

Algorytmy i złożoność obliczeniowa

Laboratorium 1: Powtórzenie podstaw programowania w języku Java

1. Klasy.

Cały kod musi być zawarty wewnątrz klas. Każda klasa publiczna musi znajdować się w pliku o nazwie zgodnej z nazwą klasy, z rozszerzeniem „.java”. Np. klasa MojaAplikacja:

```
public class MojaAplikacja {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
    }  
}
```

powinna zostać umieszczona w pliku MojaAplikacja.java.

2. Kompilacja i uruchomienie aplikacji.

Jeżeli korzystamy z narzędzia typu IDE (np. Eclipse, czy NetBeans) wystarczy wybrać odpowiednią opcję. Natomiast kompilacja z linii poleceń jest wykonywana za pomocą polecenia **javac**, np.:

```
javac MojaAplikacja.java
```

W wyniku kompilacji powstają pliki z rozszerzeniem class. Uruchomienie aplikacji z linii poleceń jest realizowane za pomocą polecenia **java**, np.:

```
java MojaAplikacja
```

UWAGA!

Podana nazwa klasy musi zawierać statyczną metodę **main** o sygnaturze takiej, jak w powyższym przykładzie.

3. Przykładowa aplikacja. Wczytywanie i wypisywanie na konsolę.

```
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
  
public class MojaAplikacja {  
  
    public static void main(String[] args) {  
        MojaAplikacja aplikacja = new MojaAplikacja();  
        aplikacja.hello();  
    }  
  
    private void hello() {  
  
        //obiekt sc, za pomoca ktorego mozna wygodnie czytac z klawiatury  
        Scanner sc = new Scanner(System.in);  
        //wypisanie komunikatu  
        System.out.println("Podaj swoje imie");  
        String s = sc.nextLine();  
        //konkatenacja napisow i wypisanie na ekran  
        System.out.println("Witaj "+s);  
    }  
}
```

4. Pętle:

```
while(warunek) {
    instrukcje ...
}

do {
    instrukcje ...
} while(warunek);

for(int i = 0; i < 10; i++){
    instrukcje ...
}
```

5. Tablice

Deklaracja tablicy pięcioelementowej zawierającej liczby całkowite:

```
int tab[5];
```

Odwołanie do elementów tablicy:

```
x = tab[0];
tab[1] = tab[2] + 3;
tab[4] = 7;
```

Tablice są indeksowane od 0.

6. Metody

Definicja metody:

```
modyfikatory zwracanyTyp nazwaMetody (listaParametrów) {
    instrukcje ...
}
```

Np.:

```
public void komunikat () {
    System.out.println("Uwaga!");
}

private double kwadrat(double x) {
    return x*x;
}
```

7. Zadania

- 1) Narysuj schemat blokowy i napisz program znajdujący NWD dla dwóch liczb naturalnych.
- 2) Narysuj schemat blokowy i napisz program sprawdzający czy wczytana liczba naturalna jest liczbą pierwszą.
- 3) Narysuj schemat blokowy i napisz program zamieniający liczbę w systemie dwójkowym na liczbę w systemie dziesiętnym.
- 4) Napisz program znajdujący liczby pierwsze metodą sita Eratostenesa.
- 5) Napisz program obliczający sumę cyfr dla zadanej liczby, potem sumę cyfr uzyskanej sumy itd. dotąd, aż uzyskamy liczbę jednocyfrową. Zdefiniuj metodę wczytującą liczbę, metodę obliczającą sumę cyfr podanej liczby oraz metodę wypisującą wynik.

8. Rekurencja

Przykładowa metoda wykorzystująca rekurencję obliczająca największy wspólny dzielnik:

```
private int nwd(int a, int b){
    if(a == b){
        return a;
    } else if(a>b){
        return nwd(a-b, b);
    } else {
        return nwd(b-a, a);
    }
}
```

Zadania:

- 6) Napisz program obliczający rekurencyjnie silnię wczytanej liczby naturalnej.
[$s(n)=n * s(n-1)$]
- 7) Napisz program obliczający rekurencyjnie n-ty wyraz ciągu Fibonacciego.
[$f(0)=0, f(1)=1, f(n)= f(n-1) + f(n-2)$ dla $n>1$]
- 8) Napisz program obliczający rekurencyjnie symbol Newtona:

$$\binom{n}{k} = \begin{cases} 1 & \text{gdy } k = 0 \text{ lub } k = n \\ \binom{n-1}{k-1} + \binom{n-1}{k} & \text{gdy } 0 < k < n \end{cases}$$

Na następnych zajęciach:

Tablice i pliki. Zbiory. Operacje na zbiorach.