|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Отчет по выполнению лабораторной работы 4

**Тема. ИНТЕРФЕЙСЫ В JAVA.**

Дисциплина Программирование на языке Джава

Выполнил

|  |  |
| --- | --- |
| студент | Болотов Михаил |
|  | Фамилия И.О. |
| группа | ИКБО-06-19 |
|  | Номер группы |

Москва 2020

**Содержание**

[​ Теория 2](#__RefHeading___Toc4128_811265792)

[​ Задание 2](#__RefHeading___Toc4136_811265792)

[​ Код 3](#__RefHeading___Toc4138_811265792)

[​ Planet.java 3](#__RefHeading___Toc52263_811265792)

[​ Nameable.java 3](#__RefHeading___Toc52265_811265792)

[​ Person.java 3](#__RefHeading___Toc52267_811265792)

[​ NameableTest.java 3](#__RefHeading___Toc52269_811265792)

[​ Скриншот 4](#__RefHeading___Toc3748_811265792)

[​ Заключение 4](#__RefHeading___Toc3750_811265792)

[​ Библиографический список 4](#__RefHeading___Toc3752_811265792)

# Теория

Механизм наследования очень удобен, но он имеет свои ограничения. В частности мы можем наследовать только от одного класса, в отличие, например, от языка С++, где имеется множественное наследование. В языке Java подобную проблему позволяют решить интерфейсы. Интерфейсы определяют некоторый функционал, не имеющий конкретной реализации, который затем реализуют классы, применяющие эти интерфейсы. И один класс может применить множество интерфейсов. Чтобы определить интерфейс, используется ключевое слово interface.

Интерфейс может определять различные методы, которые, так же как и абстрактные методы абстрактных классов не имеют реализации. В данном случае объявлен только один метод. Все методы интерфейса не имеют модификаторов доступа, но фактически по умолчанию доступ public, так как цель интерфейса - определение функционала для реализации его классом. Поэтому весь функционал должен быть открыт для реализации. И также при объявлении интерфейса надо учитывать, что только один интерфейс в файле может иметь тип доступа public. А его название должно совпадать с именем файла. Остальные интерфейсы (если такие имеются в файле java) не должны иметь модификаторов доступа. Интерфейс может определять различные методы, которые, так же как и абстрактные методы абстрактных классов не имеют реализации. В данном случае объявлен только один метод. Все методы интерфейса не имеют модификаторов доступа, но фактически по умолчанию доступ public, так как цель интерфейса - определение функционала для реализации его классом. Поэтому весь функционал должен быть открыт для реализации. И также при объявлении интерфейса надо учитывать, что только один интерфейс в файле может иметь тип доступа public. А его название должно совпадать с именем файла. Остальные интерфейсы (если такие имеются в файле java) не должны иметь модификаторов доступа.

# Задание

Создать интерфейс Nameable, с методом getName(), возвращающим имя объекта, реализующего интерфейс. Проверить работу для различных объектов (например, можно создать классы, описывающие разные сущности, которые могут иметь имя: планеты, машины, животные и т. д.).

# Код

### Planet.java

package dev.ky3he4ik.lab.lab4;  
  
public class Planet implements Nameable {  
 @Override  
 public String getName() {  
 return "Pluto";  
 }  
}

### Nameable.java

package dev.ky3he4ik.lab.lab4;  
  
public interface Nameable {  
 String getName();  
}

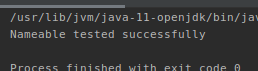
### Person.java

package dev.ky3he4ik.lab.lab4;  
  
public class Person implements Nameable {  
 @Override  
 public String getName() {  
 return "Vasya";  
 }  
}

### NameableTest.java

package dev.ky3he4ik.lab.lab4;  
  
public class NameableTest {  
 public static void main(String[] args) {  
 Nameable nameable = new Person();  
 assert nameable.getName().equals("Vasya");  
 nameable = new Planet();  
 assert nameable.getName().equals("Pluto");  
 System.out.println("Nameable tested successfully");  
 }  
}

# Скриншот



# Заключение

В данной лабораторной работе я изучил понятие интерфейса, научился создавать интерфейсы в Java и применять их в программах.

# Библиографический список

1. Зорина Н.В. Курс лекций по Объектно-ориентированному программированию на Java, МИРЭА, Москва, 2016
2. Программирование на языке Java: работа со строками и массивами. Методические указания. [Электронный ресурс] : Учебно-методические пособия — Электрон. дан. — СПб. : ПГУПС, 2015. — 24 с.
3. Кожомбердиева, Г.И. Программирование на языке Java: создание графического интерфейса пользователя: учеб. пособие. [Электронный ресурс] : Учебные пособия / Г.И. Кожомбердиева, М.И. Гарина. — Электрон. дан. — СПб.: ПГУПС, 2012. — 67 с.
4. Вишневская, Т.И. Технология программирования. Часть 1. [Электронный ресурс] / Т.И. Вишневская, Т.Н. Романова. — Электрон. дан. — М. : МГТУ им. Н.Э. Баумана, 2007. — 59 с.