

- **Programowanie**

Lekcja 1

- **Wstęp**
- **Wskaźniki – rola wskaźników w programie**
- **Zarządzanie pamięcią – dynamiczna alokacja pamięci W3**
Wykorzystanie wskaźników do pracy z tablicami i do przesyłania

C++ jest językiem programowania wysokiego poziomu, obiektowo zorientowany.

Podstawową cechą programowania obiektowo zorientowanego jest możliwość definiowania własnych typów, które czerpią zasoby z typów wcześniej zdefiniowanych.

W języku C++ są to tzw. klasy, które charakteryzują się obecnością zmiennych i funkcji, które te zmienne obsługują.

Typy zmiennych

Typy całkowite		
Nazwa	Wielkość (bajty)	Zakres
short	2	$-2^{15} \div 2^{15} - 1$, czyli przedział $[-32768, 32767]$
int	4	$-2^{31} \div 2^{31} - 1$, czyli przedział $[-2147483648, 2147483647]$
long	4	$-2^{31} \div 2^{31} - 1$, czyli przedział $[-2147483648, 2147483647]$
long long	8	$-2^{63} \div 2^{63} - 1$, czyli przedział $[-9223372036854775808, 9223372036854775807]$
unsigned short	2	$0 \div 2^{16} - 1$, czyli przedział $[0, 65535]$
unsigned int	4	$0 \div 2^{32} - 1$, czyli przedział $[0, 4294967295]$
unsigned long	4	$0 \div 2^{32} - 1$, czyli przedział $[0, 4294967295]$
unsigned long long	8	$0 \div 2^{64} - 1$, czyli przedział $[0, 18446744073709551615]$

Typ rzeczywisty - przechowuje liczby zmiennoprzecinkowe. Gdy mamy zamiar w naszym programie wykorzystać ułamki, ten typ będzie najbardziej odpowiedni. Wyróżniamy następujące typy:

Nazwa	Wielkość (bajty)	Zakres
float	4	pojedyncza precyzja - dokładność 6 - 7 cyfr po przecinku
double	8	podwójna precyzja - dokładność 15 - 16 cyfr po przecinku

Rzutowanie typów

Jest to zmiana typów zmiennych, pod spodem przykłady:

```
double a = 3.34; //sposób pierwszy
int b = (int) a; //rzutowanie zmiennej rzeczywistej na typ całkowity - zmienna "b" przyjmie wartość
int c = int (a); //zauważmy, że rzutowanie wykonujemy poprzez wpisanie w nawiasie zmiennej,
char znak = 'q';
cout << znak << ": " << (int)znak; //wyświetlenie kodu ASCII znaku q
```

Tablice

```
long double fib[100000]; // przykład definiowania tablicy
```

Char

```
Char napis[20] null;
```

```
String tekst = "Aka na kota" //String jest tablicą znaków
```

Instrukcja warunkowa if

```
if (true){ cout<<"prawda"; } else { cout<<"fałsz"; }
```

Pętla for, while, do while

```
for(int i=1; i<=10; i=i+2)
{
    cout << endl << i;
}

while(strzal != liczba)
{
    [instrukcje realizujące odgadywanie liczby]
}

do
{
    [instrukcje realizujące odgadywanie liczby]
} while(strzal != liczba);
```

Dostęp do składników klas Wszystkie składowe klasy są widoczne wewnątrz klasy, ale mogą mieć różny poziom dostępności z zewnątrz. Poziom dostępności jest określany jednym ze słów kluczowych:

public - nazwy mogą być używane we wszystkich miejscach programu, gdzie widoczna jest definicja klasy,

private - nazwy mogą być używane tylko przez metody tej samej klasy, lub funkcje z daną klasą zaprzyjaźnione

protected - nazwy mogą być używane przez metody klasy i jej funkcje zaprzyjaźnione,

Klasy Definicję klasy dzieli się na tzw. sekcje - każda sekcja rozpoczyna się od jednego z w/w słów kluczowych z następującym po nim : . Sekcja rozciąga się do końca definicji klasy lub do rozpoczęcia innej sekcji.

```
class MyClass {           // The class
    public:                // Access specifier
        int myNum;         // Attribute (int variable)
        string myString;   // Attribute (string variable)
};

int main() {
    MyClass myObj;        // Create an object of MyClass

    // Access attributes and set values
    myObj.myNum = 15;
    myObj.myString = "Some text";

    // Print attribute values
    cout << myObj.myNum << "\n";
    cout << myObj.myString;
    return 0;
}
```

Garbage collection - zbieranie nieużytków, automatyczna dealokacja – jedna z metod automatycznego zarządzania dynamicznie przydzielaną pamięcią, w której za proces jej zwalniania odpowiedzialny jest nie programista, lecz programowy zarządca noszący nazwę garbage collector

C++ nie posiada wbudowanego GC, zwalnianie pamięci można wykonywać przy pomocy wskaźników

WSKAŹNIKI

Cztery możliwości jakie dają nam wskaźniki:

1. Dynamiczne rezerwowanie i zwalnianie obszarów Pamięci
2. Zwiększenie szybkości Zapisu i/ odczyty komórek w tablicy
3. Dawanie funkcjom do pracy oryginałów zmiennych z programu wywołującego
4. Możliwość współpracy z urządzeniem zewnętrznym, (np. Miernik temperatury)

Zadanie 1)

Stwórz zmienną LICZBA, zawierającą liczbę całkowitą(naturalną) 1234, następnie stwórz wskaźnik o nazwie W, i przypisz do niego adres alokacji zmiennej LICZBA, Wypisz ją na ekranie konsoli w zapisie dziesiętnym.

1B) Zmodyfikuj powyższy kod tak żeby wskaźnik W wypisał liczbę którą przechowuje.

Zadanie 2)

- A) Napisz funkcję liczącą ciąg fibonacii
- B) Skopiuj funkcję liczącą ciąg fibonacciego do sobobnej funkcji i zmodyfikuj ją, tak aby wpisywała każdy element do tablicy. Obie funkcje umieść w funkcji main().
- C) Zmodyfikuj funkcję z zadania B), tak żeby rezerwowała przestrzeń tablicy na tyle elementów ile ma mieć ciąg.

Zadanie 3)

- A) Stwórz funkcję liczącą średnią arytmetyczną w tablicy.
- B) Zmodyfikuj tą funkcję tak żeby stosowała wskaźniki.
- C) Porównaj czas obliczenia obu funkcji.