Fragment książki
G.P.Korbaś, Podstawy C++ Zbiór zadań z rozwiązaniami e-book, wyd. GA EDUKACJA, 2020
szczegóły: www.gaedukacja.pl

Grzegorz Paweł Korbaś

Podstawy C++

Zbiór zadań z rozwiązaniami
e-book

Opole 2020

wydawnictwo GA EDUKACJA
www.gaedukacja.pl

Fragment książki
G.P.Korbaś, Podstawy C++ Zbiór zadań z rozwiązaniami e-book, wyd. GA EDUKACJA, 2020
szczegóły: www.gaedukacja.pl

Prośba do czytelnika

Wydawnictwo prosi, aby nie rozpowszechniać tej publikacji w sposób nielegalny.

E-book można zakupić na stronie www.gaedukacja.pl.

Jeśli zakupiłeś tę publikację – dziękujemy!
Jeśli e-book trafił w Twoje ręce w nielegalny sposób – zapłać za niego przez podaną stronę.

Dzięki Twojej uczciwości wydawnictwo i autor będą mogli opracować kolejną pomoc dydaktyczną.

Informacje o autorze

Autor publikacji, dr inż. Grzegorz Paweł Korbaś, jest doświadczonym dydaktykiem. Przez wiele lat wykładał programowanie w języku C/C++ na Politechnice Opolskiej oraz uczył podstaw programowania w liceum.

Informacje o e-booku

Autor:	Grzegorz Paweł Korbaś
Tytuł:	Podstawy C++ Zbiór zadań z rozwiązaniami e-book
Wydawnictwo:	GA EDUKACJA
Rok publikacji:	2020
Wydanie:	pierwsze poprawione
ISBN:	978-83-943917-8-2

Uwaga: Niniejszy e-book jest elektroniczną wersją książki: Grzegorz Paweł Korbaś, Podstawy C++ Zbiór zadań z rozwiązaniami, wydawnictwo czytnia.pl, Opole 2011, wydanie pierwsze, ISBN 978-83-943917-7-5.
Naniesiono poprawki. Poza tym układ treści nie uległ zmianie.

Wszelkie prawa zastrzeżone. Żadna część niniejszej publikacji nie może być w żaden sposób reprodukowana lub kopiowana bez pisemnej zgody wydawnictwa.

Osoba, która zakupiła prawo używania e-booka może go skopiować lub wydrukować wyłącznie do własnego, indywidualnego użytku.

Fragment książki
G.P.Korbaś, Podstawy C++ Zbiór zadań z rozwiązaniami e-book, wyd. GA EDUKACJA, 2020
szczegóły: www.gaedukacja.pl

Spis treści

Fragment książki
G.P.Korbaś, Podstawy C++ Zbiór zadań z rozwiązaniami e-book, wyd. GA EDUKACJA, 2020
szczegóły: www.gaedukacja.pl

Fragment książki
G.P.Korbaś, Podstawy C++ Zbiór zadań z rozwiązaniami e-book, wyd. GA EDUKACJA, 2020
szczegóły: www.gaedukacja.pl

Wstęp

Zawartość zbioru

Niniejszy zbiór zawiera 138 zadań z podstaw języka C++ wraz z pełnymi rozwiązaniami i komentarzami do rozwiązań. Zadania zostały zgromadzone w dziesięciu rozdziałach. Ponadto, w dodatkach do zbioru, opisano sposób uruchamiania programów na czterech różnych kompilatorach oraz przedstawiono podstawowe informacje dotyczące języka C++.

Zbiór powstawał z myślą o osobach, które rozpoczynają naukę podstaw C++ i nie mają doświadczenia związanego z programowaniem. Do podstawowych użytkowników mogą należeć:
- uczniowie szkół ponadgimnazjalnych uczący się programowania
- studenci pierwszych semestrów kierunków technicznych
- osoby samodzielnie poznające podstawy języka C++

Zbiór nie jest podręcznikiem. Powinien być używany w połączeniu z rzetelnymi źródłami wiedzy o języku C++ (lekcjami lub wykładami, literaturą, źródłami z Internetu). Zadania powinny być używane jako materiał do samodzielnych lub kierowanych ćwiczeń.

Sposób korzystania ze zbioru

Zadania ze zbioru proponuje się wykonywać w takiej kolejności, w jakiej zostały zaproponowane w zbiorze. Kolejne rozdziały wymagają utrwalonej wiedzy i umiejętności z rozdziałów poprzednich. Zadania trudniejsze zostały oznaczone gwiazdką - można je ominąć i wrócić do nich w późniejszym czasie. Z punktu widzenia dydaktycznego autor proponuje korzystać ze zbioru zgodnie z następującym schematem:
1. Poznanie zagadnień teoretycznych dotyczących danego materiału na podstawie dostępnych, wiarygodnych źródeł
2. Próba rozwiązania kolejnych zadań i porównanie własnych rozwiązań z rozwiązaniami zaproponowanymi w zbiorze. Wszelkie różnice powinny zostać przemyślane, aby zrozumieć z czego wynikają - czy własna propozycja jest inną wersją o takiej samej jakości działania, czy też różnice są istotne i świadczą o braku opanowania materiału. Pomocą w analizie porównawczej mogą być komentarze do rozwiązań.
3. Jeśli brakuje umiejętności, aby samodzielnie rozwiązać przedstawione zadania, to powinno się szczegółowo przeanalizować proponowane rozwiązania tak, aby je w pełni zrozumieć i umieć napisać podobne. Po krótkim czasie - rzędu 1-2 dni - powinna zostać podjęta kolejna próba samodzielnego rozwiązania zadania. To postępowanie należy powtarzać aż do skutku.
4. Po przerobieniu większej partii materiału można wybrać losowo 3-4 zadania i zrobić sobie samodzielnie sprawdzian. Sprawdzian ten warto samodzielnie ocenić korzystając z rozwiązań i komentarzy zbioru.

Fragment książki
G.P.Korbaś, Podstawy C++ Zbiór zadań z rozwiązaniami e-book, wyd. GA EDUKACJA, 2020
szczegóły: www.gaedukacja.pl

Errata do zbioru i prośba o uwagi

Wszystkie zadania, rozwiązania i komentarze były wielokrotnie sprawdzane, jednak w zbiorze mogą zostać znalezione pomyłki. Dlatego też na stronie www.gaedukacja.pl w zakładce „Skontaktuj się z nami" korzystający ze zbioru znajdzie odpowiedni formularz kontaktowy, poprzez który będzie można wskazać dostrzeżone pomyłki. Uwagi można również zgłaszać e-mailem: sklep@gaedukacja.pl.

Fragment książki
G.P.Korbaś, Podstawy C++ Zbiór zadań z rozwiązaniami e-book, wyd. GA EDUKACJA, 2020
szczegóły: www.gaedukacja.pl

Zadania

1. Podstawowe operacje

Zad.1.1 Napisz program, który: a) w pierwszym wierszu wypisuje tekst „START”, b) w kolejnym wierszu czeka na wprowadzenie dowolnego znaku.

Zad.1.2 Napisz program, który: a) w pierwszym wierszu wypisuje tekst „123+321=”, b) w drugim wierszu wypisuje tekst „444”, c) w kolejnym wierszu wyświetla komunikat systemowy i czeka na naciśnięcie dowolnego przycisku (zastosuj pauzę systemową).

Zad.1.3 Napisz program, który: a) pobiera dwie liczby całkowite L1 i L2, b) w kolejnych wierszach wyświetla ich sumę oraz iloczyn, c) w kolejnym wierszu czeka na wprowadzenie dowolnego znaku.

Zad.1.4 Napisz program, który: a) pobiera dwie liczby całkowite u i v, b) oblicza sumę i różnicę tych liczb i zapamiętuje w zmiennych sum i roz, c) w kolejnym wierszu wyświetla tekst „u+v=” oraz sumę, d) w kolejnym wierszu wyświetla „u-v=” oraz różnicę, e) w kolejnym wierszu czeka na wprowadzenie dowolnego znaku, f) zwraca do systemu liczbę -1.

Zad.1.5 Napisz program, który: a) wypisuje tekst „x=”, b) w tym samym wierszu pobiera liczbę rzeczywistą (zmiennoprzecinkową) x, c) w kolejnym wierszu wypisuje tekst „y=”, d) w tym samym wierszu pobiera liczbę rzeczywistą y, e) w kolejnym wierszu wypisuje tekst „x/y=”, f) w tym samym wierszu wypisuje wynik dzielenia x/y.

Zad.1.6 Napisz program, który potrafi odszukać miejsce zerowe dowolnej prostej zadanej w postaci kierunkowej $y=ax+b$ (przy założeniu, że $a\neq 0$). Program: a) pobiera od użytkownika współczynniki a i b, b) w kolejnym wierszu wypisuje znalezione miejsce zerowe, c) w kolejnym wierszu wykonuje pauzę systemową.

Zad.1.7 Napisz program, który: a) pobiera trzy znaki, b) w kolejnych wierszach wypisuje wszystkie możliwe permutacje tych znaków. Przykładowo dla znaków 'K', 'O' i 'T' program wypisuje (w dowolnej kolejności): „KOT”, „KTO”, „OKT”, „OTK”, „TOK” i „TKO”.

Zad.1.8 Napisz program, który: a) pobiera długości boków prostokąta a, b, b) oblicza i wypisuje obwód tego prostokąta, c) oblicza i wypisuje pole tego prostokąta.

Zad.1.9 Napisz program, który: a) pobiera dwie liczby całkowite u i v, b) wypisuje, ile razy u mieści się (całkowicie) w v, c) w kolejnym wierszu wypisuje resztę z dzielenia v przez u, d) w kolejnym wierszu wypisuje wartość rzeczywistą v/u.

Zad.1.10 Napisz program, który: a) pobiera od użytkownika liczbę rzeczywistą R (zakładamy, że jest dodatnia), b) w kolejnych wierszach wyświetla wartości $\sqrt{R}$, $\ln R$, R^{-7} oraz R pomnożone przez π.

Fragment książki
G.P.Korbaś, Podstawy C++ Zbiór zadań z rozwiązaniami e-book, wyd. GA EDUKACJA, 2020
szczegóły: www.gaedukacja.pl

Zad.9.7 Wykorzystaj standardowy szablon `stack`, który dostarcza funkcjonalność stosu i napisz program, który: a) pobiera od użytkownika i zapamiętuje kolejne linijki tekstów, aż do uzyskania tekstu „koniec” (ostatni zapamiętany tekst), b) wyświetla zapamiętane teksty w odwrotnej kolejności.

Zad.9.8 Wykorzystaj standardowy szablon `vector`, który dostarcza funkcjonalność tablicy i napisz program, który: a) pobiera od użytkownika i zapamiętuje kolejne liczby całkowite aż do uzyskania podania zera (ostatnia zapamiętana liczba), b) wyświetla zapamiętane liczby posortowane. Do sortowania użyj standardowego algorytmu z biblioteki `algorithm`.

10. Wybrane algorytmy

Zad.10.1 Napisz program, który: a) pobiera liczbę rzeczywistą x, b) pobiera liczbę całkowitą nieujemną N (stopień wielomianu), c) pobiera N+1 liczb rzeczywistych a_N, a_{N-1}, .., a_0 (w podanej kolejności), d) oblicza i wypisuje wartość wielomianu $a_0+a_1x+a_2x^2+..+a_Nx^N$. Należy posłużyć się schematem Hornera.

Zad.10.2 Napisz program, który: a) pobiera od użytkownika liczbę rzeczywistą dodatnią X, b) oblicza i wypisuje wartość pierwiastka z X wyznaczoną z dokładnością co najmniej 0,001. Dokładność występuje w kodzie jako zdefiniowana stała. Należy posłużyć się algorytmem Newtona-Raphsona.

Zad.10.3 Napisz program, który: a) pobiera od użytkownika liczby całkowite dodatnie A i B, b) oblicza i wypisuje największy wspólny dzielnik liczb A i B. Należy posłużyć się algorytmem Euklidesa.

Zad.10.4 Napisz program, który: a) pobiera od użytkownika liczbę całkowitą dodatnią N, b) pobiera i zapamiętuje w tablicy N tekstów, c) sortuje tablicę, d) wypisuje wartości posortowane w kolejnych wierszach. Zastosuj sortowanie szybkie - napisz odpowiednią funkcję.

Zad.10.5 Napisz program, który: a) pobiera od użytkownika liczbę całkowitą dodatnią N, b) pobiera i zapamiętuje w tablicy N liczb całkowitych, c) sortuje tablicę, d) pobiera od użytkownika liczbę całkowitą L, e) sprawdza, czy liczba L znajduje się wśród zapamiętanych wcześniej liczb - jeśli tak to wypisuje „TAK. Zastosuj sortowanie szybkie, które jest dostępne w systemie. Zadbaj o optymalne wyszukiwanie liczby L w tablicy.

Zad.10.6 Napisz program, który: a) pobiera od użytkownika liczbę całkowitą X zapisaną w systemie dziesiętnym, b) wypisuje X w kolejnych wierszach w systemach dwójkowym i szesnastkowym.

Zad.10.7 Napisz program, który: a) pobiera od użytkownika linijkę tekstu, b) szyfruje linijkę szyfrem Cezara, zapamiętuje i wypisuje wynik, c) deszyfruje linijkę i wypisuje wynik. Szyfrowanie powinno działać dla znaków o kodach od 32 do 126 włącznie.

Fragment książki
G.P.Korbaś, Podstawy C++ Zbiór zadań z rozwiązaniami e-book, wyd. GA EDUKACJA, 2020
szczegóły: www.gaedukacja.pl

Rozwiązania

1. Podstawowe operacje

Roz.1.1 Kod rozwiązania:

```
#include <iostream>
using namespace std;
int main()
{
cout << "START" << endl;
char x;
cin >> x;
return 0;
}
```

Uwaga: Dołączenie biblioteki iostream jest konieczne ze względu na korzystanie z `cin` *oraz* `cout`*. Zamiast* `cout << "START" << endl;` *można napisać* `cout << "START\n";`*. Zastosowanie polecenia* `using namespace std;` *jest konieczna dla ustalenia tzw. standardowej przestrzeni nazw. Gdyby polecenie zostało pominięte, konieczne byłoby pisanie* `std::cout`*,* `std::cin`*,* `std::endl`*.*

Roz.1.2 Kod rozwiązania:

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
cout << "123+321=" << endl;
cout << "444" << endl;
system("pause");
return 0;
}
```

Uwaga: Polecenie `system` *może zachowywać się różnie w zależności od systemu operacyjnego. Aby stosować to polecenie należy dołączyć bibliotekę* `cstdlib` *(w niektórych kompilatorach).*

Roz.1.3 Kod rozwiązania:

```
#include <iostream>
using namespace std;
int main()
{
int L1, L2;
cin >> L1;
cin >> L2;
cout << L1 + L2 << endl;
cout << L1 * L2 <<endl;
char x;
cin >> x;
return 0;
}
```

Uwaga: Obydwie liczby L1 oraz L2 mogą zostać wprowadzone nie osobno, ale w jednym poleceniu `cin >> L1 >> L2;`

Roz.1.4 Kod rozwiązania:

```
#include <iostream>
using namespace std;
int main()
{
int u, v, sum, roz;
cin >> u >> v;
sum = u + v;
roz = u - v;
cout << "u+v=" << sum << endl;
cout << "u-v=" << roz << endl;
char x;
cin >> x;
return -1;
}
```

Uwaga: Zwrócenie do systemu wartości różnej od zera może być interpretowane jako podawanie kodu błędu, który wystąpił w programie. Reakcja na taką sytuację zależy od systemu, który wywołał program.

Roz.1.5 Kod rozwiązania:

```
#include <iostream>
using namespace std;
int main()
{
float x,y;
cout << "x=";
cin >> x;
cout << "y=";
cin >> y;
cout << "x/y=" << x/y << endl;
return 0;
}
```

Uwaga: Ponieważ x oraz y są typu zmiennoprzecinkowego, stosowane tu dzielenie to dzielenie zmiennoprzecinkowe.

Roz.1.6 Kod rozwiązania:

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
float a,b;
cin >> a >> b;
cout << -b/a << endl;
system("pause");
return 0;
}
```

Roz.1.7 Kod rozwiązania:

```
#include <iostream>
using namespace std;
int main()
{
```

```
for (int i=0; i<TAB.size();i++)
  cout << TAB[i] << endl;
return 0;
}
```

Uwaga: W standardowej bibliotece C++ o nazwie `vector` istnieje klasa `vector`, która dostarcza możliwości tablicy dynamicznej. Dodanie wartości na końcu tablicy realizuje metoda push_back. Metody `begin` i `end` zwracają tzw iteratory początku i końca tablicy. Korzysta z nich standardowa metoda sortowania szybkiego `sort` zawarta w bibliotece `algorithm`. Wartościowe będzie jeśli użytkownik zapozna się z dokumentacją klasy `vector` oraz z dostępnymi standardowymi algorytmami dostarczanymi w C++.

10. Wybrane algorytmy

Roz.10.1 Kod rozwiązania:

```
#include <iostream>
using namespace std;
int main()
{
int N;
double a,x,w=0;
cin>>x;
cin>>N;
cin>>w;
for (int i=0;i<N;i++)
  {cin>>a;
   w=w*x+a;
  }
cout<<w<<endl;
return 0;
}
```

Uwaga: W programie stosowany jest schemat Hornera - najbardziej efektywna metoda obliczania wartości wielomianu. Czytelnik może napisać ten program w postaci rekurencyjnej.

Roz.10.2 Kod rozwiązania:

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
double L,p,pop,e=0.001;
cin>>L;
p=L/2;
do
  {pop=p;
  p=(p+L/p)/2;}
while (e<abs(p-pop));
cout<<p<<endl;
cout<<p*p<<endl;
return 0;
}
```

Uwaga: Zastosowano metodę Newtona-Raphsona przyjmując wartość początkową jako połowę zadanej wartości p, dla której liczony jest pierwiastek.

Fragment książki
G.P.Korbaś, Podstawy C++ Zbiór zadań z rozwiązaniami e-book, wyd. GA EDUKACJA, 2020
szczegóły: www.gaedukacja.pl

Aneks B. Wybrane elementy języka C++

Podstawowa struktura programu

Prosty program w języku C++ może wyglądać następująco:

```
#include <iostream>
using namespace std;
int main()
{
cout << "Napis" << endl;
return 0;
}
```

- Dyrektywa `include` dołącza do programu plik o nazwie `iostream`, który umożliwia używanie różnych poleceń - w tym przypadku można użyć strumienia `cout`.
- Polecenie using namespace std; określa tzw. standardową przestrzeń nazw. Dzięki temu można pisać cout, a nie std::cout.
- `int main() { ... }`, to główna funkcja programu - każdy program w C++ musi mieć taką funkcję. Funkcja ta może zwrócić do systemu pewną wartość całkowitą.
- Polecenie `cout << "Napis" << endl;` służy do wypisania na ekranie podanego tekstu i skoku do nowej linijki.
- polecenie `return 0;` kończy działanie funkcji main i zwraca do systemu liczbę 0 (zwykle interpretowaną jako poprawne zakończenie programu).
- po każdym pojedynczym poleceniu w C++ znajduje się średnik. Jeśli w miejscu, gdzie dopuszcza się jedną instrukcję istnieje potrzeba wykonania większej ich liczby, należy zgrupować je w blok za pomocą klamer, np.: `{instr1; instr2;}`.

Typy zmiennych

Do przechowywania wartości w programie używa się zmiennych różnego typu, np.:

`int` - typ całkowitoliczbowy
`float` - typ zmiennoprzecinkowy (pojedynczej precyzji)
`double` - typ zmiennoprzecinkowy (podwójnej precyzji)
`char` - typ znakowy

Znaki w programie, umieszcza się w apostrofach, np.: 'x', 'Y'.

`string` - typ tekstowy (formalnie nie jest to typ, lecz klasa)

Aby stosować typ string konieczne jest dołączenie odpowiedniej biblioteki poleceniem `include <string>`. Teksty w programie umieszcza się w cudzysłowach, np.: „Witaj kolego”, „Z”.

`bool` - typ logiczny

Zmienna typu logicznego może przyjmować dwie wartości logiczne: **true** (prawda) albo **false** (fałsz). Można je również traktować jako liczby (odpowiednio 1 lub 0).

W niektórych przypadkach możliwe jest rzutowanie typów, czyli zmiana typu zmiennej na inny typ - może ono odbywać się jawnie lub niejawnie. Przykładowo jawne rzutowanie zmiennej a na typ int może mieć postać `(int) a`, albo `int(a)`.