Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Рязанский государственный радиотехнический университет»  
(ФГБОУ ВО «РГРТУ», РГРТУ)

Кафедра «Вычислительная и прикладная математика» (ВПМ)

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 5

по дисциплине

**«ПРОГРАММИРОВАНИЕ НА JAVA»**

Выполнил:

студент группы 643

Паршина Анна Романовна

Проверил:

Пруцков Александр Викторович,

д-р техн. наук, профессор кафедры ВПМ

Рязань 2019

# Задание

Фургон кофе. Загрузить фургон определенного объема груза на определенную сумму из различных сортов кофе, находящихся к тому же в разных физических состояниях (земно, молотый, растворимый в банках и пакетиках. Учитывать объем кофе вместе с упаковкой. Провести сортировку товаров на основе соотношения цены и веса. Найти в фургоне товар.)

# Основные классы, реализующие задание

## Класс Runner

package ru.rsreu.parshina0513;

import ru.rsreu.parshina0513.coffee.Coffee;

import ru.rsreu.parshina0513.coffeevan.CoffeeVan;

public class Runner {

private Runner() {

}

public static void main(String[] args) {

StringBuilder result = new StringBuilder();

CoffeeVan van = new CoffeeVan();

result.append(Resourcer.getString("message.array")).append(van.getPrice()).append("\n")

.append(Resourcer.getString("message.vanBefore")).append("\n").append(van).append("\n");

van.sortCargo();

result.append(Resourcer.getString("message.vanAfter")).append("\n").append(van).append("\n");

Coffee findItem = van.getCargo()[1];

result.append(findItem).append(" - ").append(van.findProduct(findItem));

System.out.print(result);

}

}

## Класс CoffeeVan

package ru.rsreu.parshina0513.coffeevan;

import java.util.Arrays;

import ru.rsreu.parshina0513.Resourcer;

import ru.rsreu.parshina0513.coffee.Coffee;

import ru.rsreu.parshina0513.coffee.PriceToWeigthComparator;

public class CoffeeVan {

public static final double VOLUME = 1000;

private Coffee[] cargo;

public CoffeeVan() {

this.setCargo();

}

public final double getPrice() {

int sumPrice = 0;

for (int i = 0; i < this.cargo.length; i++) {

sumPrice += cargo[i].getPriceCoffee();

}

return sumPrice;

}

public final void setCargo() {

CoffeeVanInitializer initializator = new CoffeeVanInitializer();

this.cargo = initializator.getCargo();

}

public Coffee[] getCargo() {

return this.cargo;

}

public String findProduct(Coffee coffee) {

String result = "";

int index = Arrays.binarySearch(this.cargo, coffee);

if (index > 0) {

result = Resourcer.getString("message.foundItem");

} else {

result = Resourcer.getString("message.notfoundItem");

}

return result;

}

public void sortCargo() {

PriceToWeigthComparator comparator = new PriceToWeigthComparator();

Arrays.sort(this.cargo, comparator);

}

@Override

public String toString() {

String s = "";

for (int i = 0; i < this.cargo.length; i++) {

s += this.cargo[i].toString() + "\n";

}

return s;

}

}

## Класс PhysicalConditionCoffeeClass

package ru.rsreu.parshina0513.coffee;

import ru.rsreu.parshina0513.Resourcer;

public enum PhysicalConditionCoffeeClass {

CORN(Resourcer.getString("message.corn"), 1000) {

public double getKgPrice() {

return PhysicalConditionCoffeeClass.getBasePrice();

}

},

GROUND(Resourcer.getString("message.ground"), 500) {

public double getKgPrice() {

return PhysicalConditionCoffeeClass.getBasePrice()

\* PhysicalConditionCoffeeClass.GROUND\_CLASS\_MULTIPLICATOR;

}

},

INSTANT\_CANS(Resourcer.getString("message.instantCans"), 250) {

public double getKgPrice() {

return PhysicalConditionCoffeeClass.getBasePrice()

\* PhysicalConditionCoffeeClass.INSTANT\_CANS\_CLASS\_MULTIPLICATOR;

}

},

INSTANT\_BAGS(Resourcer.getString("message.instantBags"), 25) {

public double getKgPrice() {

return PhysicalConditionCoffeeClass.getBasePrice()

\* PhysicalConditionCoffeeClass.INSTANT\_BAGS\_CLASS\_MULTIPLICATOR;

}

};

public static final double GROUND\_CLASS\_MULTIPLICATOR = 0.5;

public static final double INSTANT\_CANS\_CLASS\_MULTIPLICATOR = 2.5;

public static final double INSTANT\_BAGS\_CLASS\_MULTIPLICATOR = 2;

private static double basePrice = 1;

private int weigthOnePackage;

private String name;

public int getWeigthOnePackage() {

return this.weigthOnePackage;

}

public String getName() {

return this.name;

}

public static double getBasePrice() {

return PhysicalConditionCoffeeClass.basePrice;

}

public static void setBasePrice(double basePrice) {

PhysicalConditionCoffeeClass.basePrice = basePrice;

}

PhysicalConditionCoffeeClass(String name, int wigthOnePackage) {

this.name = name;

this.weigthOnePackage = wigthOnePackage;

}

public abstract double getKgPrice();

}

## Класс Coffee.

package ru.rsreu.parshina0513.coffee;

import ru.rsreu.parshina0513.Resourcer;

public abstract class Coffee implements Comparable<Coffee> {

private static final double DENSITY\_COFFEE = 0.56;

private int volume;

private PhysicalConditionCoffeeClass physicalConditionCoffee;

public Coffee(int volume, PhysicalConditionCoffeeClass physicalConditionCoffee) {

this.setVolume(volume);

this.setPhysicalConditionCoffee(physicalConditionCoffee);

}

public final int getVolume() {

return this.volume;

}

public final void setVolume(int volume) {

this.volume = volume;

}

public final PhysicalConditionCoffeeClass getPhysicalConditionCoffee() {

return this.physicalConditionCoffee;

}

public final void setPhysicalConditionCoffee(PhysicalConditionCoffeeClass physicalConditionCoffee) {

this.physicalConditionCoffee = physicalConditionCoffee;

}

public int getWeigth() {

int weigthOnePackage = (int) (this.getVolume() \* Coffee.DENSITY\_COFFEE);

return weigthOnePackage;

}

public int getCountItemPackage() {

int weigthOnePackage = (int) (this.getWeigth() / this.getPhysicalConditionCoffee().getWeigthOnePackage());

return weigthOnePackage;

}

public double getPriceForWeigth(double priceKilogram) {

double weigth = this.getWeigth();

return priceKilogram \* weigth;

}

public double getPriceForVariety() {

return this.getPhysicalConditionCoffee().getKgPrice();

}

public abstract double getPriceCoffee();

public int getPriceToWeigth() {

double priceToWeigth = this.getPriceCoffee() / this.getWeigth();

int valuePriceToWeigth = (int) priceToWeigth;

return valuePriceToWeigth;

}

@Override

public int compareTo(Coffee coffee) {

return -(this.hashCode() - coffee.hashCode());

}

@Override

public int hashCode() {

return this.getVolume() + (int) this.getPhysicalConditionCoffee().getKgPrice()

+ (int) this.getPhysicalConditionCoffee().getWeigthOnePackage();

}

@Override

public boolean equals(Object obj) {

if (this == obj) {

return true;

}

if (obj == null) {

return false;

}

if (getClass() != obj.getClass()) {

return false;

}

Coffee other = (Coffee) obj;

if (this.volume != other.getVolume()) {

return false;

}

if (this.physicalConditionCoffee != other.getPhysicalConditionCoffee()) {

return false;

}

return true;

}

@Override

public String toString() {

return this.getPhysicalConditionCoffee().getName() + String.format(

Resourcer.getString("message.format"), this.getVolume(), this.getWeigth(), this.getPriceCoffee());

}

}

## Класс CoffeeVanInitializer

package ru.rsreu.parshina0513.coffeevan;

import ru.rsreu.parshina0513.coffee.Coffee;

import ru.rsreu.parshina0513.coffee.PhysicalConditionCoffeeClass;

import ru.rsreu.parshina0513.coffee.variety.ArabicaCoffee;

import ru.rsreu.parshina0513.coffee.variety.ExcelsaCoffee;

import ru.rsreu.parshina0513.coffee.variety.LibericaCoffee;

import ru.rsreu.parshina0513.coffee.variety.RobustaCoffee;

public class CoffeeVanInitializer {

public static final double VARIETY\_MULTIPLICATOR = 4;

public static final double PHYSICALCONDITION\_MULTIPLICATOR = 4;

public static final double ARABICA\_MULTIPLICATOR = 0.4;

public static final double ROBUSTA\_MULTIPLICATOR = 0.4;

public static final double EXCELSA\_MULTIPLICATOR = 0.1;

public static final double LIBERICA\_MULTIPLICATOR = 0.1;

public static final double BAGS\_MULTIPLICATOR = 0.25;

public static final double CANS\_MULTIPLICATOR = 0.25;

public static final double GROUND\_MULTIPLICATOR = 0.3;

public static final double CORN\_MULTIPLICATOR = 0.2;

public static final int SEA\_LEVEL = 250;

public static final int GRAIN\_LENGTH = 2;

public static final int TREE\_HEIGHT = 2;

public static final int COUNT\_GRAINS = 2000;

private Coffee[] cargo;

CoffeeVanInitializer() {

this.setCargo();

}

private int getVolumeArabica() {

int volume = (int) (CoffeeVan.VOLUME \* ARABICA\_MULTIPLICATOR);

return volume;

}

private int getVolumeRobusta() {

int volume = (int) (CoffeeVan.VOLUME \* ROBUSTA\_MULTIPLICATOR);

return volume;

}

private int getVolumeLiberica() {

int volume = (int) (CoffeeVan.VOLUME \* LIBERICA\_MULTIPLICATOR);

return volume;

}

private int getVolumeExcelsa() {

int volume = (int) (CoffeeVan.VOLUME \* EXCELSA\_MULTIPLICATOR);

return volume;

}

private int getVolumeCorn(int volumeCoffee) {

int volume = (int) (volumeCoffee \* CORN\_MULTIPLICATOR);

return volume;

}

private int getVolumeGround(int volumeCoffee) {

int volume = (int) (volumeCoffee \* GROUND\_MULTIPLICATOR);

return volume;

}

private int getVolumeInstantBags(int volumeCoffee) {

int volume = (int) (volumeCoffee \* BAGS\_MULTIPLICATOR);

return volume;

}

private int getVolumeInstantCans(int volumeCoffee) {

int volume = (int) (volumeCoffee \* CANS\_MULTIPLICATOR);

return volume;

}

public void setCargo() {

int countItem = (int) (PHYSICALCONDITION\_MULTIPLICATOR \* VARIETY\_MULTIPLICATOR);

cargo = new Coffee[countItem];

int weigthArabica = this.getVolumeArabica();

int weigthRobusta = this.getVolumeRobusta();

int weigthLiberica = this.getVolumeLiberica();

int weigthExcelsa = this.getVolumeExcelsa();

cargo[0] = new ArabicaCoffee(this.getVolumeCorn(weigthArabica), PhysicalConditionCoffeeClass.CORN, SEA\_LEVEL);

cargo[1] = new ArabicaCoffee(this.getVolumeGround(weigthArabica), PhysicalConditionCoffeeClass.GROUND,

SEA\_LEVEL);

cargo[2] = new ArabicaCoffee(this.getVolumeInstantBags(weigthArabica),

PhysicalConditionCoffeeClass.INSTANT\_BAGS, SEA\_LEVEL);

cargo[3] = new ArabicaCoffee(this.getVolumeInstantCans(weigthArabica),

PhysicalConditionCoffeeClass.INSTANT\_CANS, SEA\_LEVEL);

cargo[4] = new RobustaCoffee(this.getVolumeCorn(weigthRobusta), PhysicalConditionCoffeeClass.CORN,

GRAIN\_LENGTH);

cargo[5] = new RobustaCoffee(this.getVolumeGround(weigthRobusta), PhysicalConditionCoffeeClass.GROUND,

GRAIN\_LENGTH);

cargo[6] = new RobustaCoffee(this.getVolumeInstantBags(weigthRobusta),

PhysicalConditionCoffeeClass.INSTANT\_BAGS, GRAIN\_LENGTH);

cargo[7] = new RobustaCoffee(this.getVolumeInstantCans(weigthRobusta),

PhysicalConditionCoffeeClass.INSTANT\_CANS, GRAIN\_LENGTH);

cargo[8] = new LibericaCoffee(this.getVolumeCorn(weigthLiberica), PhysicalConditionCoffeeClass.CORN,

COUNT\_GRAINS);

cargo[9] = new LibericaCoffee(this.getVolumeGround(weigthLiberica), PhysicalConditionCoffeeClass.GROUND,

COUNT\_GRAINS);

cargo[10] = new LibericaCoffee(this.getVolumeInstantBags(weigthLiberica),

PhysicalConditionCoffeeClass.INSTANT\_BAGS, COUNT\_GRAINS);

cargo[11] = new LibericaCoffee(this.getVolumeInstantCans(weigthLiberica),

PhysicalConditionCoffeeClass.INSTANT\_CANS, COUNT\_GRAINS);

cargo[12] = new ExcelsaCoffee(this.getVolumeCorn(weigthExcelsa), PhysicalConditionCoffeeClass.CORN,

TREE\_HEIGHT);

cargo[13] = new ExcelsaCoffee(this.getVolumeGround(weigthExcelsa), PhysicalConditionCoffeeClass.GROUND,

TREE\_HEIGHT);

cargo[14] = new ExcelsaCoffee(this.getVolumeInstantBags(weigthExcelsa),

PhysicalConditionCoffeeClass.INSTANT\_BAGS, TREE\_HEIGHT);

cargo[15] = new ExcelsaCoffee(this.getVolumeInstantCans(weigthExcelsa),

PhysicalConditionCoffeeClass.INSTANT\_CANS, TREE\_HEIGHT);

}

public Coffee[] getCargo() {

return this.cargo;

}

}