Programming I - Laboratory lesson 8

Sead Jahić
Teaching Assistant, Information Technologies

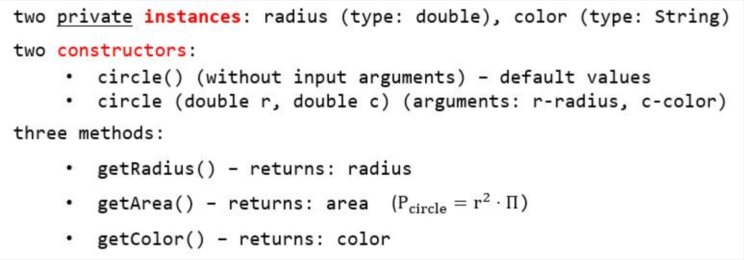
University of Primorska
Faculty of Mathematics, Natural Sciences and Information Technologies
(UP FAMNIT)

December 5^{th}, 2022.

OOP and Inheritance

Basics of OOP through example:

Exercise



two private **instances**: radius (type: double), color (type: String)

two constructors:

- circle() (without input arguments) - default values
- circle (double r, double c) (arguments: r-radius, c-color)

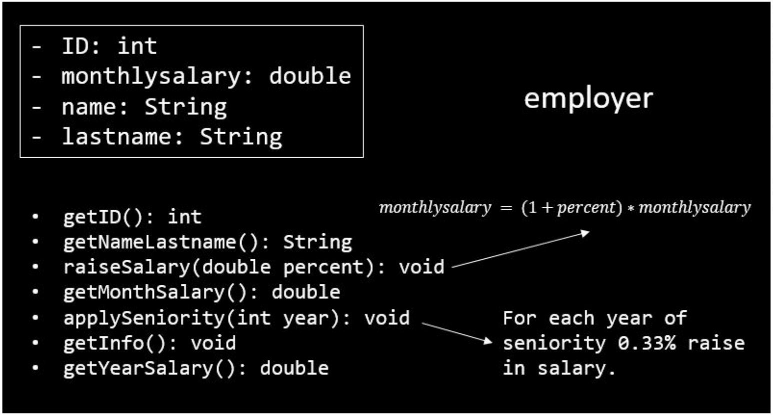
three methods:

- getRadius() - returns: radius
- getArea() - returns: area $(P_{circle} = r^2 \cdot \Pi)$
- getColor() - returns: color

Create driver class with main method and test.
Add another concept: **setter** (setRadius(double newRadius)) for instance variable radius, and another method toString()-returns information about circle (return type: String).

OOP and Inheritance - Exercise 2: The Salary

Suppose that some information about employer was given, such as: ID, name, last name, and monthly salary.
we want to calculate yearly salary of the employer. It is also known that basic salary will raise for 0.33% for the each year of the seniority.



```
- ID: int
- monthlysalary: double
- name: String
- lastname: String
```

employer

```
• getID(): int
• getNameLastname(): String
• raiseSalary(double percent): void
• getMonthSalary(): double
• applySeniority(int year): void
• getInfo(): void
• getYearSalary(): double
```

$$monthlysalary = (1 + percent) * monthlysalary$$

For each year of seniority 0.33% raise in salary.

OOP and Inheritance

Inheritance is the mechanism in java by which one class is allow to inherit the features (fields and methods) of another class.

Type of inheritance:

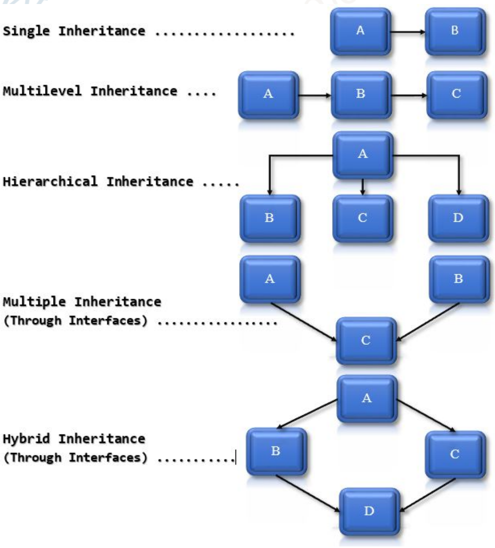
1. Single inheritance
2. Multilevel Inheritance
3. Hierarchical Inheritance
4. Multiple Inheritance (Through Interfaces)
5. Hybrid Inheritance(Through Interfaces)

In Java, it is possible to inherit attributes and methods from one class to another. We group the "inheritance concept" into two categories:

- subclass (child) - the class that inherits from another class
- superclass (parent) - the class being inherited from

To inherit from a class, use the **extends** keyword.

OOP and Inheritance



Single Inheritance ....

Multilevel Inheritance ....

Hierarchical Inheritance .....

Multiple Inheritance
(Through Interfaces) ....

Hybrid Inheritance
(Through Interfaces) ....

Inheritance

Exercise

To show use of inheritance, implements next entities:

- *base class Bicycle*
- *derived class MontainBike*

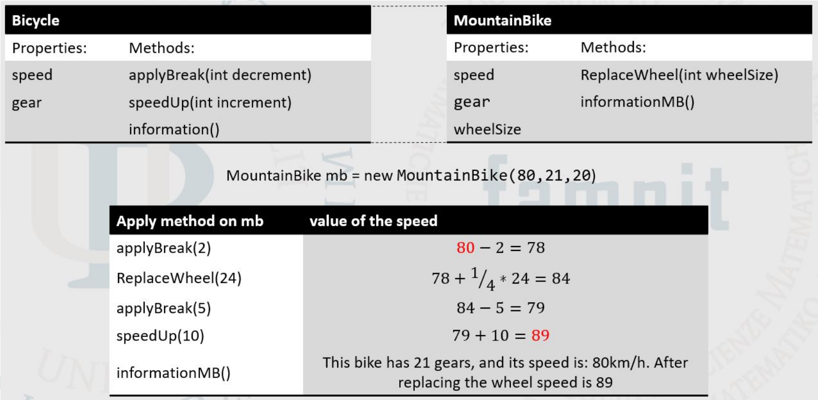
Bicycle class has next properties:

- *speed and gear and*
- *three methods:*
  - *applyBreake(int decrement) - speed is lower for decrement*
  - *speedUp(int increment) - speed is higher for increment*
  - *information() - method to print information of Bicycle.*

Class MountainBike has all Bicycle properties and:

- *Another property wheelSize (integer value) and two methods:*
  - *ReplaceWheel(int wheelSize) - method that increase speed for a quarter of wheel size*
  - *method informationMB() - method to print informations.*

Inheritance



Bicycle	
Properties:	Methods:
speed	applyBreak(int decrement)
gear	speedUp(int increment)
	information()

MountainBike	
Properties:	Methods:
speed	ReplaceWheel(int wheelSize)
gear	informationMB()
wheelSize	

MountainBike mb = new `MountainBike(80,21,20)`

Apply method on mb	value of the speed
applyBreak(2)	$80 - 2 = 78$
ReplaceWheel(24)	$78 + {}^{1}/_{4} * 24 = 84$
applyBreak(5)	$84 - 5 = 79$
speedUp(10)	$79 + 10 = 89$
informationMB()	This bike has 21 gears, and its speed is: 80km/h. After replacing the wheel speed is 89

Multilevel Inheritance

Exercise

Use multilevel inheritance to implement next entities:

- *base class* **rectangle** *with three methods area(), perimeter() (sum of its sides length) and information() method to with String return type (information about area and perimeter)*
- *derived class Cuboid and two methods volume() of Cuboid and information()*
- *another derived class Cube of class Cuboid* ($a = b = c$).