

Lekcja 1

Pierwszy program w C++

wstęp – środowisko - specyfikacja – etapy tworzenia – struktura – stałe i zmienne – przypisanie - konsola – obliczenia arytmetyczne – formatowanie

Wstęp

Język C (i jego pochodne) jest obecnie standardem kodowania. Powstał na początku lat 70-tych i był przeznaczony głównie do projektowania systemów operacyjnych i procesów związanych bezpośrednio z układami komputerów. Twórcy postawili na prostotę i minimalizm, co spowodowało, że stosowanie C w edukacji jest dość problematyczne.

Próba sprowadzenia składni do absolutnego minimum sprawiła, że kod może stać się zupełnie nieczytelny. Wszystkie funkcje znajdują się w bibliotekach, co nie jest złym pomysłem, ale brak standaryzacji sprawia, że różne wersje tych samych funkcji nakładają się na siebie prowadząc do tzw. „przeciążenia”. Różne wersje kompilatorów i środowisk też nie zachowują się identycznie. Kolejne wersje „cywilizują” C doprowadzając do możliwości, które inne języki miały wbudowane od samego początku.

Każdy początkujący programista wyobraża sobie, że po kilku minutach napisze grę komputerową, którą zachwyci cały świat. Rzeczywistość jest brutalna – prawie w każdym języku programowania naukę rozpoczyna się od matematycznych obliczeń i logicznego myślenia. Skopiowanie gotowego, wymyślonego przez kogoś kodu jest pożądane, ale tylko wtedy, gdy rozumie się, co kryje się pod tymi „dziwnymi krzaczkami na białym tle”.

Środowisko programistyczne – darmowe, najpopularniejsze

- CodeBlocs
- Microsoft Visual C++ 2010 Express (wersja darmowa)

Specyfikacja zadania (problemu)

Programy piszemy po to, by rozwiązać jakieś problemy. W programowaniu podstawowymi rzeczami, których będziemy musieli się nauczyć są porządek i planowanie. I pomimo tego, że C (przez swój minimalizm) nie ułatwia tego, to nic nie stoi na przeszkodzie, aby pisać kod zrozumiały, zwłaszcza jeśli będziemy musieli go analizować po jakimś czasie.

Opis sposobu rozwiązania jakiegoś problemu nazywamy w informatyce **specyfikacją**, w skład której wchodzi: **opis sposobu rozwiązania, dane wejściowe, uzyskiwane wyniki oraz zależności** pomiędzy danymi, a wynikami.

Tworzenie programu - etapy

specyfikacja:	planujemy, jak napisać program
implementacja:	zapisanie algorytmu w postaci koda w konkretnym języku programowania
kompilacja:	zamiana kodu źródłowego na język maszynowy
uruchomienie:	wykonanie programu przez system komputerowy
debugowanie:	sprawdzanie poprawności działania, wyszukiwanie błędów

Struktura programu

Standardowy schemat wygenerowany w nowym projekcie różni się nieco od przedstawionego poniżej. Oczywiście, to tylko schemat, ale jak to zwykle bywa mamy mnóstwo innych możliwości i wyjątków. Co wchodzi w skład typowego szkieletu programu:

deklaracja bibliotek:	<code>#include <iostream></code>
deklaracja stałych zmiennych:	<code>int liczba; const float PI=3.14;</code>
deklaracja funkcji:	<code>int suma(int a, int b){return a+b;}</code>
część główna:	<code>int main {cout << suma(1,2); return 0;}</code>

Praca w środowisku CodeBlocs (MS VC++)

Każdy projekt w C++ składa się z kilkunastu różnego rodzaju plików, czasem w kilku różnych folderach, dlatego bardzo ważne jest aby dla każdego nowego projektu – nawet najmniejszego – tworzyć osobny folder.

Naukę w C++ rozpoczynamy od najprostszych poleceń, dlatego też tworzymy w tzw. konsoli, która zupełnie nie nadaje się do zabawy z grafiką, ale zupełnie dobrze można w niej obliczać i przetwarzać teksty – i po prostu nauczyć się programować i myśleć algorytmicznie.

Program minimum rozbudowany możliwość pisania „po polsku” i podstawowe biblioteki

```
//#include <stdafx.h>
#include <stdlib.h>
#include <iostream>
#include <iomanip>
#include <cmath>

using namespace std;

int main(){
    setlocale(LC_ALL, "");
    //instrukcje
    system("pause");
}
```

stdafx.h - biblioteka wymagana tylko w środowisku VC++

stdlib.h - zawiera m.in. polecenie **system("pause")**

iostream - **cout**, **cin** i **endl**

cstdio - **printf**, **scanf**, **getchar**

iomanip - **fixed**, **setprecision** i **setw**

cmath - **pow** i **sqrt**

- przestrzeń nazw - **namespace**, aby łatwiej było „wyłuskać”

funkcje z bibliotek, nie trzeba używać dziwnych operatorów „::”

- program główny - **main**

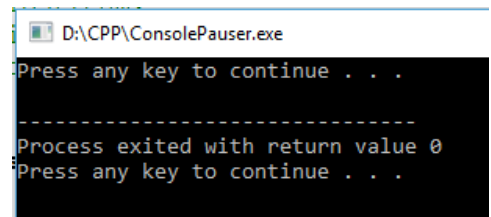
- polskie ogonki w tekstach - **setlocale**

- komentarze wpisujemy po znakach „//”

- tutaj wpisujemy instrukcje naszego programu

Kompilowanie i uruchamianie programu

Programy napisane w C++ (podobnie jak w innych językach wysokiego poziomu, np. Pascal, Java) muszą być przetłumaczone na język zrozumiały dla procesora, tzw. język maszynowy. Odbywa się to podczas kompilacji, i przy okazji sprawdzane są wszystkie błędy. Wynikiem działania kompilatora jest plik wykonywalny (aplikacja) z rozszerzeniem exe. Taki wykonywalny plik można już uruchomić, albo w środowisku, w którym pracujemy, albo też klikając podwójnie na ikonę pliku.



- **kompilowanie programu** (sprawdzenie poprawności) **F9** w CB **F7** w VC

Jeśli program nie ma błędów otrzymujemy komunikat końcowy o poprawności kompilacji. Jeśli w programie występują błędy, zostanie wygenerowany komunikat (czasem bardzo obszerny) z opisem i **numerem wiersza z błędem**

- **uruchomienie programu** (należy wcześniej skompilować) **F10** w CB **F5** w NC

Pojawia się ekran konsoli, na którym pojawiają się wyniki obliczeń.

Obsługa konsoli - PRINTF i SCANF

W języku C mamy dwa mechanizmy do obsługi konsoli: **printf** i **scanf** (z mnóstwem opcji i parametrów) oraz wygodniejsze **cout** i **cin**. W dalszej części będziemy posługiwać się wyłącznie poleceniami **cout** i **cin** z powodu bardziej „edukacyjnego” wyglądu.

- **printf()** - wypisuje na ekranie teksty lub liczby
- **scanf()** - wczytujemy z klawiatury teksty lub liczby
- formatowanie polega na wpisaniu odpowiednich parametrów do **printf** lub **scanf**, wszystkie parametry zapisane są w „” i zaczynają się od znaku (%)
- przejście do nowego wiersza „\n”

```
#include <cstdio>
...=
int liczba;
printf("wczytaj całkowitą:\n");
scanf("%i",liczba);
printf("wczytano: ");
printf("%i\n",liczba);
const float PI=3.1415926536;
printf("%20.2f\n",PI);
```

Zadanie – okrąg

Wczytaj z klawiatury promień okręgu. Wylicz jego obwód oraz pole powierzchni.

```
// biblioteki
#include "stdlib.h"
#include <iostream>
#include <iomanip>
#include <cmath>
// przestrzeń nazw
using namespace std;
int main(){
    // polskie znaki na ekranie
    setlocale(LC_ALL, "");
    // deklarowanie stałej PI typu rzeczywistego
    const double PI = 3.1415;
    // deklarowanie trzech zmiennych typu rzeczywistego
    // nie zawierają żadnych konkretnych wartości
    double R,P,O;
    // komunikat na ekranie i przejście do nowego wiersza
    cout << "POLE i POWIERZCHNIA koła" << endl;
    // komunikat na ekranie – kursor na końcu tekstu
    cout << "Podaj promień: ";
    // do zmiennej R wczytujemy z klawiatury liczbę
    cin >> R;
    // W zmiennej P zostanie obliczone pole powierzchni
    // można też P=PI*pow(R,2);
    P = PI * R * R;
    // w zmiennej O obwód okręgu
    O = 2 * PI * R;
    // komunikaty z wynikami obliczeń
    cout << "POLE=" << P << endl;
    cout << "OBWÓD=" << O << endl;
    // komunikat i oczekiwanie na klawisz ENTER
    system("pause");
}
```

Program został specjalnie rozbudowany o obszernie komentarze (zielone) – w przyszłości można je zupełnie pomijać.

Wpisuj po jednym wierszu i kompiluj – będziesz mógł od razu wyszukać i usunąć błędy. Szczególną uwagę zwracaj na cudzysłowy przy tekstach, średniki na końcach instrukcji.

Stale i zmienne

Stale i zmienne, to „pojemniki” (miejsca w pamięci komputera), w których przechowywane są liczby, teksty i inne dane, w trakcie działania programu. Jak sama nazwa wskazuje, stałe nie mogą się zmieniać, a do zmiennych możemy wstawiać różne wartości – jednak zawsze takiego samego typu. Stałe i zmienne muszą być zawsze

zadeklarowane przed ich użyciem (choć nie wszystkie języki programowania wymagają takiej deklaracji).

Instrukcja przypisania

Zapis $P = PI * R * R$; oznacza, że:

- obliczana jest wartość wyrażenia po **prawej stronie** $PI * R * R$, pod symbole PI i R podstawiane są odpowiednie stałe zadeklarowane w programie lub wczytane z klawiatury
- obliczona wartość wstawiana jest (przypisywana) do zmiennej po **lewej stronie**

Podstawowe operacje arytmetyczne

+	dodawanie	$2 + 2 = 4$	
-	odejmowanie	$4 - 2 = 2$	
*	mnożenie	$2 * 2 = 4$	
/	dzielenie	$6 / 4 = 1$	składniki są liczbami całkowitymi
		$6.0 / 4.0 = 1.5$	składniki rzeczywiste
%	dzielenie modulo	$6 \% 5 = 1$	reszta z dzielenia całkowitego

Podstawowe typy zmiennych i stałych (istnieją jeszcze inne)

- **int** liczby całkowite, np. -123 (4 bajty)
- **float** liczby rzeczywiste, np. 3.141592 (4 bajty)
- **double** liczby rzeczywiste, np. 3.141592 (8 bajtów)
- **char** znaki, np. „a” lub „0” (1 bajt)
- **string** napisy, np. „Ala ma kota” (tyle bajtów ile znaków)
- **bool** zmienne logiczne (prawda lub fałsz) (1 bajt)

UWAGA INT-DOUBLE

`cout << 1+1/3` wyświetli 1, ale `cout << 1.0+1.0/3.0` wyświetli 1.333333

deklarujemy **int J, D;**
wprowadzamy z klawiatury 1 i 3, to `cout << J+J/T` wyświetli 1
deklarujemy **double J, D;**
wprowadzamy z klawiatury 1 i 3, `cout << J+J/T` wyświetli 1.33333

Formatowanie wyników obliczeń

W języku C dość nieprecyzyjnie (w dodatku kilkoma sposobami) opisano możliwości formatowania wypisywanych wyników na ekranie monitora. Aby ni komplikować, będziemy posługiwać się biblioteką **iomanip** i instrukcjami **fixed**, **setprecision** i **setw**. W różnych opracowaniach książkowych i internetowych możesz spotkać się z wieloma innymi – pomocy szukaj w internecie!

Zadanie - PI

Zadeklaruj stałą PI o wartości 4,1415629536 i wypisz ją na ekranie z dwoma miejscami po przecinku, przesuniętą o 10 pól od poprzedniego napisu (lewego brzegu ekranu).

```
PI=      3.14
```

```
#include <iomanip>
...
const double PI = 3.1415926536;
cout << fixed << setprecision(2);
cout << "PI=";
cout << setw(10) << PI << endl;
```

Ogólne uwagi do pisanych programów

- nie możesz deklarować dwa razy takich samych zmiennych i stałych w programie
- działania i nawiasy, jak w matematyce (problem z dzieleniem $4/3*PI$ oraz $PI*4/3$)
- podczas obliczeń wykorzystaj bibliotekę **cmath** i funkcje **pow** i **sqrt**

```
#include <cmath>
```

```
wynik = pow(podstawa, potega );
```

```
wynik = sqrt(liczba);
```

- dane do funkcji (np. pow i sqrt) muszą być odpowiedniego typu (np. double), nie mieszaj różnych typów w jednej funkcji (C tego nie lubi), w przyszłości nauczysz się konwertować typy

```
POLE i OBWÓD kwadratu
BOK=      2.1
POLE=     4.41
OBWÓD=    8.40
```

```
POLE i OBWÓD prostokąta
BOK A=    2.1
BOK B=    4
POLE=     8.4
OBWÓD=   12.2
```

Zadania – wyniki na ekranie, jak na załączonych obrazkach

- 1) Oblicz pole powierzchni i obwód kwadratu dla danego boku.
- 2) Oblicz pole powierzchni i obwód prostokąta dla zadanych boków.

Walec $V=PI*R^2*H$ $P=2*PI*R^2+2*PI*R*H$

- 3) Prostokątny arkusz blachy o wymiarach 20 cm x 40 cm można zwinąć w dwojaki sposób, otrzymując powierzchnię boczną walca. W którym przypadku walec będzie miał większą objętość? Zmień zadanie wpisując wymiary blachy z klawiatury.

```
OBJĘTOŚĆ WALCA
BLACHA 20x40 V=1273.24
BLACHA 40x20 V=2546.48
```

Stożek $V=1/3*PI*R^2*H$ $P=PI*R^2+PI*R*L$

- 4) Trójkąt prostokątny o przyprostokątnych 3 cm i 4 cm obraca się raz dookoła przyprostokątnej 4 cm, a drugi raz dookoła przyprostokątnej 3 cm. Czy objętości powstałych stożków będą równe? Czy pola powierzchni bocznych będą równe? Zmień zadanie wpisując wymiary trójkąta z klawiatury.

```
STOŻEK
trójkąt 3x4 V=50.265 P=24.936
trójkąt 4x3 V=37.699 P=33.247
```

Kula $V=4/3*PI*R^3$ $P=4*PI*R^2$

- 5) Przekrój osiowy kuli jest kołem, którego obwód wynosi 18π cm. Oblicz pole powierzchni i objętość tej kuli. Zmień zadanie wpisując obwód z klawiatury.

```
KULA
P=1017.88
V=3053.63
```

- 6) Kocioł o średnicy 6 dm i wysokości 4 dm cały wypełniony jest zupą. Chochla ma kształt półkuli o promieniu 6 cm. Harcerze dostają po 2 chochle zupy. Czy zupy wystarczy dla 100 harcerzy?

```
KOCIOŁ i CHOCHLA
Vko=113097.34 Vch=904.78
Ilu harcerzy=125
```

- 7) Do garnka o promieniu 9 cm, wypełnionego wodą wpadła brzoskwinia o średnicy 6 cm. O ile cm podniesie się poziom wody w garnku?

```
BRZOSKWINIA w GARNKU
Poziom wody podniesie się o 0.44 cm
```