



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

Лабораторна робота №8

## **Основи ООП**

Виконали  
студенти групи ІА-23:  
Семашко Олександр,  
Ширяєв Даніїл,  
Степанов Нікіта.

Перевірів:  
Колеснік В. М.

## Хід роботи:

```
1
2 public class Main {
3     public static void main(String[] args) {
4         Student defaultStudent = new Student();
5         Student student1 = new Student( name: "Danya Shyriaiev", age: 18, city: "Cherkassy", group: "IA-23", course: "PF1", new int[]{5,
6         System.out.println(defaultStudent.toString());
7         System.out.println(student1.toString());
8         student1.setName("Nikita Stepanov");
9         student1.setCity("Brovary");
10        System.out.println(student1.getName());
11        System.out.println(student1.getCity());
12        Person person1 = new Person( name: "Semashko Sasha", age: 17, city: "Baranivka");
13        System.out.println(person1.toString());
14        Student student2 = new Student(person1, group: "IA-23", course: "PF1");
15        System.out.println(student2.toString());
16        student2.addMark();
17        student2.addMark(4);
18        student2.addMark(5);
19        student2.printMarks();
20        System.out.println(student2.getAverageMark());
21        System.out.println(student1.equals(student2));
22        System.out.println(student2.equals(defaultStudent));
23    }
24 }
25 }
```

```
1
2 public class Person {
3     private String name;
4     private int age;
5     private String city;
6
7     public Person() {
8         this.name = "Elon Musk";
9         this.age = 51;
10        this.city = "Pretoria";
11    }
12
13
14    public Person(String name, int age, String city) {
15        this.name = name;
16        this.age = age;
17        this.city = city;
18    }
19
20    public int getAge() { return age; }
21
22
23
24    public String getCity() { return city; }
25
26
27 }
```

```

24     3 usages
    public String getCity() { return city; }
27
    3 usages
28     public String getName() { return name; }
31
    no usages
32     public void setAge(int age) { this.age = age; }
35
    1 usage
36     public void setCity(String city) { this.city = city; }
39
    1 usage
40     public void setName(String name) { this.name = name; }
43
    1 override
44     @Override
45     public String toString() {
46         return "Person{" +
47             "name='" + name + '\'' +
48             ", age=" + age +
49             ", city='" + city + '\'' +
50             '}';
51     }
52 }
53

```

```

1  import java.util.Arrays;
2
    6 usages
3  public class Student extends Person {
    6 usages
4      private String group;
    6 usages
5      private String course;
    19 usages
6      private int[] marks;
7
    1 usage
8      public Student() {
9          this.group = "IA-23";
10         this.course = "PF1";
11         this.marks = new int[]{1};
12     }
13
    1 usage
14     public Student(String name, int age, String city, String group, String course, int[] marks) {
15         super(name, age, city);
16         this.group = group;
17         this.course = course;
18         this.marks = marks;
19     }
20

```

```

1 usage
21 @  public Student(Person person, String group, String course) {
22     super(person.getName(), person.getAge(), person.getCity());
23     this.group = group;
24     this.course = course;
25     this.marks = new int[]{};
26 }
27
no usages
28 public String getCourse() { return course; }
31
no usages
32 public String getGroup() { return group; }
35
no usages
36 public int[] getMarks() { return marks; }
39
no usages
40 public void setCourse(String course) { this.course = course; }
43
no usages
44 public void setGroup(String group) { this.group = group; }
47
no usages
48 public void setMarks(int[] marks) { this.marks = marks; }
51
2 usages
52 public void addMark(int mark) {
53     marks = Arrays.copyOf(marks, newLength: marks.length + 1);
54     marks[marks.length - 1] = mark;
54     marks[marks.length - 1] = mark;
55 }
56
1 usage
57 public void addMark() {
58     marks = Arrays.copyOf(marks, newLength: marks.length + 1);
59     marks[marks.length - 1] = 5;
60 }
61
1 usage
62 public void printMarks() { System.out.println(Arrays.toString(marks)); }
65
1 usage
66 public float getAverageMark() {
67     float sum = 0;
68     for (int i : marks) {
69         sum += i;
70     }
71     return sum / marks.length;
72 }
73
74
75 @Override
76 public String toString() {
77     return "Student{" +
78         "name=" + getName() + '\'' +
79         ", age=" + getAge() +
80         ", city=" + getCity() + '\'' +
81         ", group=" + group + '\'' +
82         ", course=" + course + '\'' +
83         ", marks=" + Arrays.toString(marks) +

```

```
75      @Override
76      public String toString() {
77          return "Student{" +
78              "name='" + getName() + '\'' +
79              ", age=" + getAge() +
80              ", city='" + getCity() + '\'' +
81              ", group='" + group + '\'' +
82              ", course='" + course + '\'' +
83              ", marks=" + Arrays.toString(marks) +
84              '}';
85      }
86  }
87  }
```

```
Student{name='Elon Musk', age=51, city='Pretoria', group='IA-23', course='PF1', marks=[1]}
Student{name='Danya Shyriaiev', age=18, city='Cherkassy', group='IA-23', course='PF1', marks=[5, 5, 5, 5]}
Nikita Stepanov
Brovary
Person{name='Semashko Sasha', age=17, city='Baranivka'}
Student{name='Semashko Sasha', age=17, city='Baranivka', group='IA-23', course='PF1', marks=[]}
[5, 4, 5]
4.6666665
false
false
```

**Висновок:** на цій лабораторній роботі ми вивчили основи ООП, особливості мутабельних та імутабельних об'єктів, дізналися про перевантаження (overloading) та заміщення (overriding) і зрозуміли різницю між ними, працювали з полями завдяки методам та використовували свої знання ООП для реалізації практичних завдань, вивчивши предметну область задачі та покращили навички кодування на java.