

2009

[OOP2009_fn9999_lab05]

Милен Петров,
milen.petrov@gmail.com

Версия:

Чернова 2 (draft-2)

SE2008

Последна редакция:
2007-30.3.2009 г.

Упражнение 5
Person/Student &&
Point3d/Line(cases)

[ОБЕКТНО-ОРИЕНТИРАНО ПРОГРАМИРАНЕ (JAVA)]

Упражнение 5 /Седмица 6 (Student case/Line case)

Упражнение 5 Наследяване

Цел: Наследяване. Родителски клас и клас наследник. Свойства. Презаписване на методи (@Override). Достъп до конструктори, свойства и методи на родителския клас.

Ключови думи: extends, super, protected, package.

Практически умения (NetBeans, Eclipse):

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+/
ctrl+shift+/*

Refactoring:

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

Задача 1: (Пример за наследяване)

- а) да се създаде клас **Човек** – който има например име (пол, възраст, рождена дата);
- б) и класа **Студент** – който е наследник на човек, т.е. има име и факултетен номер (курс, специалност).
- в) Да се създадат двата класа (т.е. с три вида конструктори), метод toString и методи за достъп.
- г) Да се създаде метод showInfo() , който да връща символен низ за двата класа, като за обектите от класа **Човек** да изписва „Човек с име ..., пол, възраст, роден на ...дата”
- д) Да се създаде метод showInfo() , който да връща символен низ за двата класа, като за обектите от класа **Студент** да изписва „Човек с име ..., пол, възраст, роден на ...дата”
- е) Да се напише метод, който връща броя на дните до следващия рожден ден, по зададена дата (ден-месец-година); **int nextBirthday(int currentDay, int currentMonth, int currentYear)**
- ж) Да се създаде тестов клас StudentPersonTest, в който да се тестват класовете с различни входни данни (да се създадат по 3 обекта от всеки клас и

```
public class Person {  
    String name;  
    public Person(String name) {  
        this.name = name;  
    }  
}
```

```
public class Student extends Person {  
    String facultyNumber;  
    public Student (String name, String facultyNumber) {  
        super(name); //or super.name = name; – каква е разликата?
```

```
}  
}
```

Задача 2:

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.

Да се направи тестов клас LineTest, който да създава 3 обекта от тип Line и да ги сравнява, като изписва кой от тях е по дълъг.

Задача 3: 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.

Въпроси:

1. Какъв е синтаксиса на наследяване на класове? Коя ключова дума се използва?
2. Какво означава термина презареждане на методи? (overloading)
myMethod(int a){}
myMethod(int a, int b){}
myMethod(String name, int age){}
3. Какво е припокриване на методи? (overriding)
4. Как се извиква метод на родителския клас? (конструктор, достъп до поле?)