|  |  |
| --- | --- |
|  | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | |
| Институт инновационных технологий и государственного управления (ИНТЕГУ) | |
| Кафедра информационных технологий в государственном управлении (ИТГУ) | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ** | |
| **по дисциплине** | |
| **«Объективно-ориентированное программирование»**  **на тему**  **«НАСЛЕДОВАНИЕ В JAVA»** | |
|  | |
| Выполнил студент группы ГИБО-03-17 | *Стояновска М.* |
| Принял  *Старший преподаватель* | *Хлебникова В.Л*  *.* |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_201\_\_\_ г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_201\_\_\_ г. |  |

Москва 2019

Оглавление

[**Цель работы 3**](#_Toc27156600)

[**Задание 3**](#_Toc27156601)

[**Код и результат 4**](#_Toc27156602)

# Цель работы

Цель данной лабораторной работы - изучить понятие наследования, и

научиться реализовывать наследование в Java.

**Задание**

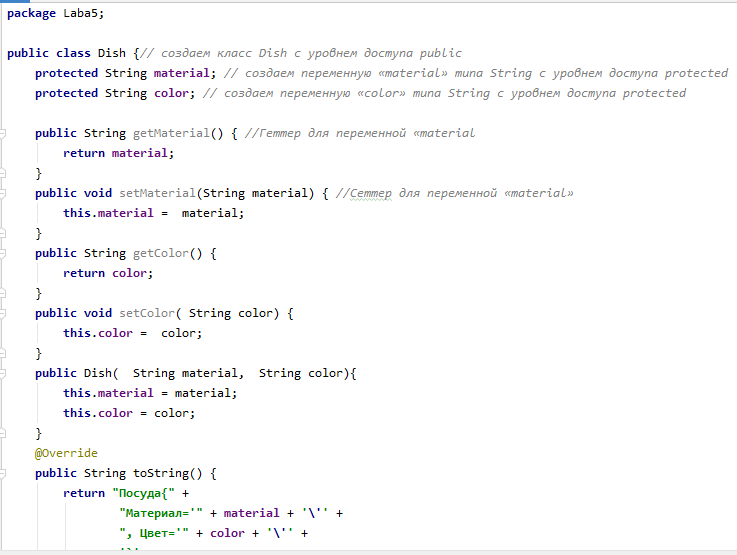
Создать абстрактный класс, описывающий посуду(Dish). С помощью

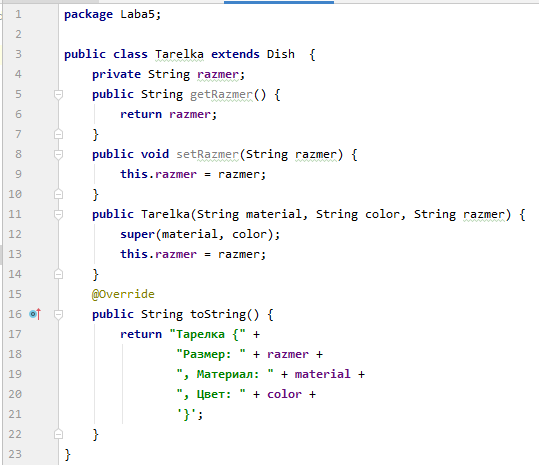
наследования реализовать различные виды посуды, имеющие свои свойства и

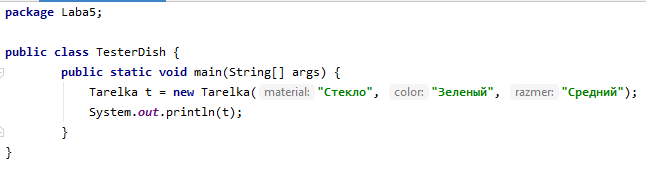
методы. Протестировать работу классов.

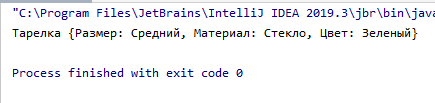
# Код и результат

**Скрины:**









**Код программы:**

**package** Laba5;  
  
**public class** Dish {*// создаем класс Dish с уровнем доступа public* **protected** String **material**; *// создаем переменную «material» типа String с уровнем доступа protected* **protected** String **color**; *// создаем переменную «color» типа String с уровнем доступа protected* **public** String getMaterial() { *//Геттер для переменной «material* **return material**;  
 }  
 **public void** setMaterial(String material) { *//Cеттер для переменной «material»* **this**.**material** = material;  
 }  
 **public** String getColor() {  
 **return color**;  
 }  
 **public void** setColor( String color) {  
 **this**.**color** = color;  
 }  
 **public** Dish( String material, String color){  
 **this**.**material** = material;  
 **this**.**color** = color;  
 }  
 @Override  
 **public** String toString() {  
 **return "Посуда{"** +  
 **"Материал='"** + **material** + **'\''** +  
 **", Цвет='"** + **color** + **'\''** +  
 **'}'**;  
 }  
}

**package** Laba5;  
  
**public class** Tarelka **extends** Dish {  
 **private** String **razmer**;  
 **public** String getRazmer() {  
 **return razmer**;  
 }  
 **public void** setRazmer(String razmer) {  
 **this**.**razmer** = razmer;  
 }  
 **public** Tarelka(String material, String color, String razmer) {  
 **super**(material, color);  
 **this**.**razmer** = razmer;  
 }  
 @Override  
 **public** String toString() {  
 **return "Тарелка {"** +  
 **"Размер: "** + **razmer** +  
 **", Материал: "** + **material** +  
 **", Цвет: "** + **color** +  
 **'}'**;  
 }  
}

**package** Laba5;  
  
**public class** TesterDish {  
 **public static void** main(String[] args) {  
 Tarelka t = **new** Tarelka(**"Стекло"**, **"Зеленый"**, **"Средний"**);  
 System.***out***.println(t);  
 }  
}

**Итоги выполнения лабораторной работы**

В ходе выполнения лабораторной работы №3 мы научились: создавать абстрактный класс в нашем случае (Dish) и наследоваться от него, при создании других классов. В результате мы можем использовать методы абстрактного класса и не описывать их повторно.