

Wprowadzenie do języka C++

Konrad Pierzyński

konrad.pierzynski@amu.edu.pl

KONPIE1

Harmonogram

Data	Zajęcia	Data	Zajęcia	Data	Zajęcia
28.02	✓	11.04	✓	23.05	✓
7.03	✓	18.04	✓	30.05	✓
14.03	✓	25.04	✓	6.06	✓
21.03	✓	2.05	✗	13.06	✓
28.03	✓	9.05	✓		
4.04	✓	16.05	✓		

Warunki zaliczenia

Uzyskanie pozytywnej oceny z zadań według skali:
















- bdb >90%
- db+ >80%
- db >70%
- dst+ >60%
- dst >50%

Zajęcia I

Cele:

- pobrać, zainstalować, uruchomić, skonfigurować IDE
- skompilować i uruchomić dowolny program
- poznać podstawy języka

IDE: Integrated Development Environment

IDE	OS	FREE
Visual Studio		
Code::Blocks	  	
CLion	  	 
Xcode		
NetBeans	  	
Online C++ playground		

```
int main() {  
    return 0;  
}
```


C++

Python

kompilowany

interpretowany

silnie typowany

dynamicznie typowany

wieloparadygmatowy

wieloparadygmatowy

wydajny

przyjazny

Każdy program zaczyna się od `main`

```
int main() {  
    return 0;  
}
```

Funkcje

```
int f(int x) {  
    return 2*x + 3;  
}
```

```
def f(x):  
    return 2*x + 3
```

Zmienne

```
int a = 42;
```

```
a = 42
```

Operator

Operator	C++	Python
Przypisania	=	=
Inkrementacji	i++	
Inkrementacji	++i	
Dekrementacji	i--	
Dekrementacji	--i	

Operatory arytmetyczne

Operator	C++	Python
Dodawania	+	+
Odejmowania	-	-
Mnożenia	*	*
Dzielenia	\	\
Dzielenia całkowitego	\	//
Reszta z dzielenia	%	%

Operatory logiczne

Operator	C++	Python
Logiczne 'lub'		or
Logiczne 'i'	&&	and
Zaprzeczenie	!	not

Operatory relacyjne

Operator	C++	Python
Mniejszy	<	<
Większy	>	>
Równy	==	==
Różny	!=	!=

Typy

```
bool is_cpp_fun = false;  
char c = 'K';  
int a = 42;  
short b = 24500;  
long b = 2138000;  
float pi = 3.14f;  
double e = 2.72;  
---  
unsigned int a = 3138000;  
signed short b = -1002;  
---  
void
```

Wiedza -> praktyka

```
int f(int x) {  
    return 2*x + 2;  
}  
  
int main() {  
    int x = 2;  
    int y = f(x);  
    return 0;  
}
```

Biblioteka standardowa

```
#include <iostream>

int f(int x) {
    return 2*x + 2;
}

int main() {
    int x = 2;
    int y = f(x);

    std::cout << y << std::endl;
    return 0;
}
```

```
print(y)
```

```
#include <iostream>

int f(int x) {
    return 2*x + 2;
}

int main() {
    int x;

    std::cin >> x;
    int y = f(x);

    std::cout << y << std::endl;
    return 0;
}
```

```
x = input()
```

Warunki

```
...  
if( condition ) {  
    // code  
}  
...
```

```
if condition:  
    # code
```


Pętlą WHILE

```
while( condition ) {  
    // code  
}
```

```
while condition:  
    # code
```


Pętla FOR

```
for(`wyrażenie początkowe`; `warunek trwania`; `wyrażenie co krok`)
```

```
...  
for(int i = 0; i < 3; i++) {  
    // code  
}  
...
```

```
for(int i = 0; i < 3; i++) {  
    // code  
}
```

- `int i = 0;`
- `0 < 3;` -> true
- `i++;`
- `1 < 3;` -> true
- `i++;`
- `2 < 3;` -> true
- `i++;`
- `3 < 3;` -> false

Zadanie Lab I

Napisz programu w języku C++, który obliczy Body Mass Index (BMI) dla użytkownika na podstawie podanych danych o wzroście i masie ciała oraz określi niedowagę, wagę prawidłową, nadwagę, otyłość.

```
#include <iostream>

float calculate_bmi(float weight, float height) {
    float result = weight / (height * height);
    return result;
}

void decide(float bmi) {
    if( bmi < 18.5 ) {
        std::cout << "Masz niedowagę." << std::endl;
    } else if( bmi < 25 ) {
        std::cout << "Masz prawidłową wagę" << std::endl;
    } else if( bmi < 30 ) {
        std::cout << "Masz nadwagę" << std::endl;
    } else {
        std::cout << "Masz otyłość" << std::endl;
    }
}

int main() {
    float kg, m;

    std::cout << "Podaj swoją masę (w kg): ";
    std::cin >> kg;

    std::cout << "Podaj swój wzrost (w m): ";
    std::cin >> m;

    float bmi = calculate_bmi(kg, m);
    decide(bmi);

    return 0;
}
```

Zadanie Lab II

Napisz program w C++, który zasymuluje wzrost populacji królików zgodnie z regułami problemu Fibonacciego. Założenie jest takie, że każda para królików dojrzeva w ciągu jednego miesiąca i od drugiego miesiąca co miesiąc rodzi nową parę królików.

$$F(n) = F(n-2) + F(n-1)$$

```
#include <iostream>

int fibonacci(int n) {
    int last_last = 1;
    int last = 1;
    int current;

    for( int i = 3; i <= n; i++ ) {
        current = last_last + last;

        last_last = last;
        last = current
    }

    return current;
}

int main() {
    int months;

    std::cin >> months;

    int result = fibonacci(months);

    std::cout << "Po " << months << " miesiącach będzie: " << result << " królików." << std::endl;

    return 0;
}
```

Zadanie 0

Przetestować dostępne w Internecie IDE do programowania w C++ i wybrać najbardziej odpowiadające.

Zadanie I

Napisz program, który wczyta z wejścia standardowego (std::cin) pojedynczy znak oznaczający nukleotyd DNA (wielką literą, np. 'A') i wypisze jego komplementarny odpowiednik.

Przykładowe wejście:

'A'

Przykładowe wyjście:

'T'

Zadanie II

Rozszerz poprzedni program, dodając walidację wprowadzonego znaku. Jeśli użytkownik poda niepoprawny nukleotyd, program powinien wyświetlić komunikat o błędzie.

Przykładowe wejście:

'U'

Przykładowe wyjście:

"Błędny znak"

Zadanie III

Dodaj do programu pętlę, która umożliwi wielokrotne wprowadzanie nukleotydu.

Przykładowe działanie:

Użytkownik	Program
'A'	'T'
'Z'	"Błędny znak"
'C'	'G'