

ЛАБОРАТОРНАЯ РАБОТА № 7

«Введение в наследование»

Цель: получить представление о механизме наследования в языке программирования Java

Учебные вопросы:

1. Пример решения первой задачи;
2. Задания для самостоятельной работы;
3. Описание результата выполнения лабораторной работы;

1 ПРИМЕР РЕШЕНИЯ ПЕРВОЙ ЗАДАЧИ

```
package com.company;

public class Main {

    public static void main(String[] args) {

        superClassTest superClassObject = new superClassTest("передал в конструктор суперкласса");
        String className = superClassObject.toString();
        System.out.println(className);

        subClassTest subClassObject1 = new subClassTest("передал в конструктор подкласса");
        String subClassName = subClassObject1.toString();
        System.out.println(subClassName);

        subClassTest subClassObject2 = new subClassTest("передал в конструктор подкласса", "где два параметра");
        String subClassName2 = subClassObject2.toString();
        System.out.println(subClassName2);

    }
}
```

```
package com.company;

public class superClassTest {

    //закрытое текстовое поле
    private String str1;

    //конструктор с текстовым аргументом
    superClassTest(String strEx){
        this.str1 = strEx;
    }
}
```

```
superClassTest() {  
}
```

//@Override - аннотация указываемая для того, чтобы показать что далее мы собираемся переопределять метод суперкласса

```
@Override
```

```
public String toString() {  
    String superClassNameAndFieldValue;  
    superClassNameAndFieldValue = "super" + "\n" +  
        " Class name: " + this.getClass().getSimpleName() + "\n" +  
        " str 1 = " + this.getStr1();  
  
    return superClassNameAndFieldValue ;  
}
```

```
public String getStr1() {  
    return str1;  
}
```

```
public void setStr1(String str1) {  
    this.str1 = str1;  
}  
}
```

```
package com.company;
```

```
public class subClassTest extends superClassTest {
```

```
    private String str2;  
    private String str3;
```

//конструктор с одним параметром

```
subClassTest(String strEx) {  
    super(strEx);  
}
```

//конструктор с двумя параметрами

```
subClassTest(String strEx2, String strEx3){  
    this.str2 = strEx2;  
    this.str3 = strEx3;  
}
```

//@Override - аннотация для указываемая для того, чтобы показать что далее мы собираемся переопределять метод суперкласса

@Override

```
public String toString() {  
    String ClassNameAndFieldValue;  
    ClassNameAndFieldValue = "sub" + "\n" +  
        " Class name: " + this.getClass().getSimpleName() + "\n" +  
        " str 1 = " + this.getStr1() + "\n" +  
        " str 2 = " + this.str2 + "\n" +  
        " str 3 = " + this.str3;  
    return ClassNameAndFieldValue ;  
}
```

```
public String getStr3() {  
    return str3;  
}
```

```
public void setStr3(String str3) {  
    this.str3 = str3;  
}
```

```
public String getStr2() {  
    return str2;  
}
```

```
public void setStr2(String str2) {  
    this.str2 = str2;  
}  
}
```

2 ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Напишите программу, в которой есть суперкласс с приватным текстовым полем, конструктором с текстовым параметром и где переопределен метод `toString ()`. На основе суперкласса путем наследования создается подкласс. У него появляется еще одно приватное текстовое поле. Также подкласс должен иметь версии конструктора с одним и двумя текстовыми аргументами, а еще в нем должен быть переопределен метод `toString ()`. В обоих классах метод `toString ()` переопределяется так, что он возвращает строку с названием класса и значение текстового поля или текстовых полей.
2. Напишите программу, в которой есть суперкласс с приватным текстовым полем. В базовом классе должен быть метод для присваивания значения полю: без параметров и с одним текстовым параметром. Объект суперкласса создается передачей одного текстового аргумента конструктору. Доступное только для чтения свойство результатом возвращает длину текстовой строки. На основе суперкласса создается подкласс. В подклассе появляется дополнительное открытое целочисленное поле. В классе должны быть такие версии метода для присваивания значений полям (используется переопределение и перегрузка метода из суперкласса): без параметров, с текстовым параметром, с целочисленным параметром, с текстовым и целочисленным параметром. У конструктора подкласса два параметра (целочисленный и текстовый).
3. Напишите программу, в которой на основе суперкласса создается подкласс, а на основе этого подкласса создается еще один подкласс (цепочка наследования из трех классов). В первом суперклассе есть открытое целочисленное поле, метод с одним параметром для присваивания значения полю и конструктор с одним параметром. Во втором классе появляется открытое символьное поле, метод с двумя параметрами для присваивания значения полям (перегрузка метода из суперкласса) и конструктор с двумя параметрами. В третьем классе появляется открытое текстовое поле, метод с тремя аргументами для присваивания значений полям (перегрузка метода из суперкласса) и конструктор с тремя параметрами. Для каждого класса определите метод `toString ()` так, чтобы он возвращал строку с названием класса и значениями всех полей объекта.
4. Напишите программу, в которой использована цепочка наследования из трех классов. В первом классе есть открытое символьное поле. Во втором классе появляется открытое текстовое поле. В третьем классе появляется открытое целочисленное поле. В каждом из классов должен быть конструктор, позволяющий создавать объект на основе значений полей, переданных аргументами конструктору, а также конструктор создания копии.
5. Напишите программу, в которой есть суперкласс с защищенным текстовым полем, конструктор с текстовым параметром и метод, при вызове которого в консольном окне отображается название класса и значение поля. На основе суперкласса создаются два подкласса (оба на основе одного и того же суперкласса). В одном из классов появляется защищенное целочисленное поле, там есть конструктор с двумя параметрами и переопределен метод для отображения значений полей объекта и названия класса. Во втором подклассе появляется защищенное символьное поле, конструктор с двумя параметрами и переопределен метод, отображающий в консоли название класса и значения полей. В главном методе создайте объекты каждого из классов. Проверьте работу метода, отображающего значения полей объектов, для каждого из объектов. Вызовите этот же метод через объектную переменную суперкласса, которая ссылается на объект производного класса.

3 ОПИСАНИЕ РЕЗУЛЬТАТА ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

В отчете по лабораторной работе должны быть представлены 5 листингов программ.

Лабораторная работа принимается при наличии отчета и десяти работающих, верно, выполненных программ, и наличии архива с десятью текстовыми файлами программ с расширением class. У каждого студента будут проверены выборочно программы (3-5 штук).

Структура отчета по лабораторной работе:

1. Титульный лист;
2. Цель работы;
3. Описание задачи;
4. Ход выполнения (содержит код программы);
5. Вывод;

Оформление:

- а) шрифт Times New Roman;
- б) размер шрифта 12 или 14;
- в) межстрочный интервал 1,5.

Отчет выполняется индивидуально и направляется по адресу электронной почты proverkalab@yandex.ru. В теле письма необходимо указать ФИО студента и номер группы.