

Programownie I

Wykład 1

dr inż. Adam Zielonka

Instytut Matematyki,
Politechnika Śląska

Gliwice 05.10.2018

Karta przedmiotu

Cel przedmiotu

- Poznanie podstawowych konstrukcji programistycznych.
- Zapoznanie się z metodologią tworzenia programów w języku C++ oraz metodami weryfikacji poprawności tych programów.
- Nabycie umiejętności korzystania z narzędzi służących do pisania programów.
- Nabycie umiejętności pisania jednomodułowych programów konsolowych.
- Nabycie umiejętności doboru poznanych konstrukcji języka na potrzeby rozwiązywania problemów programistycznych.

Karta przedmiotu – efekty kształcenia

Efekt 1

Potrafi wykorzystać narzędzia do pisania i testowania programów.

Efekt 2

Zna i potrafi wykorzystać metody programowania operacji wejścia-wyjścia w języku C++.

Efekt 3

Rozumie zasady programowania proceduralnego. Zna i potrafi wykorzystać pojęcie funkcji w języku C++.

Karta przedmiotu – efekty kształcenia

Efekt 4

Zna i potrafi wykorzystać podstawowe konstrukcje programistyczne.

Efekt 5

Potrafi przy użyciu języka C++ rozwiązać określone problemy obliczeniowe.

Karta przedmiotu – zasady zaliczenia

- Dwa kolokwia praktyczne 2x20pkt.
- Kolokwium w formie testu na wykładzie 20 pkt.
- Projekt zaliczeniowy 25 pkt.
- Aktywność na zajęciach 15 pkt. (w tym 10 pkt. za 2 zadania z list.)

Karta przedmiotu

O prowadzącym

dr inż. Adam Zielonka (adam.zielonka@polsl.pl, RMS 509I) jest absolwentem Wydziału Matematyczno-Fizycznego Politechniki Śląskiej (aktualnie Wydział Matematyki Stosowanej), moje zainteresowania naukowe dotyczą szerokiego zastosowania matematyki i informatyki w technice a w szczególności zastosowania algorytmów heurystycznych w rozwiązywaniu odwrotnych zagadnień procesu krzepnięcia metali dwuskładnikowych. W pracy dydaktycznej cyklicznie prowadzę wykłady z: programowania obiektowego oraz programowania urządzeń mobilnych dla systemu Android. Jestem opiekunem koła naukowego Referencja.

Po co programowanie?

Język C++

Język C++

Stworzony w początku lat osiemdziesiątych poprzedniego wieku przez Bjarne Stroustrup – duńskiego programistę obecnie profesora <http://www.stroustrup.com>

Język C++

Stworzony w początku lat osiemdziesiątych poprzedniego wieku przez Bjarne Stroustrup – duńskiego programistę obecnie profesora <http://www.stroustrup.com>
Mechanizm obiektowości w języku C++.

Język C++

Stworzony w początku lat osiemdziesiątych poprzedniego wieku przez Bjarne Stroustrup – duńskiego programistę obecnie profesora <http://www.stroustrup.com>

Mechanizm obiektowości w języku C++.

Umożliwia stosowanie paradygmatów programistycznych języka: proceduralnego, obiektowego i generycznego.

Język C++

Stworzony w początku lat osiemdziesiątych poprzedniego wieku przez Bjarne Stroustrup – duńskiego programistę obecnie profesora <http://www.stroustrup.com>

Mechanizm obiektowości w języku C++.

Umożliwia stosowanie paradygmatów programistycznych języka: proceduralnego, obiektowego i generycznego.

Język C++ jest standaryzowany przez ISO – aktualnie C++17.

Pierwszy program

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Programowanie_1\n";
}
```

Komentarze

Komentarz wieloliniowy

```
/* Pierwsza linia  
kolejna linia  
brak zagnieżdżenia  
...  
ostatnia linia */
```

Komentarz do końca linii

```
// To jest cała linia komentarzu  
cout<<"Cześć"; //To już jest komentarz
```

Typy danych

całkowitoliczbowe

- `int` – zmienna tego typu reprezentuje wartość całkowitą ze znakiem (4 bajty)
- `unsigned int` – wartość całkowita bez znaku (4 bajty)
- `short int` *krócej short* – (2 bajty)
- `long int` *krócej long* – (4 bajtów)
- `long long int` *krócej long long* – (8 bajtów)

znakowe

- `char` – pojedyncza znak ASCII (1 bajty)

zmiennoprzecinkowe

- float – pojedyncza precyzja (4 bajty)
- double – podwójna precyzja (8 bajtów)

logiczne

- bool – wartości true false

Sposoby deklarowania zmiennych

Nazwa zmiennej – dowolny ciąg liter języka angielskiego, liczb i znaków podkreślenia, przy czym nie może zaczynać się od cyfry. Rozróżniane są wielkie i małe litery. Nie można używać słów kluczowych.

```
int ilosc=12;  
int wiek=19.3;  
int wynik {19};  
int i,j,k;
```

Przykład

```
#include <iostream>
using namespace std;
int main()
{
    int wiek=19.5;
    cout << "Wiek_□" << wiek << '\n';
    wiek=20;
    cout << "Wiek_□" << wiek;
    system("pause");
}
```

Przykład

```
#include <iostream>
using namespace std;
int main()
{
    int wiek=19.5;
    cout << "Wiek_□" << wiek << '\n';
    wiek=20;
    cout << "Wiek_□" << wiek;
    system("pause");
}
```

Wynik

```
Wiek 19
Wiek 20Press any key to continue...
```

Kilka wskazówek

- Nazwa (identyfikator) zmiennej powinna odzwierciedlać sens istnienia zmiennej w programie.

Kilka wskazówek

- Nazwa (identyfikator) zmiennej powinna odzwierciedlać sens istnienia zmiennej w programie.
- Zmienne zawsze powinny być zainicjowane.

Kilka wskazówek

- Nazwa (identyfikator) zmiennej powinna odzwierciedlać sens istnienia zmiennej w programie.
- Zmienne zawsze powinny być zainicjowane.
- Dbaj o czystość kodu (wcięcia, komentarze.)

Kilka wskazówek

- Nazwa (identyfikator) zmiennej powinna odzwierciedlać sens istnienia zmiennej w programie.
- Zmienne zawsze powinny być zainicjowane.
- Dbaj o czystość kodu (wcięcia, komentarze.)
- Programuj tak aby w przyszłości można było program dalej rozwijać.

Operatory

Operatory arytmetyczne

- + dodawanie $a+b$
- - odejmowanie $a-b$
- * mnożenie $a*b$
- / dzielenie a/b
- % reszta z dzielenia $a\%b$

Operatory

Operatory arytmetyczne

- + dodawanie $a+b$
- - odejmowanie $a-b$
- * mnożenie $a*b$
- / dzielenie a/b
- % reszta z dzielenia $a\%b$

Operator przypisania

`a=a+b;`

Operatory

Operatory arytmetyczne

- + dodawanie $a+b$
- - odejmowanie $a-b$
- * mnożenie $a*b$
- / dzielenie a/b
- % reszta z dzielenia $a\%b$

Operator przypisania

$a=a+b$;

Złożone operatory przypisania

$+=$, $-=$, $*=$, $/=$, $\%=$

$a+=3$; równoważne $a=a+3$;