Języki programowania
Nowoczesne techniki programowania

Wykład 1

Witold Dyrka
witold.dyrka@pwr.wroc.pl

5/10/2012

Prawa autorskie

Slajdy do wykładu powstały
w oparciu o slajdy Bjarne Stroustrupa
do kursu Foundations of Engineering II (C++)
prowadzonego w Texas A&M University

http://www.stroustrup.com/Programming

Cele przedmiotu

- Uzyskanie podstawowej wiedzy z zakresu tworzenia **algorytmów i struktur danych** we współczesnym języku programowania

- Nabycie podstawowych umiejętności w zakresie **projektowania i implementacji** prostych aplikacji na przykładzie języka **C++**

Program wykładów

1. Pierwszy program 5/10
2. Obiekty, typy i wartości. Wykonywanie obliczeń. 12/10
3. Błędy. Pisanie programu 19/10
4. Kończenie programu 26/10
5. Techniki pisania funkcji i klas 9/11
6. Strumienie wejścia/wyjścia 16/11
7. Wskaźniki i tablice 23/11

Program laboratorium

- Tworzenie algorytmów
- Pisanie programów w języku C++

Materiały

Literatura

- Bjarne Stroustrup. *Programowanie: Teoria i praktyka z wykorzystaniem C++*. Helion (2010)
- Jerzy Grębosz. *Symfonia C++ standard.* Edition 2000 (2008)
- Dowolny podręcznik języka C++ w standardzie ISO 98

Środowisko programistyczne

- Microsoft Visual C++ (rekomendowane)
- Dowolne środowisko korzystające z GCC

Warunki zaliczenia

- Kolokwium zaliczeniowe
  - piątek, 30/11/2012 (w terminie wykładu)
  - poprawka: w tyg. 3-7/12/2012 (poza terminem wykładu)

- Na zaliczenie >50% pkt.

Zasady

- Nie toleruję plagiatów
  - dot. programów na laboratorium
- Nie toleruję ściągania
  - dot. kolokwium i sprawdzianów na laboratorium
- Zachęcam do korzystania z konsultacji
  - ze mną: PN 10-12, pok. 118/D-1
  - z innymi Prowadzącymi
  - pomiędzy sobą

Dlaczego programowanie?

Programowanie w medycynie

Na przykład:

- Tomografia
  oprogramowanie skanera, przesyłanie sygnałów, przetwarzanie obrazu
- Chirurgia wspomagana robotami
  wizualizacja, precyzyjne sterowanie narzędziami, sterowanie na odległość
- Inne: automatyczne badania (np. krwii), urządzenia ultradźwiękowe, komputery kieszonkowe i tzw. *mobile health care*
- Natychmiastowy dostęp do danych pacjenta

- Projekty medyczne realizowane przez znajomą, niewielką firmę informatyczną z Wrocławia:
  - obsługa magazynu lekarstw i apteki szpitala
  - optymalizacja produkcji granulatu w dużej firmie farmaceutycznej

ResolutionMD

Demaerschalk et al. *Stroke* 2012: http://stroke.ahajournals.org/content/early/2012/09/11/STROKEAHA.112.669325.full.pdf

Czym jest programowanie?

- Powiedzeniem **bardzo** szybkiemu kolesiowi co ma **dokładnie** robić
- Planem rozwiązywania problemu przez komputer
- Określeniem kolejności wykonywanych działań
- Z tym, że:
  - współczesne programy zawierają miliony linii kodu
  - i koncentrują się na manipulacji danymi
- Może więc:

  Określeniem struktury i zachowania programu oraz sprawdzeniem czy wykonuje on zadanie prawidłowo i dostatecznie sprawnie

Czy programowanie jest trudne?

- Zasadniczo jest łatwe:
  - Napisz maszynie co ma robić
- Dlaczego jest więc trudne? ;-)
  - Chcemy by komputer robił złożone rzeczy, a on jest tylko maszyną
  - Rzeczywistość jest bardziej skomplikowana niż nam się wydaje, a my nie zawsze zdajemy sobie sprawę z implikacji tego czego chcemy
  - „Programowanie jest rozumieniem”, tego co chcemy zautomatyzować
  - Programowanie łączy teorię i praktykę

Pierwszy program

```
#include <iostream>                    // dołącza plik nagłówkowy biblioteki przydatnych
                                       // narzędzi

int main()                             // main() jest początkiem każdego programu C++

{
    std::cout << "Hello, world!\n";    // wyświetla 13 znaków: Hello, world!
                                       // i przechodzi do nowej linii
    return 0;                          // zwraca wartość oznaczającą sukces

}


// Cudzysłów typu " " ogranicza literał łańcuchowy

// \n oznacza początek nowej linii

// Średnik ; kończy każdą instrukcję

// Nawiasy klamrowe { … } grupują instrukcje w blok

// main() jest funkcją, która nie pobiera argumentów i zwraca wartość całkowitą (int)
    oznaczającą sukces (0) lub błąd (!0)
```

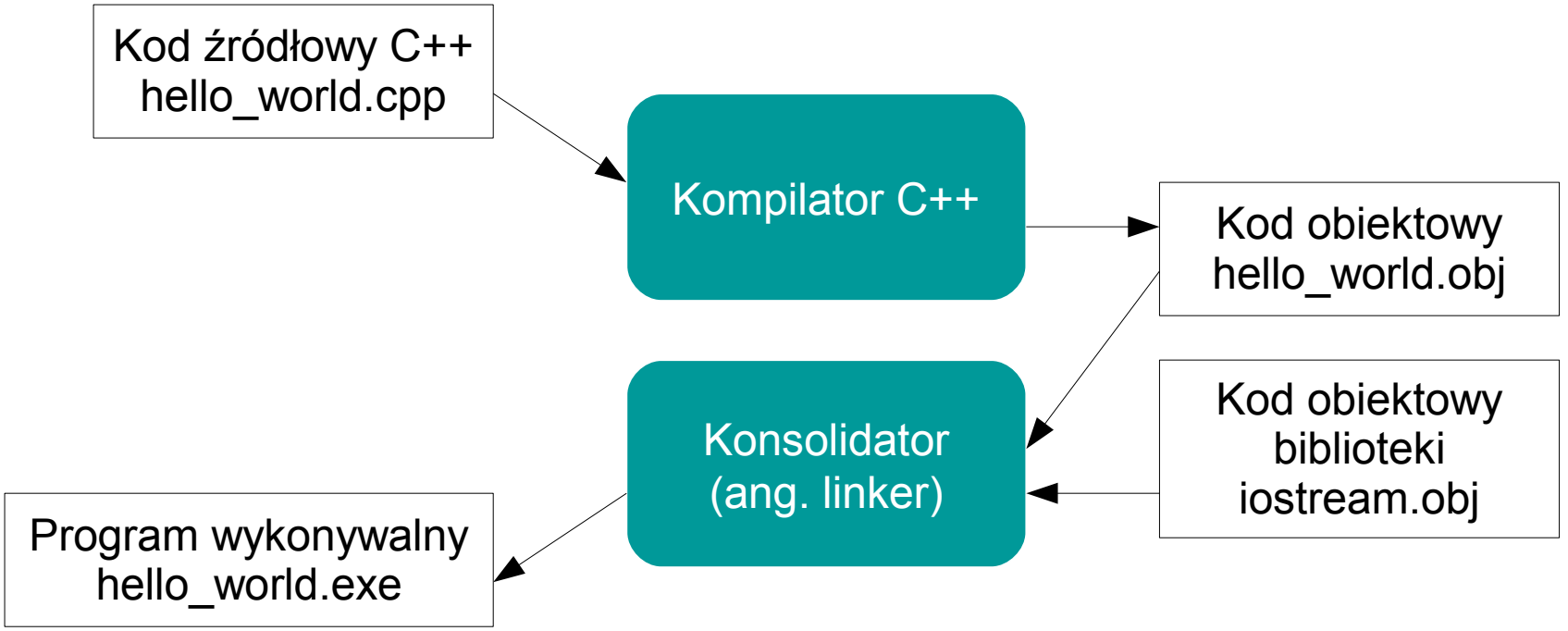
Pierwszy program (2)

```
/* Pierwszy program - cele
     – poznanie:
          kompilatora,
          środowiska deweloperskiego (IDE)
          środowiska uruchomieniowego

     – sprawdzenie reakcji na błędy
          zapomniany nagłówek,
          zapomniany cudzysłów kończący łańcuch znakowy
          pomyłka w nazwie (np. retrun)
          zapomniany średnik lub nawias klamrowy
*/

#include <iostream>                  // dołącza plik nagłówkowy biblioteki narzędzi

int main()                            // main() jest początkiem każdego programu C++

{
    cout << "Hello, world!\n";       // wyświetla 13 znaków: Hello, world!
                                     // i przechodzi do nowej linii
    system("pause");                 // czeka na wciśnięcie klawisza
    return 0;                        // zwraca wartość oznaczającą sukces

}
```

Liczy się styl

- Większa część kodu to jakby gotowy formularz
- Jedyna instrukcja, która „coś” robi to:
  cout << "Hello, world!\n"
- I taki podział jest zupełnie normalny:-)
  - bez komentarzy, dołączania bibliotek itp.
    kod nie będzie prosty, zrozumiały, godny zaufania i efektywny
- Nie wystarczy by jakoś działało
  - kod musi być elegancki i poprawny

Kompilacja i konsolidacja



- Kompilator
  - sprawdza kod napisany w języku programowania
  - generuje kod obiektowy, specyficzny dla architektury komputera i systemu operacyjnego
- Konsolidator dołącza kod obiektowy wykorzystywanych bibliotek i porządkuje przestrzeń programu

Dziś najważniejsze było to, że...

- Warto, aby inżynier biomedyczny nauczył się dobrze programować
- W programowaniu liczy się styl

A za tydzień...

- Obiekty, typy, wartości
- Wykonywanie obliczeń