

Упражнение 6 Интерфейси

Цел: Интерфейси. Имплементиране на интерфейси.

Ключови думи: interface, implements, final

Практически умения:

1. Създаване на проект от съществуващи кодове.
2. Добавяне на кодове към съществуващ проект;
3. Деклариране на интерфейс;
4. Имплементиране на интерфейс

Комбинации за бърз-достъп (shortcuts):

1. NetBeans -> recommends; (ctrl+enter)
2. NetBeans -> format code; (alt+shift+f) / ctrl+shift+f
3. NetBeans -> Стартиране, Компилиране, откриване на грешки
4. NetBeans-> коментиране/разкоментиране на няколко реда ctrl+/
5. **NetBeans-> имплементиране на интерфейс.**

Refactoring:

1. Преименуване на клас (Refactor->Rename; doRefactoring)
2. Създаване на методи за достъп (Refactor->Encapsulate Fields)

Теория: Дефиниране на интерфейс

```
[visibility] interface InterfaceName [extends other interfaces] {  
    constant declarations  
    member type declarations  
    abstract method declarations  
}
```

Правила:

1. Методите в интерфейса са: по подразбиране всички методи са публични и абстрактни. (public и abstract).
2. Полетата по подразбиране са статични и публични.
3. Интерфейсите могат да наследяват (extends) един или повече интерфейси
4. Класовете могат да имплементират един или повече интерфейси.

Конвенции:

1. Интерфейсите се именуват с префикс I и името на интерфейса (например IMachine)

Задача 1: (Пример за интерфейс)

- а) да се създаде интерфейс **Машина**, (IMachine) който да позволява включване на захранването, изключване на захранването и започване на работа;
- б) да се създаде клас **Кола** (Car), който имплементира методите на интерфейса машина;

- с) да се създаде клас **Камион** (Truck), който имплементира методите на интерфейса Машина;
- д) Да се създаде тестов клас MachineTest, в който да се тестват класовете с различни входни данни

Примерен код:

```
//създайте интерфейс в класа IMachine.java
public interface IMachine {
    //декларират се само константи и сигнатури на методи!
    public void powerOn();
    public void powerOff();
    public void drive();
}

//създайте клас във файл Car.java
public class Car implements IMachine {
    public void powerOn() {
        System.out.println("Car is switched on!");
    }
    public void powerOff() {
        System.out.println("Car is switched off!");
    }
    public void drive() {
        System.out.println("Car is running!");
    }
}
```

Задача 2 (за самостоятелна работа):

Дефинирайте подобен интерфейс за класа компютър , и го имплементирайте в клас Компютър (както задачата в упражнение 2).

Задача 3 (За работа в час!!):

- заредете примерният код даден към упражнението и го стартирайте
- Напишете класовете Point, Line, Rectangle, Circle (със характеристики център и радиус). Ако сте ги писали в упражнение 4 и 5 – ги добавете към проекта;
- Декларирайте интерфейс **IDrawable** с метод public void drawIt();
- Имплементирайте интерфейса във трите класа.
- Да се МОДИФИЦИРА кода на приложения код, с цел да може да се изчертават обекти от тип Circle, Line и Rectangle.

Задача за клас Point:

Нека имаме класа Point (с полета x, y и label). Да се създаде клас Line, наследник на класа Point със два атрибута – начална точка (наследена от класа Point) и крайна точка (startPoint и endPoint). Нека началната точка да е с координати – наследената точка, а крайната точка да се дефинира. Класа да се дефинира с трите вида конструктори, методи за достъп, метод toString и метод int compare(Line line), който връща -1 ако текущата линия е по къса, 0 ако двете линии са с еднаква дължина и 1 ако текущата линия е по дълга.

Задача за клас Rectangle

Да се направи клас, представляващ правоъгълник (Rectangle), който наследява класа Точка (Point) и има две допълнителни характеристики – широчина и височина (width и height). Да се напишат допълнително метод, който връща лицето на правоъгълника.

Задача за клас Circle

Да се напише клас Кръг (Circle) (с трите вида конструктори, метод toString, с характеристики стартова точка – от тип класа **Точка** (Point) характеристики център (centerPoint) и радиус (radius). Да се напишат методи за достъп. Да се напише метод, който връща лицето на кръга.

(За самостоятелна работа) Да се направи тестов клас GeometryTest, който да създава по няколко различни обекта от класовете Point, Rectangle, Circle, Line и да извежда някаква информация за тях.

(към упражнение 5- наследяване)

Модификатори за достъп				
Модификатор	В класа	В текущия пакет	В класовете-наследници	Във други пакети, различни от текущия
public	ДА	ДА	ДА	ДА
protected	ДА	ДА	ДА	НЕ
(без модификатор)	ДА	ДА	НЕ	НЕ
private	ДА	НЕ	НЕ	НЕ

Въпроси:

1. Какъв е синтаксиса дефиниране на интерфейс? Коя ключова дума се използва?
2. Как се имплементира интерфейс?