Увод (Методе или функције)

- Функција је начин да се изврши неки задатак (eng. *Task*)
- Функција представља низ инструкција које решавају специфичан задатак

Предности

- Функције омогућавају поновно коришћење кода (eng. *Reusability* of code)
- Функције омогућавају лакше тестирање кода и лакше мењање лоших делова
    - Одвојено тестирање сваке функције, уместо целог пројекта одједном

Статичке и не статичке методе

- Статичке функције су оне које нису везане за објекат, тј постоје независно од њега
- Функције које нису статичке постоје само као део неког објекта
- Напомена: Сада нам није толико важно шта ово значи, научићемо током ООП дела

Декларација и позивање функција

- Декларација:
    - <Спецификатор приступа> <Повратна вредност> <Назив> (<Листа параметара>) { <Тело> }

```
// Пример функције
// Из декларације закључујемо шта функција враћа - Излаз (double)
// и шта прима - Улазни параметри (int[] - тј низ целих вредности)
public static double average(int[] grades){
    int sum = 0;

    for(int i = 0; i < grades.length; i++){
        sum += grades[i];
    }

// Морамо да претворимо један од аргумената дељења у double, да бисмо избегли целобројно дељење
    return (double)sum / grades.length;
}
```

    - Спецификатор приступа: public, private, protected
    - Повратна вредност: Било који тип `int, int[], String, bool, Object, double[][]`
    - Назив: Име функције, можемо скоро било шта да ставимо, видети праксе на интернету
    - Листа параметара: Број и тип параметара може да буде произвољан (тј шта год је функцији потребно, у примеру изнад имамо један параметар типа низ целоброојних вредности)
    - Тело функције садржи све што функција ради/рачуна и сл.
    - return - Повратна вредност функције (Шта је крајњи резултат функције који желимо да користимо и ван ње?)
- Функције позивамо тако што напишемо име функције и правилно пролседимо параметре

```
int[] ocene = new int[]{5, 5, 4};

// Приметите начине на који прослеђујемо параметре
System.out.println(average(ocene));
System.out.println(average(new int[]{3,3,5}));

// У позадини, шта год проследимо функцији, она ће то користити као промењиву grades
```

    - Редослед прослеђивања параметара је битан јер функција редом узима параметре
        - нпр. `sum(int a, int b)`
            - Шта год проследимо прво биће `a`
            - Шта год проследимо друго биће `b`
            - Конкретно, за сабирање нам није важно
        - нпр. `div(double a, double b)`
            - Шта год проследимо прво биће `a`
            - Шта год проследимо друго биће `b`
            - Овде је веома важно да ли делимо нпр `10 са 2` или `2 са 10`

- Функцијски потпис (Веома важно, приметити шта све можемо да закључимо)
  - `String blabla(int x, double y, int[] z)`
    - Функција `blabla` враћа String (Од ње као резултат добијамо текст)
    - Функција `blabla` прима тачно 3 параметра
      - Први параметар је `int`
      - Други параметар је `double`
      - Трећи параметар је `int[]`, тј низ целих вредности
  - У преводу, функција очекује тачно такве параметре, и враћа тачно овај тип
  - Ово је нарочито важно када пишете ваше функције, питања која псотављате себи:
    - Шта желим да добијем од функције? - Дефинишемо повратну вредност
    - Шта функцији све треба да би урадила то што треба? - Дефинишемо параметре

```
// Пример
// Шта желим?
//  - Од функције желим да ми да низ свих оцена које су веће од неке оцене
//  - Повратна вредност је сигурно низ, тип елемената низа је int, јер су у питању оцене
// Шта функцији треба да би испунила оно изнад?
//  - Функцији треба низ оцена да би знала које су веће од неке (int[])
//  - Функцији треба оцена од које све треба да буду веће (int)

public static int[] functionName(int[] grades, int grade);

// Остало је да осмислимо functionName, ту бирамо нешто што би боље описало шта функција ради
// Нпр овде: gradesLargerThan(int[] grades, int grade);
// Вероватно може и нешто боље, отворена дискусија
```

Уобичајени обрасци задатака и разумевање шта функција враћа

1. Функција проверава - Функција враћа `boolean`
2. Функција враћа ... - Функција враћа нешто, нема исписивања
   - ... низ бројева - Враћа int[] или double[] (или слично)
   - ... сва имена - Враћа низ или неку другу колекцију (нпр List) ниски (`String[]`, `List<String>`)
   - ... колико има нечега - Функција враћа `int`
   - ... специфичан део текста - Функција враћа `String`
3. Функција исписује - Функција не враћа ништа (`void`), само исписује

Примери:

```
// Функција која враћа највећи од 3 цела броја

public static int maximumNum(int a, int b, int c){
    int max; // може и без променљиве
    ... // Магија где у max смештамо највећи

    return max;
}

// Функција која враћа највећи елемент низа (int)

public static int maximum(int[] arr){
    int max = arr[0];
    for(...){
        ...
    }

    return max;
}

// Функција која исписује највећи елемент низа (int)

public static int writeMaximum(int[] arr){
    int max = arr[0];
    for(...){
        ...
    }

    System.out.println(max);
}
```

```java
// Функција која враћа низ елемената који су већи од прослеђеног броја (double)

public static double[] filterLargerThan(double[] arr, double n){
    double[] temp = new double[arr.length];
    int count = 0;

    for(...){
        ... // Убацимо у temp бројеве веће од n и повећамо count сваки пут кад наиђемо на такав
    }

    double[] res = new double[count]; // count означава колико има бројева већих од n

    for(...){
        ... // Убацујемо редом елементе из temp у res
    }

    return res; // Враћамо низ елемената који су већи од n
}

// Функција која враћа најмањи и највећи елемент низа целих вредности
// Тако што резултат смешта у низ где је нулти елемент најмањи, а први елемент највећи

public static int[] minAndMax(int[] arr){ // Враћамо низ од два елемента
    int mix = arr[0];
    int max = arr[0];

    for(...){
        ... // Мењамо вредности min/max кад наиђемо на мањи/већи редом
    }

    return new int[]{min, max}; // Враћамо низ тако да је на нултој позицији min, а на првој позицији max
}


// Функција која враћа имена свих особа које имају просечну оцену већу од X
// - Функција прима: Низ имена, Низ просечних оцена и X
// - Број елемената ова два низа је исти, и свака оцена одговара одговарајућој особи
// Нпр. - ["Pera", "Zika", "Ana"]
//        [7.4,6.8,9.4]
// Значи да Пера има просек 7.4, Жика 6.8 и Ана 9.4

public static String[] averageHigherThan(String[] names, double[] averages, double x){
    // Сличан принцип као код функције filterLargerThan
    // Морамо да чувамо индексе да бисмо извукли из низа , уместо да правимо низ од низа просечних оцена
}
```