Programowanie komputerowe

Zajęcia 4

Typ logiczny

- Wartości logiczne są reprezentowane przez typ `bool`.
- Typ bool posiada tylko dwie wartości: `true` i `false`.
- Zamiast wartości logicznych można używać wartości całkowitych (`int`) lub znakowych (`char`). Wtedy wartość 0 jest interpretowana jako `false`, a każda niezerowa wartość jako `true`.
- Przykład:

```
int n=10;
while(n--)
    cout << n;
```

drukuje 9876543210

Referencje

Jeśli T jest pewnym typem, to T& jest typem referencyjnym do T. Wartością typu referencyjnego jest **zmienna** typu T.

Typu referencyjnego możemy użyć jako parametru funkcji. Wówczas nie powstaje kopia wartości, a funkcja wywoływana używa zmiennej zadeklarowanej w innej funkcji. Robimy tak jeśli:

- chcemy zmieniać zmienne z funkcji wywołującej,
- nie chcemy (lub nie możemy) tworzyć kopii istniejącej zmiennej.

Przekazanie parametru za pomocą referencji nazywamy przekazaniem **przez zmienną**, a w zwykły sposób przekazaniem **przez wartość**.

Referencje – przykład

Poniższa funkcja zamienia wartości dwóch zmiennych:

```
void zamien(int& a, int& b) {
    int c=a;
    a=b;
    b=c;
} // Przykład użycia poniżej. Co się stanie jeśli pominiemy znaczki & ?
int main() {
    int x=3, y=7;
    zamien(x,y);
    cout << "x=" << x << ", y=" << y << endl;
}
```

Tablice

- Tablice służą do przechowywania wielu zmiennych tego samego typu.
- Tablicę deklaruje się następująco:

  typ nazwa[liczba_elementów];

  Tak zadeklarowana tablica zawiera elementy
  nazwa[0], nazwa[1], nazwa[2], … , nazwa[liczba_elementów-1],
  wszystkie one są typu podanego w deklaracji.
- Poszczególne elementy tablicy zachowują się jak zwykłe zmienne.
- Można ustawić elementy tablicy w momencie deklaracji, np. tak:

```
int t[5]={3,7,-1,2,4};
```

Tablice jako parametry funkcji

- Tablice są zawsze przekazywane przez zmienną – funkcja nie działa na kopii tylko na oryginalnej tablicy.
- Nie jest przekazywany rozmiar tablicy; żeby go przekazać należy użyć dodatkowego parametru, np.

  ```
  int f(int tablica[], int rozmiar) { … }
  ```
- Innym sposobem jest przekazanie **wskaźnika** do tablicy, np. tak:

  ```
  int f(int* tablica, int rozmiar) { … }
  ```

  O wskaźnikach – później.
- Napisy są zapamiętywane jako tablice znaków (`char`).

Ćwiczenia

1. Napisać funkcję
   `void drukuj(int t[], int r)`
   która drukuje elementy tablicy t o długości r na ekranie.
2. Napisać funkcję
   `void wczytaj(int t[], int r)`
   która prosi o podanie wartości tablicy t o długości r.
3. Napisać funkcję
   `int max(int t[], int r)`
   która zwraca największy element tablicy t o długości r.
4. Napisać funkcję
   `int suma(int t[], int r)`
   która zwraca sumę elementów tablicy t o długości r.

Ćwiczenia (2)

5. Napisać funkcję
   `bool rowne(int t1[], int t2[], int r)`
   która zwraca true jeśli tablice t1 i t2 są równe (mają takie same elementy).
6. Napisać funkcję
   `bool niemalejaca(int t[], int r)`
   która zwraca true jeśli t[0]<=t[1]<= … <=t[r-1].
7. Napisać funkcję
   `int nwd(int t[], int r)`
   która zwraca największy wspólny dzielnik elementów tablicy t.
8. Napisać funkcję
   `double srednia(int t[], int n)`
   zwracającą średnią elementów tablicy.

Ćwiczenia (3)

9. Napisać funkcję
   `double mediana(double t[], int r)`
   zwracającą medianę elementów tablicy.
10. Napisać funkcję
    `double odchStd(double t[], int r)`
    zwracającą odchylenie standardowe elementów tablicy t.

11. Napisać funkcję
    `double kowariancja(double t1[], double t2[], int r)`
    zwracającą kowariancję pomiędzy elementami tablic t1 i t2.