

Podstawy Programowania

Monika Wrzosek

Instytut Matematyki
Uniwersytet Gdański

Matematyka 2017/18

Sprawy organizacyjne

- E-mail: mwrzosek@mat.ug.edu.pl
- Konsultacje: p.323
- Materiały: <http://www.mat.ug.edu.pl/~mwrzosek>

Literatura

- Stephen Prata, "Język C++. Szkoła programowania". Wydanie VI. Helion 2012.
- Jerzy Grębosz, "Symfonia C++ Standard".
Jerzy Grębosz, "Opus Magnum C++11".
Kody źródłowe przykładowych programów z książki i ćwiczenia z odpowiedziami: www.ifj.edu.pl/private/grebosz/opus.html
- Mirosław J. Kubiak, "C++. Zadania z programowania z przykładowymi rozwiązaniami", Helion 2011.

Sprawy organizacyjne

Zintegrowane środowisko programistyczne

- Microsoft Visual Studio
- Inne: Dev C++, Code::Blocks

Egzamin

Egzamin pisemny, złożony z zadań i pytań teoretycznych, np.:

- Przeanalizuj podany program i podaj wynik jego działania.
- Zmodyfikuj podany program, używając pętli while.
- Podaj przykład typu fundamentalnego.
- Zdefiniuj tablicę o elementach typu float złożoną z 10 wierszy i 8 kolumn. Do elementu o indeksie wierszowym 3 i kolumnowym 5 wpisz wartość 4.5.
- Jaka jest różnica między wskaźnikiem stałym a wskaźnikiem do obiektu stałego?

Szkic wykładu

- Instrukcje sterujące
- Typy
- Operatory
- Tablice
- Funkcje
- Rekurencja
- Wskaźniki
- Napisy
- Struktury
- Struktury dynamiczne

Wprowadzenie

Algorytm - skończony ciąg jasno zdefiniowanych czynności elementarnych, koniecznych do wykonania postawionego zadania.

Przykład: przepis na ciasto.

Dane wejściowe: 200g mąki, 100g cukru, 100g masła, 2 jajka.

Dane wyjściowe: ciasto.

Lista kroków:

1. Rozgrzej piekarnik do 180 stopni.
2. Wrzuć do miski mąkę, cukier, masło i jajka.
3. Miksuj składniki przez 5 min.
4. Formę do pieczenia wysmaruj masłem i oprósz bułką tartą.
5. Przełóż masę z miski do formy.
6. Formę umieść w piekarniku.
7. Odczekaj 40 min.
8. Wyjmij ciasto i wyłącz piekarnik.

Cechy algorytmów

- poprawność - algorytm daje oczekiwane wyniki,
- jednoznaczność - zawsze daje te same wyniki przy takich samych danych wejściowych,
- skończoność - wykonuje się w skończonej liczbie kroków,
- sprawność - czasowa (szybkość działania) i pamięciowa

Sposoby zapisu algorytmów

- 1 Lista kroków
- 2 Pseudokod - metoda pośrednia między 1. i 4.
- 3 Schemat blokowy
- 4 Język programowania

Przykład: średnia arytmetyczna dwóch liczb rzeczywistych

Sposób 1: lista kroków

- 1 pobierz pierwszą liczbę
- 2 pobierz drugą liczbę
- 3 dodaj liczby do siebie
- 4 wynik dodawania podziel przez dwa
- 5 wypisz otrzymaną wartość
- 6 zakończ

Sposób 2: pseudokod

- 1 pobierz x
- 2 pobierz y
- 3 wykonaj działanie $(x + y)/2$
- 4 wypisz wynik
- 5 zakończ

Sposób 3: schemat blokowy

Sposób 4a: program w języku C++

```
#include <iostream>
int main()
{
    float x,y;
    std::cin >> x;
    std::cin >> y;
    std::cout << "Średnia arytmetyczna wynosi: " << (x+y)/2;
}
```


Sposób 4b: program w języku Pascal

```
Program Srednia;  
Var x,y: Real;  
Begin  
  Readln(x);  
  Readln(y);  
  Writeln('Średnia arytmetyczna wynosi: ' (x+y)/2);  
End.
```

Sposób 4c: program w języku Python

```
x = float(input())  
y = float(input())  
print("Średnia arytmetyczna wynosi: ", (x+y)/2)
```

Pierwszy program w C++

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Witaj w świecie C++";
    return 0;
}
```

- Każdy program musi zawierać specjalną funkcję o nazwie **main**. Od tej funkcji zaczyna się wykonywanie programu.
- Definicja funkcji składa się z wiersza **int main()**, będącego *nagłówkiem funkcji* oraz ujętej w nawiasy klamrowe { } *treści funkcji*.
- Nagłówek to krótki opis interfejsu funkcji.
- Treść funkcji zawiera instrukcje wykonywane przez tę funkcję.
- Instrukcje to wyrażenia C++ zakończone średnikami.
- Ostatnia instrukcja funkcji, **return** (nazywana *instrukcją powrotu*), kończy wykonywanie programu.

Pierwszy program w C++

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Witaj w świecie C++";
    return 0;
}
```

- Zmienna **cout** (console out) pozwala wyświetlić komunikat w konsoli.
- Zapis << oznacza wysłanie komunikatu ujętego w cudzysłowach (Witaj w świecie C++) do **cout**.
- Przed użyciem **cout** należy dołączyć do programu plik **iostream**, zawierający m.in. definicję **cout**. Służy do tego dyrektywa **#include**.
- Za pomocą dyrektywy **using** do programu dodajemy standardową (std) przestrzeń nazw. Dzięki temu piszemy krótko **cout** zamiast **std::cout**.

Instrukcja deklaracji i instrukcja przypisania

```
#include <iostream>
using namespace std;
int main()
{
    int i;
    i = 5;
    cout << "Zmienna i";
    cout << "o wartości ";
    cout << i;
    cout << endl;
    i = i + 1;
    cout << "Teraz zmienna i ma wartość " << i;
    return 0;
}
```

Wynik działania programu:

```
Zmienna i o wartości 5
Teraz zmienna i ma wartość 6
```

Instrukcja deklaracji i zmienne

- Aby zapisać w komputerze jakąkolwiek informację, trzeba wskazać położenie tej danej oraz określić ilość zajmowanego przez nią miejsca.
- Instrukcja `int i;` mówi, że program potrzebuje tyle pamięci, ile zajmuje liczba całkowita oznaczana nazwą `int` (ang. *integer*).
- Już sam kompilator zajmie się szczegółami alokacji i oznaczenia odpowiedniego miejsca w pamięci.
- Deklaracja nie tylko określa typ, ale deklaruje nazwę tak, że napotykając dalej na nazwę `i`, program będzie odnosił się zawsze do tego samego miejsca w pamięci.
- `i` to *zmienna*.
- W C++ wszystkie zmienne trzeba deklarować. Jeśli jakaś deklaracja zostanie pominięta, kompilator zgłosi błąd.
- Deklaracja `int i;` to *deklaracja definiująca* lub krócej *definicja*.

Instrukcja przypisania

- Instrukcja przypisania powoduje przypisanie wartości pewnemu miejscu w pamięci.
- Instrukcja `i = 5;` powoduje przypisanie liczby 5 miejscu oznaczonemu jako zmienna `i`.
- Symbol `=` to *operator przypisania*.
- Przypisanie wykonywane jest od strony prawej do lewej.
- W instrukcji `i = i + 1` kompilator najpierw wykona działanie znajdujące się po prawej stronie operatora przypisania: `i + 1`, otrzymując wartość 6. Następnie operator przypisania wstawi tę nową wartość do zmiennej `i`.

Wykonując polecenie `cout << i;`, program nie wyświetli znaku `i`, ale pokaże liczbę z tej zmiennej - czyli 5.

Instrukcja *cin*

```
#include <iostream>
using namespace std;
int main()
{
    int i;
    cout << "Podaj jakąś liczbę ";
    cin >> i;
    cout << "Nadałeś zmiennej wartość: " << i;
    return 0;
}
```

Wynik działania programu w przypadku, gdy użytkownik wpisał z klawiatury 2:

```
Podaj jakąś liczbę
Nadałeś zmiennej wartość: 2
```