

Wybór wielokrotny w języku C++

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Schemat interaktywny](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)

The image shows two light switches on a wall. The switch on the left is a single-pole switch, and the switch on the right is a double-pole switch. A dark semi-transparent rectangle is overlaid on the switches, containing the title text. The background consists of vertical wooden slats.

Wybór wielokrotny w języku C++

Źródło: Karim Manjra, domena publiczna.

Instrukcja warunkowa sprawdza się bardzo dobrze, gdy trzeba wybrać do zrealizowania jeden z dwóch scenariuszy opisanych w kodzie programu. Sprawy komplikują się wraz ze zwiększaniem liczby scenariuszy. Lepszym rozwiązaniem okazuje się wówczas instrukcja wielokrotnego wyboru – `switch`.

W tym e-materiale poznasz instrukcję wielokrotnego wyboru w języku C++.

Ciekawi cię, jak wygląda omawiana kwestia w innych językach programowania? Możesz się z nimi zapoznać w dwóch pozostałych lekcjach z tej serii:

- [Wybór wielokrotny w języku Java](#),
- [Wybór wielokrotny w języku Python](#).

Twoje cele

- Zastosujesz wyrażenia trójargumentowe.
- Poznasz składnię instrukcji wielokrotnego wyboru.
- Przekonasz się, że instrukcja `switch` w pewnych przypadkach okazuje się bardziej efektywna niż instrukcja warunkowa.
- Wykorzystasz instrukcję `switch` do rozwiązania prostych zadań.

Przeczytaj

Wybór wielokrotny w języku C++

Trójargumentowe wyrażenie wyboru

Używane jest w sytuacjach, gdy wystarczy zwrócić prostą informację zależną od jednego warunku. W takim przypadku rozbudowywanie kodu jest niecelowe.

```
1 warunek_logiczny ? wartość_gdy_prawda : wartość_gdy_fałsz;  
2 wartość_gdy_prawda if warunek_logiczny else wartość_gdy_fałsz;
```

Instrukcja taka odpowiada poleceniom:

```
1 if warunek_logiczny:  
2     wartość_gdy_prawda  
3 else:  
4     wartość_gdy_fałsz
```

Porównajmy kody dwóch funkcji. Funkcje przyjmują dwa argumenty. Każda z nich ma sprawdzić czy pierwsza liczba przekazana jako argument jest podzielna przez drugą.

Pierwsza funkcja wykorzystuje zwykłą konstrukcję `if / else`:

```
1 bool czy_podzielna_przez(int liczba, int dzielnik) {  
2     if (liczba % dzielnik == 0)  
3         return true;  
4     else  
5         return false;  
6 }
```

Druga funkcja wykorzystuje wyrażenie trójargumentowe:

```
1 bool czy_podzielna_przez(int liczba, int dzielnik) {  
2     return liczba % dzielnik == 0 ? true : false;  
3 }
```

Obie powyższe funkcje mogą być zapisane również następująco:

```
1 bool czy_podzielna_przez(int liczba, int dzielnik) {
2     return liczba % dzielnik == 0
3 }
```

Innym przykładem zastosowania wyrażenia trójargumentowego jest podanie jednego z dwóch komunikatów, zależnych od spełnienia pewnego warunku.

```
1 int wartosc = 15;
2 cout << (wartosc > 20 ? "Dużo" : "Mało");
3 // Mało
4
5 wartosc = 23;
6 cout << (wartosc > 20 ? "Dużo" : "Mało");
7 // Dużo
```

Instrukcja switch w języku C++

Instrukcja wielokrotnego wyboru w języku C++ rozpoczyna się od słowa kluczowego switch:

```
1 int numer_zawodnika = 2;
2
3 switch (numer_zawodnika) {
4     case 0:
5         // wypisz informacje o zawodniku 0.
6         break;
7     case 1:
8         // wypisz informacje o zawodniku 1.
9         break;
10    case 2:
11        // wypisz informacje o zawodniku 2.
12        break;
13    case 3:
14        // wypisz informacje o zawodniku 3.
15        break;
16    case 4:
17        // wypisz informacje o zawodniku 4.
```

```
18         break;
19     }
```

Przedstawiliśmy sposób użycia instrukcji `switch` ze zmienną typu całkowitego. Nie ma jednak przeszkód, aby wykorzystać zmienne typu znakowego (`char`):

```
1 char litera = 'c';
2
3 switch (litera) {
4     case 'a':
5         // wartość zmiennej litera to 'a'
6         break;
7     case 'b':
8         // wartość zmiennej litera to 'b'
9         break;
10    case 'c':
11        // wartość zmiennej litera to 'c'
12        break;
13    case 'd':
14        // wartość zmiennej litera to 'd'
15        break;
16    case 'e':
17        // wartość zmiennej litera to 'd'
18        break;
19 }
```

Ważne!

Definiując zmienne znakowe użyliśmy znaków apostrofu (`'`), a nie cudzysłowów. W języku C++ stosujemy apostrofy, kiedy chcemy przypisać zmiennej typu `char` konkretny znak. Użycie cudzysłowu oznacza, że chodzi o ciąg znaków. Jest on zawsze zakończony tzw. znakiem pustym, mającym w kodzie ASCII wartość 0.

Zmienna typu `char` może przechowywać tylko pojedyncze znaki. Zastosowanie cudzysłowów spowodowałoby dodanie znaku pustego.

Polecenie `break`

Przedstawiając składnię komendy `switch` użyliśmy instrukcji `break`. Kończy ona zestaw poleceń towarzyszących konkretnemu przypadkowi (`case`). Gdy program napotka komendę `break`, przerywane jest wykonywanie całej instrukcji wyboru wielokrotnego.

Przyjrzyjmy się ostatniemu przykładowi. Gdy zacznie być przetwarzana linia 10, program sprawdzi, czy wartość zmiennej `litera` jest równa przypadkowi `case 'c'`. Ponieważ okaże się, że tak właśnie jest, rozpocznie się wykonywanie fragmentu kodu zapisanego w linii numer 11. Później pojawia się komenda `break`. Oznacza to zakończenie instrukcji `switch` i przejście za linię numer 19.

Łączenie warunków

Wiadomo już, dlaczego stosujemy instrukcję `break`. Pomyśl, jak zachowa się poniższy program:

```
1 char litera = 'B';
2
3 switch(litera) {
4     case 'a':
5     case 'A':
6         // wartość zmiennej 'litera' to 'a' lub 'A'
7         break;
8     case 'b':
9     case 'B':
10        // wartość zmiennej 'litera' to 'b' lub 'B'
11        break;
12 }
```

Użycie dwóch słów kluczowych `case` spowoduje, że możemy dokonać wyboru na podstawie większej liczby wyrażeń. Warto pamiętać, że po wykonaniu instrukcji program będzie kontynuował sprawdzanie następnych warunków, chyba że natrafi na instrukcję `break`.

Skoro wartość zmiennej `litera` to 'B', wykona się linijka 10. Efekt byłby taki sam, gdyby `litera` była równa 'b'.

Polecenie default

W instrukcji wyboru wielokrotnego mamy do dyspozycji polecenie `default`. Oznacza ono: „w każdym przypadku”.

Innymi słowy `default` towarzyszy instrukcjom wykonywanym w sytuacji, gdy wartość zmiennej przekazanej poleceniu `switch` nie odpowiada żadnemu przypadkowi. Mamy dzięki temu pewność, że program wykona fragment kodu przypisany poleceniu `default`. Oto przykład:

```
1 int numer_zawodnika = 40;
2
3 switch (numer_zawodnika) {
4     case 0:
5         // wypisz informacje o zawodniku z numerem 0.
6         break;
7     case 1:
8         // wypisz informacje o zawodniku z numerem 1.
9         break;
10    case 2:
11        // wypisz informacje o zawodniku z numerem 2.
12        break;
13    default:
14        // nie ma zawodnika o podanym numerze
15 }
```

Wartość zmiennej `numer_zawodnika` nie pojawiła się w opisie żadnego przypadku, a zatem wykonane zostaną instrukcje towarzyszące słowu kluczowemu `default`. Jeżeli ten fragment kodu jest zapisany na końcu instrukcji wielokrotnego wyboru, nie ma potrzeby używania komendy `break`. W innym przypadku jest to niezbędne.

Słownik

break

używane w języku C++ polecenie służące do opuszczenia instrukcji `switch` po wykonaniu sekwencji komend odpowiadającej konkretnemu przypadkowi (`case`)

Schemat interaktywny

Polecenie 1

Napisz program, który wyświetla informację zwrotną w zależności od oceny otrzymanej ze sprawdzianu.

Specyfikacja problemu:

Dane:

- ocena - liczba naturalna z przedziału $\langle 2, 6 \rangle$

Wynik:

Komunikat dopasowany do danej oceny:

- Ocena 6: „Wspaniale!”
- Oceny 5 i 4: „Bardzo dobrze”
- Ocena 3: „Spróbuj powtórzyć najważniejsze informacje”
- Ocena 2: „Podejdź do sprawdzianu jeszcze raz”
- Ocena 1: „Ten materiał sprawił ci trudność, może porozmawiaj na ten temat z osobą prowadzącą przedmiot?”

Polecenie 2

Porównaj swoje rozwiązanie z filmem.

Wystąpił błąd

Wybór wielokrotny

Instrukcja switch w C++



Film dostępny pod adresem </preview/resource/RrasWvxnypz30>

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

Film nawiązujący do treści lekcji dotyczący wyboru wielokrotnego w instrukcji switch w C++.




Polecenie 3

Uczniowie napisali prosty kalkulator, który wykonuje cztery podstawowe działania na dwóch liczbach. Zapoznaj się ze schematem i napisz ten program, używając języka C++. Przetestuj jego działanie dla różnicy 72 i 48.

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

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

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Ćwiczenie 2



Ćwiczenie 3



Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Wybór wielokrotny w języku C++

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres 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 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):

- Zastosujesz wyrażenia trójargumentowe.
- Poznasz składnię instrukcji wielokrotnego wyboru.
- Przekonasz się, że instrukcja `switch` w pewnych przypadkach okazuje się bardziej efektywna niż instrukcja warunkowa.
- Wykorzystasz instrukcję `switch` do rozwiązania prostych zadań.

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.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Wybór wielokrotny w języku C++”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Nauczyciel wyświetla uczniom temat zajęć oraz cele. Prosi, by na ich podstawie uczniowie sformułowali kryteria sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Uczniowie tworzą pytania dotyczące tematu zajęć, na które odpowiedzą w trakcie lekcji.

Faza realizacyjna:

1. Uczniowie rozwiązują Polecenie 1 z sekcji „Schemat interaktywny”.

2. **Ćwiczenie umiejętności.** Uczniowie, pracując w parach, wykonują ćwiczenie nr 1 z sekcji „Sprawdź się”. Nauczyciel sprawdza poprawność pisanych kodów, porównuje je i omawia wraz z uczniami. Wskazuje najbardziej efektywne rozwiązanie.
3. Nauczyciel dzieli uczniów na grupy czteroosobowe. Uczniowie wykonują ćwiczenie nr 2: „Napisz program, który sprawdzi, czy różnica między wartościami zmiennych `wiek_osoby_b` i `wiek_osoby_a` wynosi 1, 2 lub 3, czy też jest inna. W każdym z wymienionych przypadków należy wyświetlić odpowiedni komunikat.” a następnie każda grupa wyznacza jedną osobę, którą wymienia się z inną grupą. W nowych zespołach uczniowie porównują swój kod i wybierają najbardziej efektywne rozwiązanie.

Faza podsumowująca:

1. Na koniec zajęć nauczyciel raz jeszcze wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie”. W odniesieniu do ich realizacji dokonuje szczegółowej oceny rozwiązania zastosowanego przez wybranego ucznia.
2. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązania ćwiczeń z programowania w języku C++.

Praca domowa:

1. Uczniowie wykonują ćwiczenie nr 3: „Małgosia otrzymuje wynagrodzenie za wykonywanie prac domowych. Każda czynność jest opłacana według określonej stawki. Napisz program, który na podstawie wysokości otrzymanej kwoty poinformuje, czym zajmowała się Małgosia.” z sekcji „Sprawdź się”.

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:

- Treści w sekcji „Schemat interaktywny” można wykorzystać na lekcji jako podsumowanie i utrwalenie wiedzy uczniów.