**Белорусский национальный технический университет**

**Кафедра «Программное обеспечение информационных систем и технологий»**

**Дисциплина «Программирование на языке Java»**

**Лабораторная работа №6**

Выполнили: студенты гр. 10702122 Тимощенко Н.В   
 Кудрявец Д.Е

Проверила: пр.- ст. Савчик А.

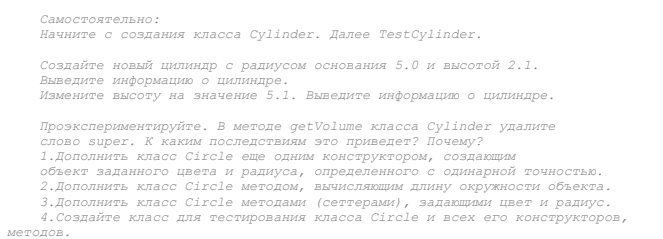
Минск 2024

TestCylinder

**Цель работы:**

Изучить классы и объекты(экземпляры классов).

**Задание:**



**Код программы:**

public class TestCircle {  
 public static void main(String[] args) {  
 *// Testing all constructors* Circle circle1 = new Circle();  
 Circle circle2 = new Circle(2.5);  
 Circle circle3 = new Circle(3.0, "blue");  
 Circle circle4 = new Circle(4.0f, "green");  
  
 *// Testing methods* System.*out*.println(circle1.toString());  
 System.*out*.println("Area: " + circle1.getArea());  
 System.*out*.println("Circumference: " + circle1.getLength());  
  
 circle2.setRadius(5.0);  
 circle2.setColor("yellow");  
 System.*out*.println(circle2.toString());  
 System.*out*.println("Area: " + circle2.getArea());  
 System.*out*.println("Circumference: " + circle2.getLength());  
 }  
}

*/\*\*  
 \* Control program for the Cylinder class  
 \*  
 \* @author Tsimoshchenko M.V, Kudryavets D.E  
 \* @version 1.0  
 \*/*public class TestCylinder {  
 public static void main(String[] args) {  
 Cylinder cylinder1 = new Cylinder(5.0, 2.1);  
 cylinder1.setHeight(3.1);  
 System.*out*.println(cylinder1.toString());  
 }  
}  
  
*/\*\*  
 \* The Cylinder class is a subclass of the Circle class  
 \* The Circle class was used earlier  
 \*/*class Cylinder extends Circle {  
 private double height;  
  
 public Cylinder() {  
 super();  
 this.height = 1.0;  
 }  
  
 public Cylinder(double radius, double height) {  
 super(radius);  
 this.height = height;  
 }  
  
 public double getHeight() {  
 return this.height;  
 }  
  
 public void setHeight(double height) {  
 this.height = height;  
 }  
  
 public double getVolume() {  
 return super.getArea() \* this.height;  
 }  
  
 @Override  
 public double getArea() {  
 return 2 \* Math.*PI* \* getRadius() \* this.height + 2 \* super.getArea();  
 }  
  
 @Override  
 public String toString() {  
 return "Cylinder: radius = " + getRadius() + ", height = " + this.height;  
 }  
}  
  
class Circle {  
 *// Private variables* private double radius;  
 private String color;  
  
 *// Constructors (overloaded)* public Circle() { *// 1st Constructor* this.radius = 1.0;  
 this.color = "red";  
 }  
  
 public Circle(double radius) { *// 2nd Constructor* this.radius = radius;  
 this.color = "red";  
 }  
  
 public Circle(double radius, String color) { *// 3rd Constructor* this.radius = radius;  
 this.color = color;  
 }  
  
 public Circle(float radius, String color) { *// 4th Constructor* this.radius = radius;  
 this.color = color;  
 }  
  
 *// Public methods* public double getRadius() {  
 return this.radius;  
 }  
  
 public String getColor() {  
 return this.color;  
 }  
  
 public double getArea() {  
 return radius \* radius \* Math.*PI*;  
 }  
  
 public double getLength() {  
 return Math.*PI* \* radius \* 2;  
 }  
  
 public void setRadius(double radius) {  
 this.radius = radius;  
 }  
  
 public void setColor(String color) {  
 this.color = color;  
 }  
}

**Скриншоты выполнения программы:**

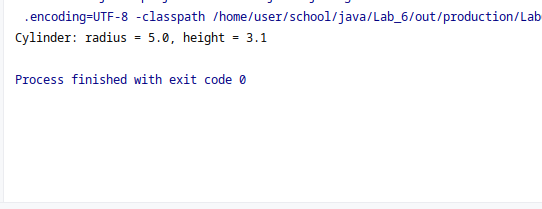
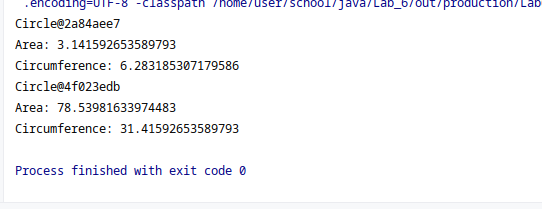


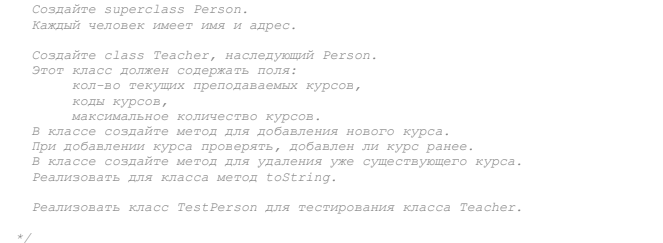
Рис. 1 – выполнение программы TestCylinder

**TestTeacher**

**Цель работы:**

Изучить классы и конструкторы.

**Задание:**



**Код программы:**

import java.util.HashSet;  
import java.util.Set;  
  
*/\*  
 \* @author Tsimoshchenko M.V, Kudryavets D.E  
 \* @version 1.0  
 \*/*class Person {  
 private String fullName;  
 private String location;  
  
 public Person(String fullName, String location) {  
 this.fullName = fullName;  
 this.location = location;  
 }  
  
 public String getFullName() {  
 return fullName;  
 }  
  
 public String getLocation() {  
 return location;  
 }  
  
 @Override  
 public String toString() {  
 return "Individual{fullName='" + fullName + "', location='" + location + "'}";  
 }  
}  
  
class Teacher extends Person {  
 private int currentCourseCount;  
 private Set<String> courseIdentifiers; *// Используем Set для хранения уникальных идентификаторов курсов* private final int maximumCourses;  
  
 public Teacher(String fullName, String location, int maximumCourses) {  
 super(fullName, location);  
 this.maximumCourses = maximumCourses;  
 this.courseIdentifiers = new HashSet<>();  
 this.currentCourseCount = 0;  
 }  
  
 public boolean addCourse(String courseIdentifier) {  
 if (currentCourseCount < maximumCourses) {  
 if (courseIdentifiers.add(courseIdentifier)) { *// Добавляет курс, если его еще нет* currentCourseCount++;  
 return true; *// Успешно добавлено* } else {  
 System.*out*.println("Курс " + courseIdentifier + " уже добавлен.");  
 return false; *// Курс уже существует* }  
 } else {  
 System.*out*.println("Достигнуто максимальное количество курсов.");  
 return false; *// Превышено максимальное количество курсов* }  
 }  
  
 public boolean removeCourse(String courseIdentifier) {  
 if (courseIdentifiers.remove(courseIdentifier)) { *// Удаляет курс, если он существует* currentCourseCount--;  
 return true; *// Успешно удалено* } else {  
 System.*out*.println("Курс " + courseIdentifier + " не найден.");  
 return false; *// Курс не найден* }  
 }  
  
 @Override  
 public String toString() {  
 return "Teacher{" +  
 "fullName='" + getFullName() + '\'' +  
 ", location='" + getLocation() + '\'' +  
 ", currentCourseCount=" + currentCourseCount +  
 ", courseIdentifiers=" + courseIdentifiers +  
 ", maximumCourses=" + maximumCourses +  
 '}';  
 }  
}  
  
*// Пример использования классов*class TestTeacher {  
 public static void main(String[] args) {  
 Teacher teacher = new Teacher("Алексей Смирнов", "Минск", 5);  
  
 teacher.addCourse("C++");  
 teacher.addCourse("Python");  
  
 System.*out*.println(teacher);  
  
 teacher.removeCourse("C++");  
  
 System.*out*.println(teacher);  
  
 teacher.addCourse("C++");  
 teacher.addCourse("Java");  
 teacher.addCourse("Go");  
 teacher.addCourse("Rust");  
 teacher.addCourse("Swift");  
  
 System.*out*.println(teacher);  
 }  
}

**Скриншоты выполнения программы:**

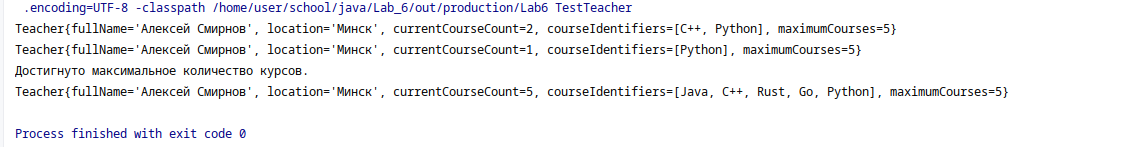


Рис. 2 – выполнение программы TestTeacher

**Контрольные вопросы:**

**Конструктор по умолчанию** – это специальный метод класса, который вызывается при создании объекта и не принимает параметров. Если в классе не определен явный конструктор, компилятор автоматически создает конструктор по умолчанию.

**Создание объектов** происходит с помощью оператора new. Например:

// MyClass obj = new MyClass();

**Наличие конструкторов** в классе не обязательно. Если конструктор не определен, компилятор автоматически создает конструктор по умолчанию.

**Наследование** – это механизм, позволяющий одному классу (наследнику) унаследовать поля и методы другого класса (родителя). Это позволяет создавать новые классы на основе существующих, добавляя или изменяя функциональность.

**Строка** public class Cylinder extends Circle означает, что класс Cylinder наследует класс Circle. Это значит, что Cylinder будет иметь все свойства и методы Circle, а также может добавлять свои собственные или переопределять унаследованные.