Przerywanie pętli w języku C++

- Wprowadzenie
- Film samouczek
- Przeczytaj
- Sprawdź się
- Dla nauczyciela

Źródło: Mitchell Luo, domena publiczna.

W e-materiale Przerywanie pętli poznaliśmy optymalizujące algorytm instrukcje `break` oraz `continue` . Są to dodatkowe instrukcje sterujące zachowaniem pętli, które pozwalają wymusić zakończenie iteracji i ominąć pewne instrukcje lub całkowicie zatrzymać działanie pętli.

W tym e-materiale zaimplementujemy omawiane instrukcje w języku C++.

Ciekawi cię, jak wyglądają implementacje w innych językach programowania? Możesz się z nimi zapoznać w dwóch pozostałych e-materiałach z tej serii:

- Przerywanie pętli w języku Java,
- Przerywanie pętli w języku Python.

Więcej zadań? Sięgnij do Przerywanie pętli – zadania maturalne.

> **Twoje cele**
>
> - Wyjaśnisz, na czym polegają instrukcje `break` i `continue`.
> - Przeanalizujesz sytuacje, podczas których warto zastosować te instrukcje.
> - Wykorzystasz instrukcje `break` i `continue` do optymalizacji algorytmów.

Film samouczek

Polecenie 1

Przeanalizuj film, a następnie przedstaw przebieg działania pętli cykl po cyklu.

Film dostępny pod adresem /preview/resource/R18hZfeacOimH

Nagranie filmowe przedstawiające przerywanie pętli.

Problem 1

Specyfikacja problemu:

Wypisz i zsumuj kolejne liczby nieparzyste od 1 do 100. Operację przerwij, gdy suma przekroczy liczbę 50.

Dane:

Liczba jest parzysta, jeżeli dzieli się przez 2 bez reszty.

Wynik:

Na konsoli wyświetlą się liczby nieparzyste, a pętla zostanie przerwana gdy suma będzie większa niż 50.

Lista kroków:

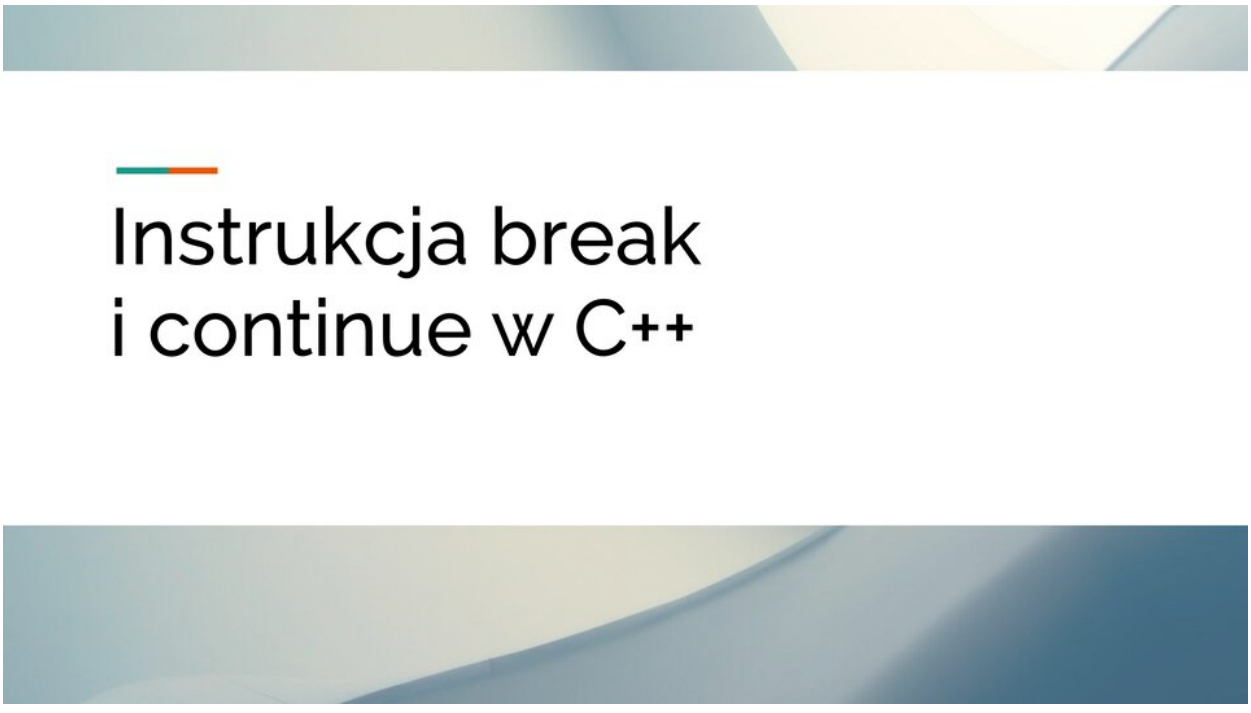
```
```

```cpp
#include <iostream>

using namespace std;

int main() {
    // tutaj dopisz kod


}
```

```
```

Polecenie 2

Porównaj swoje rozwiązanie z filmem poniżej.



Film dostępny pod adresem /preview/resource/RcCqIC6B4ACGd

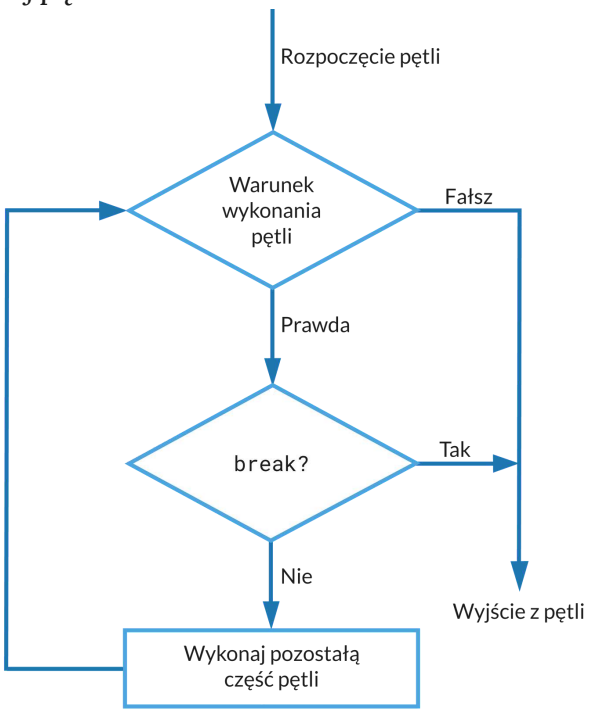
Nagranie filmowe przedstawiające instrukcję break i continue w c++.

Przeczytaj

Instrukcja `break`

Instrukcja `break` pozwala na wcześniejsze zakończenie pętli bez przerywania działania całego programu. Komendę `break` można stosować w pętlach `for`, `while` oraz `do-while`.

Należy pamiętać, że instrukcja `break` znajdująca się w pętli zagnieżdżonej zakończy działanie tylko wewnętrznej pętli.



Źródło: GroMar Sp. z o.o., licencja: CC BY-SA 3.0.

Jeśli chcielibyśmy znaleźć najmniejszą liczbę całkowitą z przedziału od 0 do 100, której kwadrat jest większy od 100, nasza pętla musiałaby przejść aż 101 cykli. Proces ten możemy zoptymalizować, używając instrukcji `break`, ponieważ mamy pewność, że każda kolejna liczba podniesiona do kwadratu daje wynik większy niż poprzedni. W omawianym przypadku pętla zostanie wykonana 12, nie zaś 101 razy.

```
#include <iostream>

int main () {
    for (int i = 0; i <= 100; i++) {
        int x = i * i;
        std::cout << "Kwadrat liczby " << i << " to " << x << std

        if (x > 100) {
            std::cout << "Szukana liczba to " << i << std::endl;
```

```
            break;
        }
    }
}
```

Wynik działania programu wygląda następująco:

```
Kwadrat liczby 0 to 0
Kwadrat liczby 1 to 1
Kwadrat liczby 2 to 4
Kwadrat liczby 3 to 9
Kwadrat liczby 4 to 16
Kwadrat liczby 5 to 25
Kwadrat liczby 6 to 36
Kwadrat liczby 7 to 49
Kwadrat liczby 8 to 64
Kwadrat liczby 9 to 81
Kwadrat liczby 10 to 100
Kwadrat liczby 11 to 121
Szukana liczba to 11
```

Instrukcja `continue`

Działanie instrukcji `continue` jest podobne do działania instrukcji `break` – instrukcja `continue` służy jednak do zakończenia aktualnego cyklu pętli. Możemy ją wykorzystać, gdy jakiś element nie spełnia określonych warunków i dlatego chcemy pominąć pozostałe operacje w bieżącym obiegu pętli.

Prosty przykład działania omawianej instrukcji zawiera polecenie wypisania wszystkich liczb parzystych od 1 do 10:

```
#include <iostream>

int main () {
    for (int i = 1; i <= 10; i++) {
        if (i % 2 != 0) {
            continue;
        }

        std::cout << i << std::endl;
```

```
    }
}
```

Wykonanie przedstawionego wyżej kodu spowoduje, że zostaną wypisane wszystkie liczby parzyste z zadanego przedziału:

```
2
4
6
8
10
```

Kiedy przerywać działanie pętli?

Zadanie z wypisywaniem liczb parzystych można rozwiązać również bez użycia instrukcji `continue`. Wystarczy zmienić warunek wewnątrz pętli, aby uzyskać ten sam efekt:

```
#include <iostream>

int main () {
    for (int i = 1; i <= 10; i++) {
        if (i % 2 == 0) {
            std::cout << i << std::endl;
        }
    }
}
```

Możemy również przeprojektować całą pętlę, aby pozbyć się instrukcji warunkowej:

```
#include <iostream>

int main () {
    for (int i = 2; i <= 10; i += 2) {
        std::cout << i << std::endl;
    }
}
```

Kiedy korzystać z instrukcji `break` i `continue`? Wtedy, gdy chcemy ograniczyć liczbę iteracji. Musimy jednak pamiętać o takim wykorzystaniu możliwości przerywania pętli, które nie zaburzy czytelności programu – ma to szczególne znaczenie w skomplikowanych algorytmach, w których dba się o przejrzystość kodu. Operacje tego typu wykonuje się więc nawet kosztem nieznacznych strat w prędkości działania programu.

Słownik

> **zagnieżdżenie pętli**
>
> wywołanie jednej pętli wewnątrz drugiej
> **optymalizacja**
>
> (od łac. *optimus* – najlepszy) poprawa wydajności programu komputerowego lub algorytmu uzyskana dzięki zmniejszeniu liczby operacji niezbędnych do otrzymania wyniku

Sprawdź się

Pokaż ćwiczenia: 

Ćwiczenie 1

Używając instrukcji `break` lub `continue`, napisz program, który wypisze liczby podzielne przez 3 należące do przedziału 0–100 (włącznie).

Twoje zadania

1. Wypisz wszystkie liczby podzielne przez 3 należące do przedziału od 0 do 100 włącznie.

Ćwiczenie 2

Napisz program, dzięki któremu odnajdziesz najmniejszą wspólną wielokrotność (NWW) liczb 49 i 63.

Twoje zadania

1. Program oblicza NWW liczb 49 i 63.

Ćwiczenie 3

Sprawdź, czy liczba n jest liczbą pierwszą. Wykorzystaj instrukcje `break` lub `continue` tak, aby wykonać jak najmniej operacji. Jeśli liczba nie jest pierwsza, wyświetl odpowiedni komunikat.

Twoje zadania

1. Sprawdź czy 1537 jest liczbą pierwszą.

Dla nauczyciela

Autor: Adam Żółtowski
Przedmiot: Informatyka

Temat: Przerywanie pętli w języku C++

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum

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 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, na czym polegają instrukcje `break` i `continue`.
- Przeanalizujesz sytuacje, podczas których warto zastosować te instrukcje.
- Wykorzystasz instrukcje `break` i `continue` do optymalizacji algorytmów.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimedium 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.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Przerywanie pętli w języku C++". Nauczyciel prosi uczniów o zapoznanie się z multimedium w sekcji „Przeczytaj".

Faza wstępna:

1. Nauczyciel inicjuje rozmowę wprowadzającą w temat lekcji. Przedstawia cele zajęć oraz kryteria sukcesu.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel ocenia, na podstawie informacji na platformie, stan przygotowania uczniów do zajęć. Jeżeli jest ono niewystarczające prosi o ciche zapoznanie się z treścią w sekcji „Film samouczek".
2. **Praca z multimedium.** Nauczyciel wyświetla zawartość sekcji „Prezentacja multimedialna". Uczniowie wspólnie analizują prezentację, a następnie przedstawiają przebieg działania pętli cykl po cyklu.

3. **Ćwiczenie umiejętności.** Uczniowie wykonują indywidualnie ćwiczenie nr 1: „Używając instrukcji `break` lub `continue`, napisz program, który wypisze liczby podzielne przez 3 należące do przedziału 0–100 (włącznie)", a następnie porównują swoje odpowiedzi z kolegą lub koleżanką.
4. Liga zadaniowa – uczniowie pracując w parach, wykonują ćwiczenie nr 2: „Napisz program, dzięki któremu odnajdziesz najmniejszą wspólną wielokrotność (NWW) liczb 49 i 63." z sekcji „Sprawdź się", a następnie dzielą się swoimi wynikami przez porównywanie napisanego kodu z inną grupą, która również zakończyła zadanie.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat i cele lekcji zawarte w sekcji „Wprowadzenie". W kontekście ich realizacji następuje omówienie ewentualnych problemów z rozwiązaniem ćwiczeń z sekcji „Sprawdź się".
2. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności.

Praca domowa:

1. Uczniowie wykonują ćwiczenie nr 3 z sekcji „Sprawdź się". Ich zadaniem jest sprawdzenie, czy liczba `n` jest liczbą pierwszą. Wykorzystują instrukcję `break` lub `continue` tak, aby wykonać jak najmniej operacji. Jeśli liczba nie jest pierwsza, wyświetlają odpowiedni komunikat.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.

Wskazówki metodyczne:

- Nauczyciel może wykorzystać multimedium w sekcji „Przeczytaj" do pracy przed lekcją. Uczniowie zapoznają się z jego treścią i przygotowują do pracy na zajęciach w ten sposób, żeby móc samodzielnie rozwiązać zadania dołączone do e-materiału „Przerywanie pętli w języku C++".