektórych językach programowania można zetknąć się również z pojęciem **procedura**. Jest to konstrukcja podobna do funkcji. Różnica polega na tym, że procedura pobiera dane, które następnie przetwarza, ale nie zwraca żadnej wartości. W dalszej części tekstu będziemy posługiwać się wyłącznie terminem funkcja.

Umiejętność definiowania funkcji i posługiwania się nimi jest bardzo istotna dla każdego programisty. Dzielenie większych i powtarzalnych zadań na funkcje nie tylko skraca kod, ale także ułatwia jego analizowanie.

Błędy obliczeń na przykładzie wyznaczania miejsc zerowych funkcji kwadratowej

Zapewne niejednokrotnie zdarzyło ci się rozwiązywać równanie kwadratowe, czyli znajdować miejsca zerowe funkcji opisanej wzorem:

$$f(x) = ax^2 + bx + c$$

Współczynniki a , b i c są stałymi, przy czym $a \neq 0$. Gdyby współczynnik $a = 0$, mielibyśmy do czynienia z funkcją liniową.

Równanie kwadratowe ma postać:

$$ax^2 + bx + c = 0$$

Znasz wzory służące do wyznaczania liczby rozwiązań takiego równania oraz obliczania wartości pierwiastków:

$$\Delta = b^2 - 4ac$$

- Dla $\Delta > 0$ istnieją dwa rozwiązania:

$$x_1 = \frac{(-b - \sqrt{\Delta})}{2a}$$

$$x_2 = \frac{(-b + \sqrt{\Delta})}{2a}$$

- Dla $\Delta = 0$ istnieje jedno rozwiązanie:

$$x = \frac{-b}{2a}$$

- Dla $\Delta < 0$ równanie nie ma rozwiązania.

Przedstawione wzory moglibyśmy wykorzystać, pisząc program obliczający pierwiastki równania kwadratowego. Należy jednak pamiętać o ograniczeniach związanych ze sposobem zaokrąglania liczb podczas zapisywania wartości pierwiastka. Jeśli spróbujemy przechowywać go w zmiennej typu całkowitego, utracimy całą część ułamkową.

Błąd taki łatwo jest naprawić – wystarczy użyć typu zmiennoprzecinkowego. Niestety, także i w tym wypadku nie da się wykluczyć, że wynik zostanie zaokrąglony. Błąd tego typu jest nie do usunięcia.

Gdy rozwiązujemy równanie kwadratowe, może również dojść do pojawienia się błędu podczas wykonywania następującego działania:

$$x_2 = \frac{(-b + \sqrt{\Delta})}{2a}$$

Jeżeli licznik ułamka będzie miał wartość bardzo bliską zera, mogą pojawić się błędy zaokrąglania. Wyeliminujemy je, stosując inny [algorytm](#) obliczeń.

Ważne!

Algorytmy niestabilne są algorytmami, których zastosowanie dla pewnych danych wejściowych skutkuje uzyskaniem niedokładnych wyników. Zastosowanie algorytmów stabilnych eliminuje natomiast wszystkie możliwe do uniknięcia błędy związane z zaokrągleniami.

Stabilny algorytm rozwiązywania równania kwadratowego

W celu wyeliminowania błędów zaokrągleń należy użyć wzoru służącego do obliczania wyróżnika równania kwadratowego (delty) oraz jednego z wzorów Viète'a.

Wzorów tych możemy używać, gdy chcemy obliczyć miejsca zerowe dla funkcji kwadratowej postaci:

$$f(x) = ax^2 + bx + c$$

Gdzie, aby funkcja była funkcją kwadratową, $a \neq 0$.

Wzory te prezentują następujące zależności:

$$x_1 + x_2 = \frac{-b + \sqrt{\Delta}}{2a} + \frac{-b - \sqrt{\Delta}}{2a} = -\frac{2b}{2a} = -\frac{b}{a}$$

Ostatecznie pierwszy przyjmuje postać:

$$x_1 + x_2 = -\frac{b}{a}$$

oraz analogicznie:

$$x_1 \cdot x_2 = \frac{-b + \sqrt{\Delta}}{2a} \cdot \frac{-b - \sqrt{\Delta}}{2a} = \frac{(-b)^2 - (\sqrt{\Delta})^2}{(2a)^2} = \frac{b^2 - \Delta}{4a^2} = \frac{b^2 - (b^2 - 4ac)}{4a^2} = \frac{4ac}{4a^2} = \frac{c}{a}$$

Natomiast drugi przyjmuje postać:

$$x_1 \cdot x_2 = \frac{c}{a}$$

Skorzystamy z drugiego wzoru. Po przekształceniu zależność ta pozwala obliczyć wartość x_2 :

$$x_2 = \frac{c}{(a \cdot x_1)}$$

Jeśli obliczymy wartość x_1 , która nie jest obarczona wspomnianym już błędem zaokrąglenia, możemy również wyznaczyć x_2 , korzystając ze wzoru Viète'a. Pozwoli to uniknąć błędu zaokrąglenia także w przypadku drugiego miejsca zerowego.


```
1 a = wprowadź liczbę całkowitą
2 b = wprowadź liczbę całkowitą
3 c = wprowadź liczbę całkowitą
4
5 delta = b * b - 4 * a * c
6 jeżeli delta < 0 wykonaj:
7     wypisz "brak rozwiązań"
8 jeżeli delta == 0 wykonaj:
9     x = -b / (2 * a)
10    wypisz x
11 jeżeli delta > 0 wykonaj:
12    pomocnicza = c/a
13    jeżeli b > 0 wykonaj:
14        x1 = (-b - pierwiastek(delta)) / (2 * a)
15        x2 = pomocnicza/x1
16    jeżeli b < 0 wykonaj:
17        x2 = (-b + pierwiastek(delta)) / (2 * a)
18        x1 = pomocnicza/x2
19    wypisz x1, x2
```

Słownik

algorytm

skończona lista instrukcji prowadzących do zrealizowania pewnego zadania; często przedstawiany jest w postaci schematu blokowego lub listy kroków

argument

wartość, która jest przekazywana funkcji

funkcja

wydzielony fragment kodu programu, odpowiedzialny za wykonanie pewnych operacji;

funkcja może być wykorzystywana wielokrotnie oraz używać własnych zmiennych lokalnych;

funkcje są nazywane również podprogramami lub – gdy nie zwracają żadnej wartości – procedurami

Prezentacja multimedialna

Polecenie 1

Napisz program sprawdzający liczbę miejsc zerowych wielomianu drugiego stopnia.

Specyfikacja problemu:

Dane:

- a, b, c – liczby rzeczywiste; współczynniki wielomianu drugiego stopnia

Wynik:

Program wyświetla liczbę miejsc zerowych wielomianu drugiego stopnia.

1

1

Polecenie 2

Porównaj swój kod z pseudokodem przedstawionym w prezentacji.



Na ilustracji znajduje się wykres funkcji, którego miejsce zerowe wynosi 24. Odczytanie miejsca zerowego z funkcji nie zawsze jest tak proste.

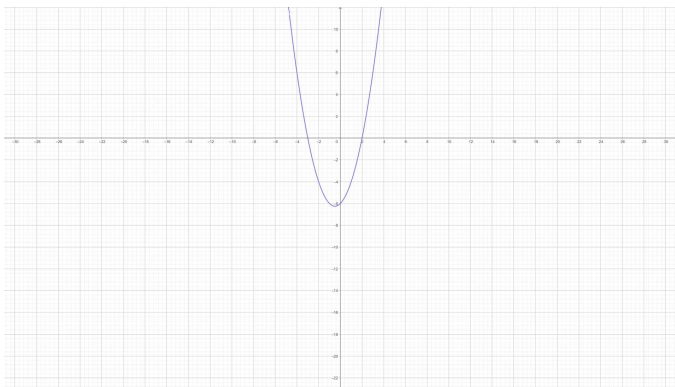
Źródło: Contentplus.pl sp. z o.o., CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PgB3dsjkg>

Funkcje są wyodrębnionymi fragmentami kodu, które mogą być wykorzystywane w programie wielokrotnie. Po zapisaniu pewnego uniwersalnego fragmentu kodu wywołuje się go, ilekroć zajdzie taka potrzeba. Unika się dzięki temu powielania tych samych instrukcji w wielu miejscach.

2



Na ilustracji znajduje się wykres funkcji. Spróbuj odczytać miejsca zerowe. Wzór funkcji to $f(x) = x^2 + x - 6$.

Źródło: Contentplus.pl sp. z o.o., CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PgB3dsjkg>

Odpowiednio zdefiniowaną funkcję możemy wykorzystać np. do rekurencyjnego obliczania silni wybranej liczby. Pojedynczy fragment kodu zostanie wywołany wiele razy (liczba powtórzeń zależy od tego, jakiej liczby silnię chcemy wyznaczyć).

3

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PgB3dsjkg>

Gdyby okazało się, że funkcja obliczająca silnię zawiera błędy, wystarczyłoby poprawić dwie linie (tyle bowiem trzeba do zapisania odpowiedniego kodu) zamiast modyfikować (w skrajnym przypadku) nawet n linii (gdzie n oznacza liczbę, której silnię wyznaczamy).

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PgB3dsjkg>

Także w przypadku, gdy obliczamy rozwiązanie równania kwadratowego, możemy skrócić i uprościć kod programu, pisząc odpowiednią funkcję. Powinniśmy to zrobić zwłaszcza wtedy, gdy wyznaczamy miejsca zerowe wielokrotnie, dla różnych współczynników wielomianu.

5

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PgB3dsjkg>

Jeśli poprawnie zdefiniujemy funkcję liczącą miejsca zerowe i będziemy ją wykorzystywali za każdym razem podczas obliczania miejsc zerowych, to wszystkie obliczenia będą obarczone jedynie błędami takiego samego rodzaju. Oznacza to, że w przypadku algorytmów niestabilnych nasze wyniki będą obarczone błędami numerycznymi. Jednakże w przypadku poprawnego zaimplementowania algorytmu stabilnego unikniemy wszystkich błędów, których da się potencjalnie uniknąć.

6

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PgB3dsjkg>

Jeśli zadeklarujemy w pseudokodzie funkcję:

```
1 obliczanieMiejscZerowych(a  
  , b, c)
```

okaże się, że jest ona podatna na błędy. Należy pamiętać, że miejsca zerowe da się za pomocą takiej funkcji obliczyć tylko wtedy, gdy współczynnik a jest różny od zera. W innym przypadku funkcja zwróci błąd, ponieważ

w pewnym momencie dojdzie do dzielenia przez zero.

7

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PgB3dsjkg>

Założmy, że piszemy taką funkcję dla kogoś, kto nie jest świadomy możliwości pojawienia się błędu; co więcej, funkcja ma być tylko jedną z kilkuset funkcji przygotowywanych dla wspomnianej osoby.

8

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PgB3dsjkg>

Do tak rozbudowanego zestawu funkcji (który już w zasadzie możemy nazwać biblioteką) musimy dostarczyć dokumentację. Jest to opis poszczególnych funkcji (do czego każda z nich służy, jakie argumenty przyjmuje, co zwraca oraz – ewentualnie – co konkretna funkcja robi, a także jaką rolę pełni w kontekście zawartość całej biblioteki, wraz z informacją, gdzie jest wykorzystywana).

9

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PgB3dsjkg>

Oto przykładowa funkcja:

```
1  
2 funkcja  
   obliczanieMiejscZerowych(a  
   , b, c)
```

```

3      delta = b * b - 4 * a
      * c
4      jeżeli delta < 0
      wykonaj:
5          wypisz „brak
      rozwiązań”
6      jeżeli delta == 0
      wykonaj:
7          x = - b / (2 * a)
8          wypisz x
9      jeżeli delta > 0
      wykonaj:
10         pomocnicza = c / a
11         jeżeli b > 0
      wykonaj:
12             x1 = (-b -
      pierwiastek(delta)) / (2 *
      a)
13             x2 =
      pomocnicza/x1
14             jeżeli b < 0
15                 x2 = (-b +
      pierwiastek(delta)) / (2 *
      a)
16             x1 =
      pomocnicza / x2
17             wypisz x1, x2

```

zerowe, jeśli wyróżnik równania kwadratowego (delta) jest równy 0, lub dwa różne miejsca zerowe, jeśli wyróżnik równania kwadratowego jest większy od zera. Współczynniki muszą być liczbami rzeczywistymi.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PgB3dsjkg>

11

W ten sposób informujemy, jakie argumenty przyjmuje oraz co oblicza i zwraca funkcja. Zawartość dokumentacji może zmieniać się w zależności od użytego języka programowania. W niektórych przypadkach dopuszczalne jest zwracanie przez funkcję raz jednej, a raz dwóch wartości; istnieją też języki, w których jest to nieakceptowalne.

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Polecenie 3

Zapisz notatkę, w której podsumujesz najważniejsze informacje przedstawione w prezentacji multimedialnej.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Uzupełnij tekst brakującymi elementami.

to fragment kodu, który napisany raz, może zostać wykorzystany wielokrotnie, bez konieczności kopiowania go. W przypadku programów, w których jedna czynność wykonywana jest wielokrotnie, pomagają one uniknąć zbędnych powtórzeń.

Algorytmy to takie algorytmy, których zastosowanie dla pewnych danych wejściowych skutkuje otrzymaniem niedokładnych wyników. Algorytmy pozwalają wyeliminować wszystkie możliwe do uniknięcia błędy związane z zaokrągleniami.

to fragment kodu, który nie zwraca żadnej wartości.

niestabilne

procedura

stabilne

procedura

funkcja

algorytm

stabilne

Ćwiczenie 2



Wskaż, które ze stwierdzeń jest prawdziwe.

☐ Funkcja zawsze zwraca pewną wartość.

☐ Funkcja może nie zwracać żadnej wartości.

☐ Funkcja zawsze musi pobierać jakieś argumenty.

Ćwiczenie 3



Wskaż stwierdzenia nieprawdziwe.

☐

Stosując stabilny algorytm obliczania pierwiastka kwadratowego, otrzymujemy wynik najbardziej zbliżony do wyniku prawdziwego.

☐

Stabilny algorytm obliczania pierwiastka kwadratowego jest niepodatny na błędy zaokrągleń.

☐

Stosując stabilny algorytm obliczania pierwiastka kwadratowego, otrzymujemy zawsze dokładny wynik.

Ćwiczenie 4



Uzupełnij tekst.

Gdy delta jest:

•

to równanie ma jedno rozwiązanie;

•

to równanie ma dwa rozwiązania;

•

to równanie nie ma rozwiązań w przestrzeni liczb rzeczywistych.

mniejsza od zera

równa zero

większa od zera

większa od zera

mniejsza od zera

równa zero

Ćwiczenie 5



Wskaż bezpośrednie korzyści wynikające ze stosowania funkcji.

☐

łatwiejsze odnajdowanie ewentualnych błędów w kodzie

☐

brak konieczności używania zewnętrznych baz wiedzy

☐

uczynienie kodu nadmiarowym

☐

skrócenie kodu i sprawienie, że jest on bardziej czytelny

Ćwiczenie 6



Wskaż, które stwierdzenia są prawdziwe.

☐

Algorytm może być niejasny.

☐

Przykładem algorytmu jest instrukcja składania mebli.

☐

Algorytm musi mieć koniec.

☐

Algorytm musi składać się ze skończonej liczby kroków.

Ćwiczenie 7



Napisz program podnoszący liczbę do kwadratu.

Specyfikacja problemu:

Dane:

- *liczba* – liczba rzeczywista

Wynik:

Program oblicza kwadrat podanej liczby i wypisuje komunikat na temat wartości potęgi.

Przykładowe wyjście:

```
1 Liczba 2 podniesiona do kwadratu wynosi 4
```

Przetestuj działanie swojego programu dla liczby 10.

```
1
```

```
1
```



Ćwiczenie 8



Napisz program liczący objętość wody, jaką pomieści akwarium w kształcie prostopadłościanu o wymiarach podanych w cm. Swój program przetestuj dla akwarium o bokach długości 45 cm, 70 cm, 90 cm.

Specyfikacja problemu:

Dane:

- *bok1, bok2, bok3* – liczby naturalne

Wynik:

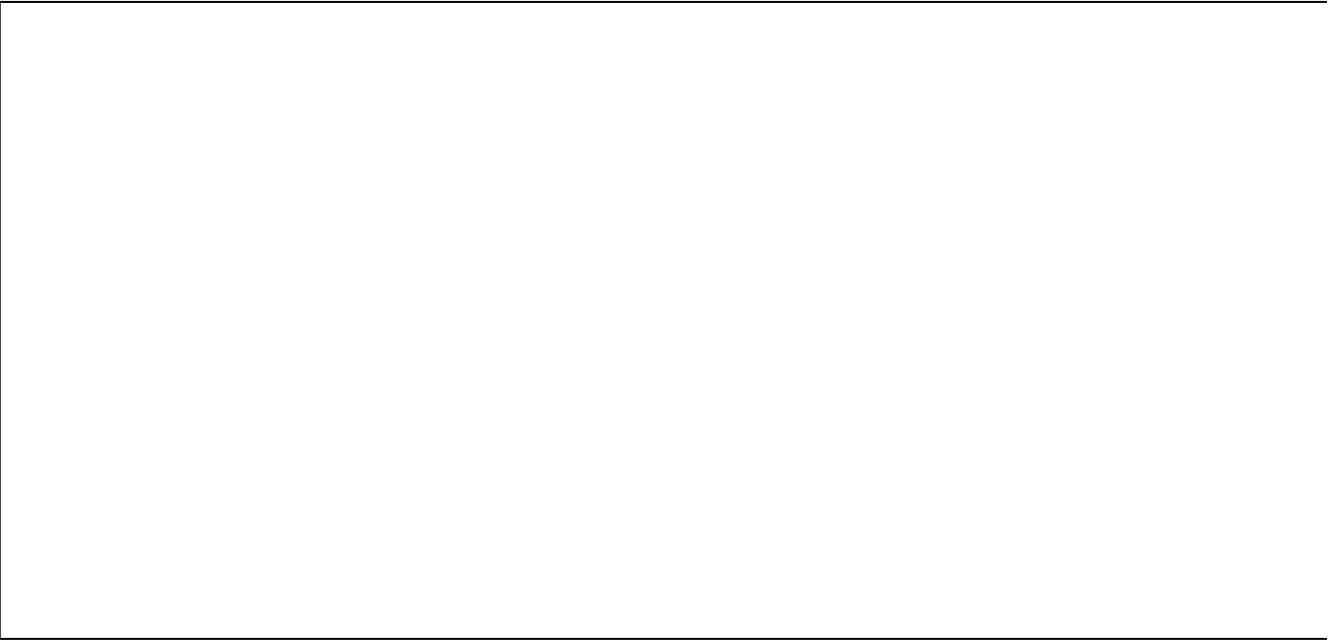
Program wypisuje komunikat na temat tego, ile wody w litrach pomieści akwarium.

Przykładowe wyjście:

```
1 Twoje akwarium pomiesci 100 l wody
```

```
1
```

```
1
```



Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Proste projekty i funkcje

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy i rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres podstawowy. Uczeń:

1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

2) stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów;

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Wyjaśnisz, czym jest funkcja oraz co odróżnia ją od procedury.
- Rozróżnisz algorytmy stabilne od niestabilnych.
- Obliczysz miejsca zerowe funkcji kwadratowej, korzystając z algorytmu stabilnego.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji);
- oprogramowanie dla języka Python 3 (lub nowszej wersji), w tym PyCharm lub IDLE.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Proste projekty i funkcje”. Uczniowie zapoznają się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Wyświetlenie przez nauczyciela tematu i celów lekcji. Określenie wiążących dla uczniów kryteriów sukcesu.

Faza realizacyjna:

1. **Praca z tekstem.** Uczniowie analizują treści z sekcji „Przeczytaj” wyświetlone na tablicy.
2. **Praca z multimedialnym.** Nauczyciel wyświetla zawartość sekcji „Prezentacja multimedialna”. Uczniowie zapoznają się z Problemem 1 i w parach wypracowują rozwiązanie. Następnie porównują je z pseudokodem opisanym w prezentacji.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują indywidualnie ćwiczenia nr 1–3 z sekcji „Sprawdź się”. Po wykonaniu każdego z nich następuje omówienie rozwiązania przez nauczyciela.
4. W kolejnym etapie uczniowie dobierają się w pary i wykonują ćwiczenia nr 4–7 z sekcji „Sprawdź się”. Następnie konsultują swoje rozwiązania z inną parą uczniów i ustalają jedną wersję odpowiedzi.

Faza podsumowująca:

1. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązywania ćwiczeń.

Praca domowa:

1. Uczniowie wykonują ćwiczenie 8 z sekcji „Sprawdź się”.

Wskazówki metodyczne:

- Uczniowie mogą wykorzystać treści w sekcjach: „Przeczytaj”, „Prezentacja multimedialna”, „Sprawdź się” jako materiał do lekcji powtórkowej.