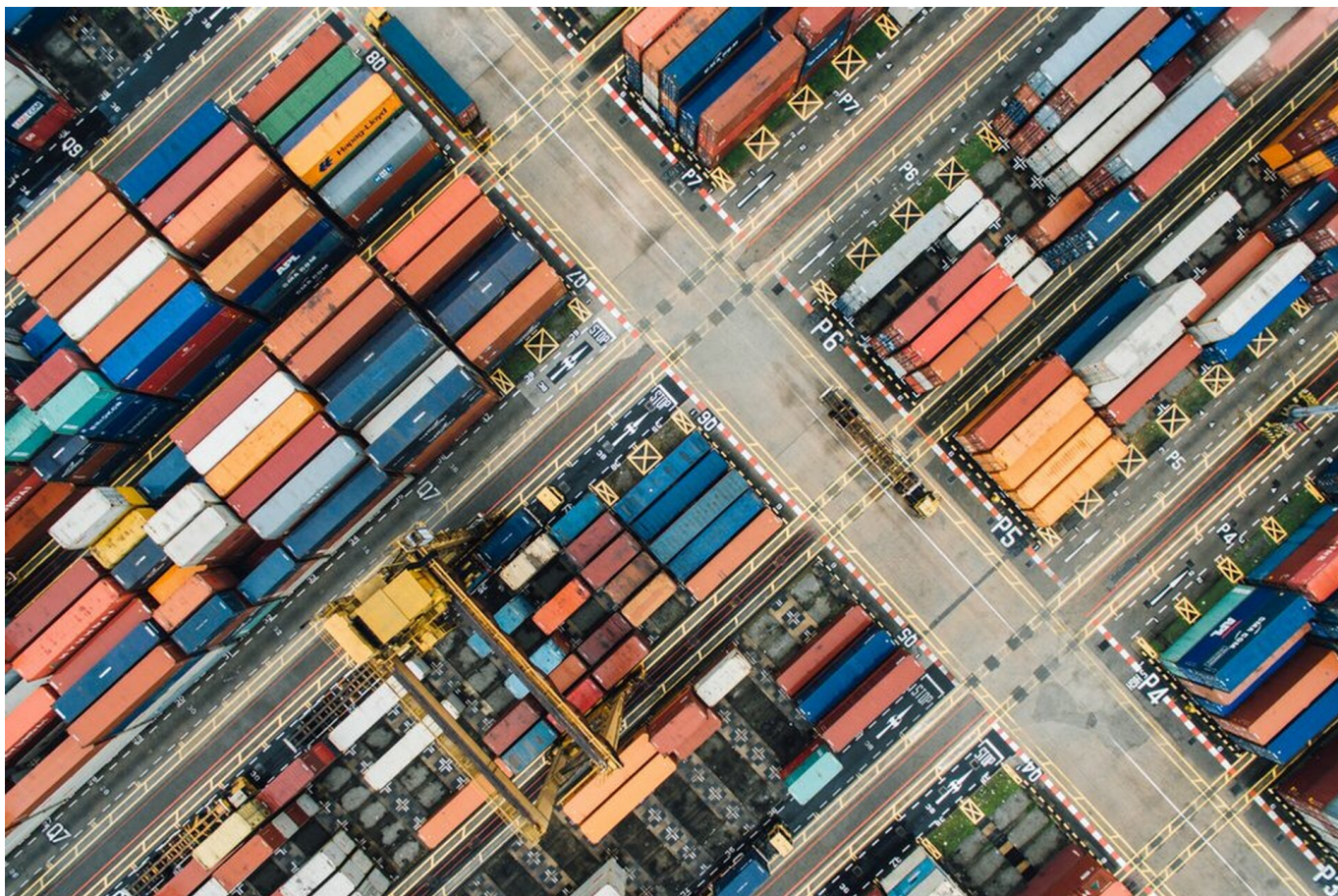




Podstawowe typy zmiennych w języku C++

- [Wprowadzenie](#)
- [Film samouczek](#)
- [Przeczytaj](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Podstawowe typy zmiennych w języku C++

Źródło: Chuttersnap, domena publiczna.

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

Wiesz już, że programy wykorzystują pojemniki na dane – zmienne. Dowiedzieliśmy się tego z e-materiału [Podstawowe typy zmiennych](#).

Czy wiesz jednak, jaka jest ich specyfika w danym języku programowania?

W tym e-materiale zapoznamy się z podstawowymi typami zmiennych w języku C++.

Charakterystykę podstawowych typów zmiennych w innych językach programowania znajdziesz w e-materiałach:

- [Podstawowe typy zmiennych w języku Java](#),
- [Podstawowe typy zmiennych w języku Python](#).

Więcej zadań? Sięgnij do [Podstawowe typy zmiennych – ćwiczenia](#).

Twoje cele

- Przeanalizujesz różnice między zmiennymi w języku C++.
- Dobierzesz do realizowanego algorytmu zmienne właściwego typu.
- Rozwiążesz proste problemy z zastosowaniem zmiennych w języku C++.

Film samouczek

Polecenie 1

Utwórz cztery zmienne, których zadaniem będzie przechowywanie następujących informacji na temat użytkownika:

- imię,
- wiek,
- średnia ocen,
- to, czy użytkownik skończył 18 lat.

```
1 #include <iostream>
2
3 int main () {
4
5     // Tutaj dodaj kod. Żeby coś wypisać, użyj polecenia:
6     std::cout
7 }
```

```
1
```

Polecenie 2

Zapoznaj się z filmem i porównaj z nim swoje rozwiązanie.

Podstawowe typy zmiennych

Typy zmiennych w C++

Film dostępny pod adresem </preview/resource/RC8DFZcUAJo37>

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

Nagranie filmowe dotyczące typu danych C++

Polecenie 3

Przeanalizuj prezentację, a następnie podaj jedno dodatkowe zastosowanie dla każdego poznanego typu zmiennej.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Pq3Xegode>

Założmy, że tworzymy program, który ma za zadanie przyjmować różne informacje na temat biletów autobusowych. Musimy wprowadzić odpowiednie typy zmiennych.

Są to: numer autobusu, w którym zostały skasowane bilety; informacje, czy bilety są jeszcze ważne; pozostały czas ważności (w sekundach); typ biletu; pierwsza litera nazwiska posiadacza. Program ma przechowywać informacje o trzech biletach jednocześnie.

Ważność biletu to maksymalnie 24 godziny.

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Pq3Xegode>

Zacznijmy od numeru autobusu.

Jest to wartość liczbowa, więc już na początku możemy odrzucić zmienne typu logicznego i znakowego. Użyjemy jednego z typów zmiennych reprezentujących liczby.

Numery autobusów najczęściej są wyrażane w liczbach całkowitych, więc zmienna będzie właśnie tego typu. Nigdy nie są one ujemne oraz z zasady, ich wartość jest mniejsza niż 1000.

3

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Pq3Xegode>

Dogodnym typem zmiennej w przypadku numeru autobusu będzie więc `short`.

Zmienne wprowadzimy w następujący sposób:

```
1 short bilet1_nr_autobusu;  
2 short bilet2_nr_autobusu;  
3 short bilet3_nr_autobusu;
```

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Pq3Xegode>

Informacja, czy bilety są jeszcze ważne, wymaga, żeby zmienna była w stanie przyjąć przynajmniej dwie wartości.

Oznacza to, że zastosujemy dowolny typ zmiennej, gdzie przyjmujemy dwa warianty – prawdę i fałsz.

Przykładowo, używając zmiennej typu znakowego, możemy przyjąć, że *t* oznacza prawdę, a *n* – fałsz.

5

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Pq3Xegode>

Możliwości jest wiele, jednak warto użyć zmiennej, która rzeczywiście może przyjąć wyłącznie dwie wartości – prawda i fałsz. Jest to zmienna typu logicznego.

Wprowadzamy ją następująco:

```
1 bool bilet1_czy_wazny;  
2 bool bilet2_czy_wazny;  
3 bool bilet3_czy_wazny;
```

6

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Pq3Xegode>

Informacja o pozostałym czasie ważności, podobnie jak numer autobusu, należy do wartości liczbowych. Podana będzie

w sekundach, dlatego musimy obliczyć, jaka będzie największa wartość przechowywana w tej zmiennej.

Mnożymy więc 24 przez 60, a następnie znowu przez 60. Wynik to 86400.

Oznacza to, że nie możemy użyć zmiennej typu `short`, której maksymalny zasięg to 32767, a nawet `unsigned short`, z zakresem do 65535.

7

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Pq3Xegode>

Odpowiednim typem zmiennej będzie `int`, ponieważ jej zakres znacząco przewyższa maksymalną wartość liczbową sekund, które mają być w niej przechowane.

Zmienne wprowadzamy tak:

```
1 int bilet1_pozostały_czas;  
2 int bilet2_pozostały_czas;  
3 int bilet3_pozostały_czas;
```

8

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Pq3Xegode>

W przypadku typu biletu, podobnie jak z informacją, czy jest on jeszcze ważny, problem można rozwiązać na kilka sposobów. Mamy dwa typy biletu: ulgowy i normalny.

Raz jeszcze określamy specyficzne znaki bądź liczby, które będą odpowiadać określonym

typom biletu. Typ zmiennej będzie więc zależał od sposobu, w jaki rozpoznajemy ulgę.

9

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Pq3Xegode>

Użyjemy zmiennej typu logicznego:

```
1 bool bilet1_czy_ulgowy;  
2 bool bilet2_czy_ulgowy;  
3 bool bilet3_czy_ulgowy;
```

Warto zwrócić uwagę na nazewnictwo zmiennych. Sugeruje ono, że zmienne przechowują informacje o tym, czy bilet jest ulgowy, i możemy odpowiednio zinterpretować wartości przez nie przyjęte.

Przykładowo, `false` będzie oznaczać brak ulgi.

10

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Pq3Xegode>

Pierwsza litera nazwiska posiadacza to znak. Z tego względu zastosujemy tu zmienną typu znakowego. Wprowadzamy ją w następujący sposób:

```
1 char bilet1_litera_nazw;  
2 char bilet2_litera_nazw;  
3 char bilet3_litera_nazw;
```

Udało nam się znaleźć i wprowadzić odpowiednie typy zmiennych dla danych. Teraz wymyśl dodatkowe zastosowanie dla każdego opisanego typu zmiennych.

Przeczytaj

Podsumowanie informacji na temat zmiennych w języku C++

Typy całkowite

Jeżeli zmienna ma przechowywać liczbę całkowitą, wykorzystujemy jeden z typów całkowitych. Od tego, który z nich wybierzemy, zależy rozmiar zmiennej, czyli ilość miejsca zarezerwowanego dla niej w pamięci. Rozmiar przekłada się z kolei na zakres zmiennych, czyli wielkość zapisywanych liczb.

W języku C++ mamy do dyspozycji następujące typy zmiennych całkowitych:

Nazwa	Rozmiar	Zakres
short	2 bajty	[−32768, 32767]
int	4 bajty	[−2147483648, 2147483647]
long	4 bajty	[−2147483648, 2147483647]
unsigned short	2 bajty	[0, 65535]
unsigned int	4 bajty	[0, 4294967295]
unsigned long	4 bajty	[0, 4294967295]

Tabela przedstawia najczęściej używane typy zmiennych całkowitych. Głównym czynnikiem decydującym o wyborze typu jest przewidywana maksymalna wartość liczby, jaką zmienna ma przechowywać.

Rozmiar zmiennych wyrażony jest w bajtach. Zmienna typu `short` zajmuje w pamięci 2 bajty, zaś zmienna typu `int` – 4 bajty.

Jeżeli nie przewidujemy, że zmienna będzie przechowywać wartość ujemną, możemy użyć słowa kluczowego `unsigned`. W rezultacie podwoi się dodatnia część zakresu (stanie się to kosztem części ujemnej).

Założmy, że postanowiliśmy wprowadzić do programu zmienną, która przechowuje liczbę 2 .

Wykorzystujemy typ zmiennej całkowitej `short`, który pozwala zapisywać stosunkowo małe liczby. Ponieważ w programie możemy użyć więcej niż jednej zmiennej danego typu, każdej z nich musimy nadać inną nazwę.

Definiując zmienną w kodzie programu najpierw podajemy jej typ, a następnie nazwę. Instrukcję kończymy znakiem średnika:

```
1 short zmienna;
```

Teraz przyjmijmy, że wprowadzamy do programu zmienną, która ma przechowywać liczbę 33000.

W tym przypadku wyjdziemy już poza zakres zmiennej całkowitej typu `short`.

Mamy wówczas dwie możliwości:

1. Wybrać typ zmiennej całkowitej `int` (będziemy mogli użyć jej także do przechowywania liczb ujemnych).
2. Jeżeli będziemy przechowywać wyłącznie liczby dodatnie, możemy niemal podwoić zakres zmiennej typu `short`, rezygnując jednak z zapisywania liczb ujemnych. Efekt taki osiągniemy dodając słowo `unsigned` przed określeniem typu całkowitego. W opisywanym przypadku byłby nim `unsigned short`.

Definicje zmiennych odpowiadających obydwu opisanym wariantom przedstawiamy niżej:

```
1 int zmienna1 = 33000;  
2 unsigned short zmienna2 = 33000;
```

Typy rzeczywiste

Jeśli zamierzamy posługiwać się ułamkami, wybierzemy zmienną typu rzeczywistego:

Nazwa	Rozmiar	Zakres
<code>float</code>	4 bajty	6–7 cyfr po przecinku
<code>double</code>	8 bajtów	15–16 cyfr po przecinku
<code>long double</code>	12 bajtów	19–20 cyfr po przecinku

W tabeli podano najczęściej używane typy zmiennych rzeczywistych.

Założmy, że wprowadzamy do programu zmienną, która ma przechowywać liczby z dokładnością do 10 miejsc po przecinku. Zastosujemy typ `double`; samą zmienną zdefiniujemy w następujący sposób:

```
1 double zmienna;
```

Jeżeli zależy nam na przechowywaniu liczb mających znacznie dłuższe rozwinięcia dziesiętne, przyda się typ `long double`.

A oto sposób definiowania zmiennej typu `long double`:

```
1 long double zmienna;
```

Typ znakowy

Zmienne typu znakowego służą do przechowywania symboli takich jak litery, cyfry, operatory arytmetyczne, znaki przestankowe i specjalne. Należy zdawać sobie sprawę z faktu, że każdej zmiennej typu znakowego odpowiada określony [kod ASCII](#).

W kodzie ASCII literze 'a' odpowiada liczba 97. Z kolei cyfrze 0 przyporządkowano wartość liczbową 48. Pisząc programy będziemy wykorzystywać te właściwości zmiennych typu znakowego.

Zmienne typu znakowego definiujemy podobnie jak zmienne służące do przechowywania liczb:

```
1 char zmienna;
```

Najpierw podajemy typ zmiennej, a następnie jej nazwę. Dobrą praktyką jest nadawanie zmiennym nazw kojarzących się z konkretnymi obiektami. Gdybyśmy chcieli zapisać w zmiennej pierwszą literę imienia, powinniśmy wybrać nazwę `pierwsza_litera_imienia`, a nie `zmienna`.

Z tablicą ASCII warto zapoznać się choćby pobieżnie i zapamiętać przynajmniej liczby odpowiadające literom A i a oraz cyfrze 0. Wiedza taka pozwala obliczyć kod odpowiadający każdej literze i cyfrze.

Typ logiczny

Zmienne tego typu przechowują jedną z dwóch wartości: `true` (prawda) albo `false` (fałsz).

Wartość logiczna `true` jest równa 1, a `false` ma wartość 0.

Nazwa	Rozmiar	Zakres
<code>bool</code>	1 bajt	<code>false</code> , <code>true</code>

Zmienne logiczne definiujemy używając znanej już składni:

```
1 bool zmienna;
```

Ciekawostka

W języku C++ każda liczba większa od 0 (nie tylko 1) przypisana do zmiennej typu `bool` oznacza `true` (prawda).

```
1 bool a = 1;  
2 bool b = 30;
```

Dwie zdefiniowane wyżej zmienne będą przechowywać taką samą wartość: `true` (prawda).

Słownik

kod ASCII

(ang. *American Standard Code for Information Interchange*) system kodowania znaków, w którym literom łacińskim, cyfrom oraz innym symbolom przyporządkowane są unikatowe odpowiedniki liczbowe z zakresu 0 – 127

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Wskaż, jaki typ zmiennej zastosujemy do przechowania litery A.

☐ short

☐ bool

☐ int

☐ char

Ćwiczenie 2



Wskaż, jaki typ zmiennej zastosujemy do przechowania liczby 3, 14.

☐ float

☐ char

☐ bool

☐ int

Ćwiczenie 3



Zaznacz wszystkie poprawne odpowiedzi. Które z wymienionych typów zmiennych mogą przechować wartość 80000?

☐ double

☐ char

☐ short

☐ int

Ćwiczenie 4



Uzupełnij zdanie.

Do przechowania wartości `true` zastosujemy .

float

bool

int

char

Ćwiczenie 5



Opisz krótko typ zmiennych: całkowite.

Ćwiczenie 6



Opisz krótko typ zmiennych: rzeczywiste.

Ćwiczenie 7



Opisz krótko typ zmiennych: znakowe.

Ćwiczenie 8



Opisz krótko typ zmiennych: logiczne.

Dla nauczyciela

Autor: Zespół autorski Contentplus.pl sp. z o.o.

Przedmiot: Informatyka

Temat: Podstawowe typy zmiennych 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:

1) projektuje i tworzy rozbudowane programy w procesie rozwiązywania problemów, wykorzystuje w programach dobrane do algorytmów struktury danych, w tym struktury dynamiczne i korzysta z dostępnych bibliotek dla tych struktur;

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):

- Przeanalizujesz różnice między zmiennymi w języku C++.
- Dobierzesz do realizowanego algorytmu zmienne właściwego typu.
- Rozwiążesz proste problemy z zastosowaniem zmiennych w języku C++.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

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

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: „Podstawowe typy zmiennych w języku C++”. Uczniowie mają zapoznać się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Nauczyciel wyświetla uczniom temat, wskazuje cele zajęć oraz ustala z uczestnikami zajęć kryteria sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel zadaje uczniom pytania dotyczące ich aktualnego stanu wiedzy w obszarze poruszanego tematu i programowania, np.
 - jakie znasz typy zmiennych stosowane przy programowaniu w języku C++?
 - co to jest typ zmiennej całkowitej?
 - czym różni się typ znakowy od logicznego?

Chętni uczniowie udzielają na nie odpowiedzi.

Faza realizacyjna:

1. **Praca z tekstem.** Metodą burzy mózgów uczniowie referują najważniejsze informacje, jakie znaleźli w sekcji „Przeczytaj”, przygotowując się do lekcji. W razie potrzeby nauczyciel wyjaśnia niezrozumiałe kwestie.
2. **Praca z multimediami.** Uczniowie zapoznają się z treścią polecenia 1 z sekcji „Film samouczek” i w parach rozwiązują je. W kolejnym kroku porównują swoje rozwiązanie z zaprezentowanym w filmie.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują ćwiczenia nr 1-8 z sekcji „Sprawdź się”. Nauczyciel sprawdza poprawność wykonanych zadań, omawiając je wraz z uczniami.

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. Na koniec zajęć z programowania w C++ nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak...”.

Praca domowa:

1. Uczniowie wykonują polecenie 3 z sekcji „Film samouczek”.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).

Wskazówki metodyczne:

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