МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ**

**ЛАБОРАТОРНАЯ РАБОТА № 3**

по дисциплине

‘ПРОГРАММИРОВАНИЕ’

Вариант №90006

*Выполнил:*

Студент группы P3131

Кадилов Михаил Владимирович

*Преподаватель:*

Николаев Владимир Вячеславович



Санкт-Петербург, 2022

**Содержание**

[**Текст**](#_heading=h.rkkjfxwgz5dy) **4**

[**Диаграмма классов**](#_heading=h.fmu5semu0dlz) **4**

[**Исходный код**](#_heading=h.eqww32m17wbu) **5**

[Main.java](#_heading=h.2et92p0) 5

[CosmicFood.java](#_heading=h.tyjcwt) 6

[Cutlet.java](#_heading=h.3dy6vkm) 7

[Soup.java](#_heading=h.1t3h5sf) 7

[Sausage.java](#_heading=h.4d34og8) 9

[TomatoJuiсe.java](#_heading=h.2s8eyo1) 10

[Edible.java](#_heading=h.17dp8vu) 10

[NonEdible.java](#_heading=h.oxzqygus4tsb) 10

[Human.java](#_heading=h.bg4jzx8s423z) 11

[Neznayka.java](#_heading=h.26in1rg) 12

[Ponchik.java](#_heading=h.lnxbz9) 13

[Результат работы](#_heading=h.2xcytpi) **14**

[Вывод](#_heading=h.1ci93xb) **15**

# Текст

Друзья спустились в пищевой отсек. Незнайке совсем еще не хотелось есть, и он только для того, чтоб составить компанию Пончику, съел одну космическую котлетку. Но Пончик решил не теряться в создавшейся обстановке и отнесся к делу со всей серьезностью. Он заявил, что должен произвести в пищевом отсеке ревизию и проверить качество всех космических блюд, а для этого ему нужно съесть хотя бы по одной порции каждого блюда. Эта задача оказалась, однако, для него не под силу, потому что уже на десятой или на одиннадцатой порции его сморил сон, и Пончик заснул с недоеденной космической сосиской во рту. В этом ничего удивительного не было, так как ночью Пончик спал мало, к тому же каждый, кто находится в состоянии невесомости, может заснуть в любой позе, не укладываясь для этого специально в постель.

# Диаграмма классов

# 

# 

# 

# 

# 

# 

# Исходный код

#### Main.java

import people.\*;  
import food.\*;  
  
public class Main {  
 public static void main(String[] args) {  
 Soup soup1 = new Soup("борщ (красный)", 10);  
 Soup soup2 = new Soup("борщ (оранжевый)", 10);  
 Cutlet cutlet = new Cutlet("Котлета по-киевски", 12);  
 Sausage sausage = new Sausage("Баварская колбаска к пиву", 11);  
 TomatoJuice tomatoJuice = new TomatoJuice("Томатный сок", 10);  
 System.*out*.println();  
  
  
 System.*out*.println("Проверка equals, hashCode, toString для еды.");  
 System.*out*.println(soup1.equals(soup2));  
 System.*out*.println(soup1.hashCode());  
 System.*out*.println(tomatoJuice);  
 System.*out*.println();  
  
  
 Neznayka neznayka = new Neznayka("главная палуба", 2100);  
 Ponchik ponchik = new Ponchik("главная палуба", 350);  
 System.*out*.println();  
  
  
 System.*out*.println("Проверка equals, hashCode, toString для персонажей.");  
 System.*out*.println(neznayka.equals(ponchik));  
 System.*out*.println(ponchik.hashCode());  
 System.*out*.println(neznayka);  
 System.*out*.println();  
  
 neznayka.go("пищеблок");  
 ponchik.go("пищеблок");  
 System.*out*.println();  
  
 neznayka.showHunger();  
 ponchik.showHunger();  
 System.*out*.println();  
  
 neznayka.eat(cutlet, 1);  
 neznayka.showHunger();  
 ponchik.eat(sausage, 11);  
 ponchik.showHunger();  
 System.*out*.println();  
  
  
 System.*out*.println("Незнайка хочет спать? " + neznayka.getSleepy());  
 System.*out*.println("Пончик хочет спать? " + ponchik.getSleepy());  
 System.*out*.println();  
  
 ponchik.sleep();  
 }  
}

#### 

#### CosmicFood.java

package food;  
  
import java.util.Objects;  
  
public abstract class CosmicFood{  
 private String name;  
 private int calories;  
 private double weight;  
 private int numberOfServings;  
  
 protected CosmicFood() {  
 }  
  
 public String getName(){  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public int getCalories(){  
 return calories;  
 }  
  
 public void setCalories(int calories) {  
 this.calories = calories;  
 }  
  
 public double getWeight() {  
 return weight;  
 }  
  
 public void setWeight(double weight) {  
 this.weight = weight;  
 }  
  
 public int getNumberOfServings(){  
 return numberOfServings;  
 }  
  
 public void setNumberOfServings(int numberOfServings) {  
 this.numberOfServings = numberOfServings;  
 }  
  
 public void eat(int amount){  
 numberOfServings -= amount;  
  
 if (numberOfServings == 0){  
 System.*out*.println("Внимание! На ракете закончился продукт: " + name);  
 }  
 }  
  
 @Override  
 public String toString() {  
 return "Космическая еда " +  
 "под назавнием:'" + name + '\'' +  
 ", калорийность:" + calories +  
 ", вес:" + weight +  
 ", количество порций:" + numberOfServings;  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(name, calories, weight, numberOfServings);  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 CosmicFood that = (CosmicFood) o;  
 return calories == that.calories;  
 }  
  
 public abstract FoodType getType();  
  
}

#### Cutlet.java

package food;  
  
public class Cutlet extends CosmicFood implements Edible{  
 public Cutlet(String name, int numberOfServings){  
 setName(name);  
 setNumberOfServings(numberOfServings);  
 setWeight(180);  
 setCalories(200);  
 System.*out*.println("Продукт " + name + " в количестве " + numberOfServings + " загружен в ракету.");  
 }  
  
 public FoodType getType() {  
 return FoodType.*SECOND*;  
 }  
  
 @Override  
 public void use() {  
 if (getNumberOfServings() > 0) {  
 setNumberOfServings(getNumberOfServings() - 1);  
 System.*out*.println("Съеден(а) 1 " + getName());  
 } else {  
 System.*out*.println("Есть уже нечего...");  
 }  
 }  
}

#### Soup.java

package food;  
  
public class Soup extends CosmicFood implements Edible{  
 public Soup(String name, int numberOfServings){  
 setName(name);  
 setNumberOfServings(numberOfServings);  
 setWeight(300);  
 setCalories(180);  
 System.*out*.println("Продукт " + name + " в количестве " + numberOfServings + " загружен в ракету.");  
 }  
  
 public FoodType getType() {  
 return FoodType.*FIRST*;  
 }  
  
 @Override  
 public void use() {  
 if (getNumberOfServings() > 0) {  
 setNumberOfServings(getNumberOfServings() - 1);  
 System.*out*.println("Съеден(а) 1 " + getName());  
 }  
 else {  
 System.*out*.println("Есть уже нечего...");  
 }  
 }  
}

#### Sausage.java

package food;  
  
public class Sausage extends CosmicFood implements Edible{  
 public Sausage(String name, int numberOfServings){  
 setName(name);  
 setNumberOfServings(numberOfServings);  
 setWeight(170);  
 setCalories(210);  
 System.*out*.println("Продукт " + name + " в количестве " + numberOfServings + " загружен в ракету.");  
 }  
  
 public FoodType getType() {  
 return FoodType.*SECOND*;  
 }  
  
 @Override  
 public void use() {  
 if (getNumberOfServings() > 0) {  
 setNumberOfServings(getNumberOfServings() - 1);  
 System.*out*.println("Съеден(а) 1 " + getName());  
 } else {  
 System.*out*.println("Есть уже нечего...");  
 }  
 }  
  
}

#### TomatoJuiсe.java

package food;  
  
public class TomatoJuice extends CosmicFood implements NonEdible {  
 public TomatoJuice(String name, int numberOfServings) {  
 setName(name);  
 setNumberOfServings(numberOfServings);  
 setWeight(120);  
 setCalories(60);  
 System.*out*.println("Продукт " + name + " в количестве " + numberOfServings + " загружен в ракету.");  
 }  
  
 public FoodType getType() {  
 return FoodType.*DRINK*;  
 }  
  
  
 @Override  
 public void throwAway() {  
 if (getNumberOfServings() > 0) {  
 setNumberOfServings(getNumberOfServings() - 1);  
 System.*out*.println("Выкинул " + getName() + ". Кто вообще ЭТО ест?");  
 } else {  
 System.*out*.println("Какой-то добрый человек уже выкинул эту гадость за тебя.");  
 }  
 }  
}

#### Edible.java

1. package food;
2. public interface Edible {
3. void use();
4. }

#### NonEdible.java

* + - 1. package food;
      2. public interface NonEdible {
      3. void throwAway();
      4. }

#### Human.java

package people;  
import food.CosmicFood;  
  
import java.util.Objects;  
  
public abstract class Human{  
 private String name;  
 private String location;  
 private int satiety;  
 private boolean conscious = true;  
 private boolean sleepy;  
  
 public String getName(){  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public void setSleepy(boolean sleepy) {  
 this.sleepy = sleepy;  
 }  
 public boolean getSleepy() {  
 return sleepy;  
 }  
  
 public void setLocation (String location) {  
 this.location = location;  
 }  
 public String getLocation (){  
 return location;  
 }  
  
 public void setSatiety(int satiety) {  
 this.satiety = satiety;  
 }  
  
 public int getSatiety() {  
 return satiety;  
 }  
  
 public void setConscious(boolean conscious) {  
 this.conscious = conscious;  
 }  
  
 public boolean getConscious() {  
 return conscious;  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Human human = (Human) o;  
 return name.equals(human.name);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(name, location, satiety, conscious);  
 }  
  
 @Override  
 public String toString() {  
 return "Персонаж " +  
 "под именем '" + name + '\'' +  
 " находится в локации:'" + location + '\'' +  
 ", сыт на " + satiety +  
 ", находится в сознаии: " + conscious;  
 }  
  
 public void eat(CosmicFood dish, int amount){  
 if (dish.getNumberOfServings() - amount >= 0){  
 dish.eat(amount);  
 if (satiety + dish.getCalories() \* amount >= 2500){  
 satiety += dish.getCalories() \* amount;  
 dish.setNumberOfServings(dish.getNumberOfServings() + (satiety - 2500) / dish.getCalories());  
 if ((satiety - 2500) / dish.getCalories() != 0) {  
 System.*out*.println(name + " объелся. Он не доел " + dish.getName() + " в количестве " + (satiety - 2500) / dish.getCalories() + ". Эта еда вернулась на склад.");  
 } else System.*out*.println(name + " объелся. Он не доел " + dish.getName() + ".");  
 satiety = 2500;  
  
 }  
 else {  
 satiety += dish.getCalories();  
 System.*out*.println(name + " съел " + dish.getName() + " в количестве: " + amount);  
 }  
 }  
 else {  
 System.*out*.println("На ракете нет такого количества еды...");  
 }  
  
 }  
  
 public void go(String place){  
 System.*out*.println(name + " перешёл из " + location + " в " + place + ".");  
 location = place;  
 }  
  
 public abstract void sleep();  
  
 public void wakeUp() {  
 setConscious(true);  
 System.*out*.println(getName() + " проснулся.");  
 }  
  
 public void showHunger() {  
 if (satiety > 2000) {  
 System.*out*.println(name + " не голоден.");  
 System.*out*.println(name + " сыт на " + satiety + "/2500.");  
  
 }  
 else {  
 System.*out*.println(name + " сыт на " + satiety + "/2500.");  
 }  
 }  
}

#### Neznayka.java

package people;  
  
  
public class Neznayka extends Human{  
  
 public Neznayka(String location, int satiety) {  
 setLocation(location);  
 setSatiety(satiety);  
 setName("Незнайка");  
 setSleepy(false);  
 System.*out*.println(getName() + " появился в ракете.");  
 }  
  
 public void sleep(){  
 System.*out*.println(getName() + " заснул.");  
 setSleepy(false);  
 setConscious(false);  
 }  
  
  
}

#### Ponchik.java

package people;  
  
  
public class Ponchik extends Human{  
 public Ponchik(String location, int satiety) {  
 setLocation(location);  
 setSatiety(satiety);  
 setName("Пончик");  
 setSleepy(true);  
 System.*out*.println(getName() + " появился в ракете.");  
 }  
   
 public void sleep(){  
 if (getSatiety() >= 2500) {  
 System.*out*.println(getName() + " заснул c едой во рту.");  
 }  
 else {  
 System.*out*.println(getName() + " заснул.");  
 }  
 setSleepy(false);  
 setConscious(false);  
 }  
}

# Результат работы

# Продукт борщ (красный) в количестве 10 загружен в ракету.

# Продукт борщ (оранжевый) в количестве 10 загружен в ракету.

# Продукт Котлета по-киевски в количестве 12 загружен в ракету.

# Продукт Баварская колбаска к пиву в количестве 11 загружен в ракету.

# Продукт Томатный сок в количестве 10 загружен в ракету.

# Проверка equals, hashCode, toString для еды.

# true

# -1548755124

# Космическая еда под назавнием:'Томатный сок', калорийность:60, вес:120.0, количество порций:10

# Незнайка появился в ракете.

# Пончик появился в ракете.

# Проверка equals, hashCode, toString для персонажей.

# false

# 2014311149

# Персонаж под именем 'Незнайка' находится в локации:'главная палуба', сыт на 2100, находится в сознаии: true

# Незнайка перешёл из главная палуба в пищеблок.

# Пончик перешёл из главная палуба в пищеблок.

# Незнайка не голоден.

# Незнайка сыт на 2100/2500.

# Пончик сыт на 350/2500.

# Незнайка съел Котлета по-киевски в количестве: 1

# Незнайка не голоден.

# Незнайка сыт на 2300/2500.

# Внимание! На ракете закончился продукт: Баварская колбаска к пиву

# Пончик объелся. Он не доел Баварская колбаска к пиву.

# Пончик не голоден.

# Пончик сыт на 2500/2500.

# Незнайка хочет спать? false

# Пончик хочет спать? true

# Пончик заснул c едой во рту.

# Process finished with exit code 0

# Вывод

При выполнении данной лабораторной работы я продолжил изучение основных принципов ООП в Java, научился работать с интерфейсами и абстрактными классами.

