**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «Электронные вычислительные машины»

**Отчет о лабораторной работе № 6**

по теме

«Внутренние классы»

**Выполнили:**

Студенты гр. 045

Бригады №10

Анохин В.А.

Вашкулатов Н.А.

**Проверил:**

Асс. Бастрычкин А.С.

Цель работы:изучение внутренних классов, приобретение навыков использования inner классов в Java-программах.

Ход работы

Создать класс Cправочная Cлужба Oбщественного Tранспорта с внутренним классом, с помощью объектов которого можно хранить информацию о времени, линиях маршрутов и стоимости проезда.

Код программы:

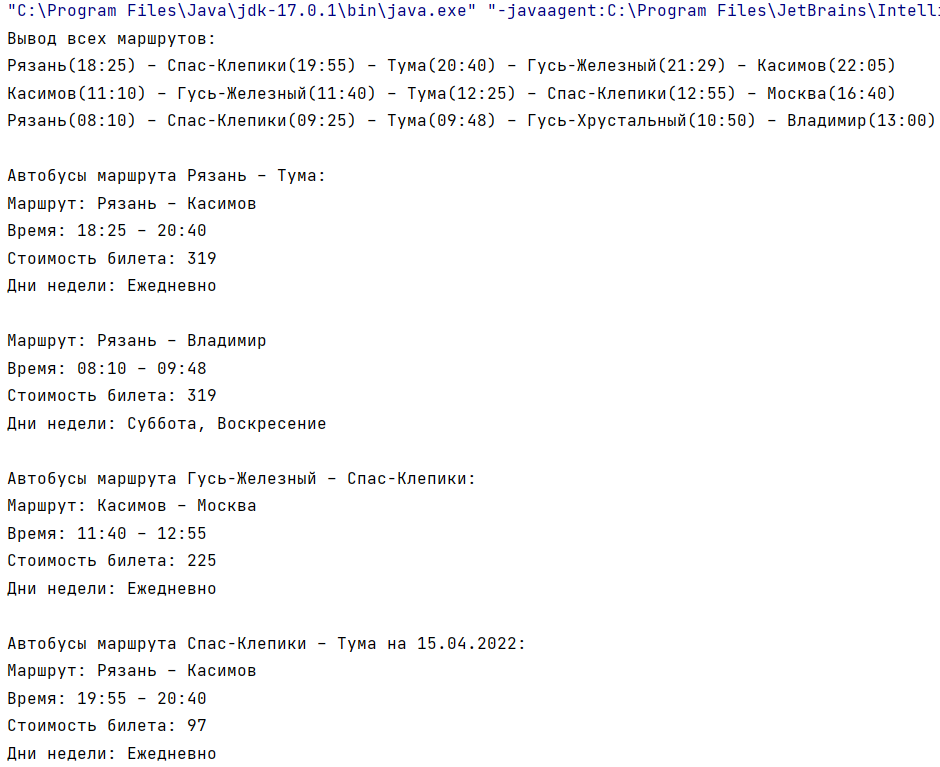
Класс TransportRefService с вложенными классами TransInfo, TransInfoBuilder и записью ArrivalPoint

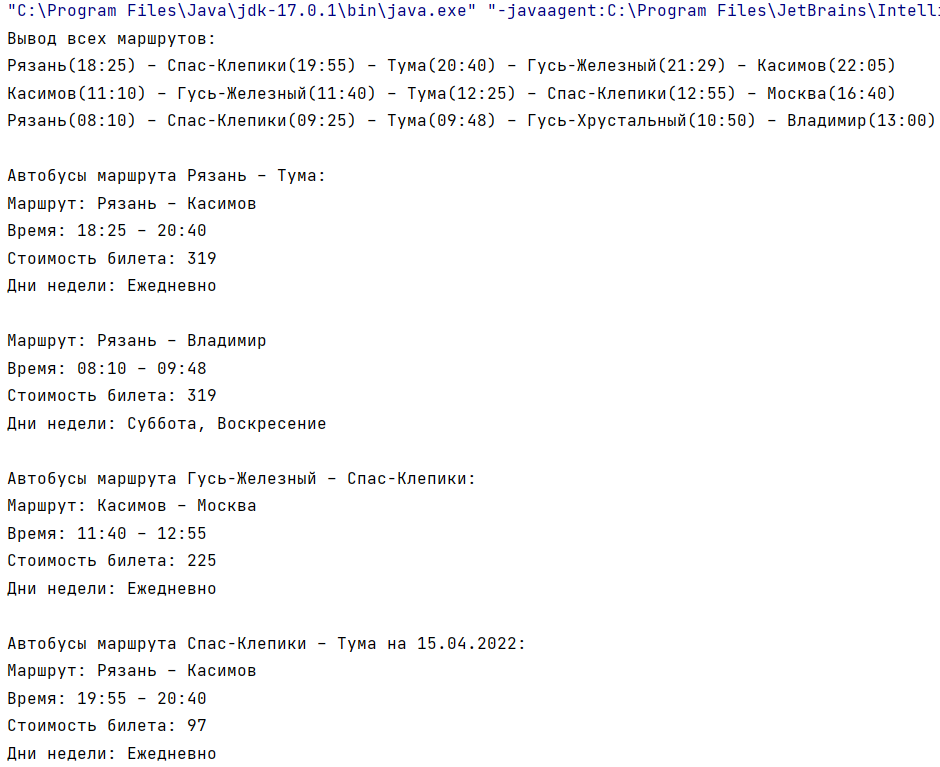
import java.time.DayOfWeek;  
import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.stream.Collectors;  
  
public class TransportRefService {  
 private ArrayList<TransInfo> buses;  
 private String startPoint, endPoint;  
  
 TransportRefService(){  
 buses = new ArrayList<>();  
 }  
  
 void addBus(TransInfo bus){  
 buses.add(bus);  
 }  
  
 TransInfoBuilder buildTransInfo(String startPoint, LocalTime sendTime){  
 return new TransInfoBuilder(startPoint, sendTime);  
 }  
  
 void printBus(String startPoint, String endPoint){  
 this.startPoint = startPoint;  
 this.endPoint = endPoint;  
 buses.stream()  
 .filter(TransInfo::existPath)  
 .forEach(this::printInfo);  
 }  
  
 void printBus(String startPoint, String endPoint, LocalDate date){  
 this.startPoint = startPoint;  
 this.endPoint = endPoint;  
 buses.stream()  
 .filter(TransInfo::existPath)  
 .filter(bus -> bus.isAvailable(date.getDayOfWeek()))  
 .forEach(this::printInfo);  
 }  
  
 void printAllPaths(){  
 buses.forEach(System.out::println);  
 }  
  
 void printInfo(TransInfo bus){  
 System.out.println("Маршрут: " + bus.getExtremePoints());  
 System.out.println("Время: " + bus.getTimes());  
 System.out.println("Стоимость билета: " + bus.getCost());  
 System.out.println("Дни недели: " + bus.getWeekdays());  
 System.out.println();  
 }  
  
 private record ArrivalPoint(String point, LocalTime time, int cost){  
 @Override  
 public String toString() {  
 return point + "(" + time + ")";  
 }  
 }  
  
 class TransInfo {  
 private ArrayList<DayOfWeek> weekdays;  
 private ArrayList<ArrivalPoint> path;  
 int start, end;  
  
 TransInfo(ArrayList<DayOfWeek> weekdays, ArrayList<ArrivalPoint> path){  
 this.weekdays = weekdays;  
 this.path = path;  
 }  
  
 private int findPoint(String pointName){  
 return path.indexOf(path.stream()  
 .filter(p -> p.point.equals(pointName))  
 .findAny()  
 .orElse(null));  
 }  
  
 void update(){  
 start = findPoint(startPoint);  
 end = findPoint(endPoint);  
 }  
  
 boolean isAvailable(DayOfWeek weekday){  
 update();  
 return weekdays.contains(weekday);  
 }  
  
 boolean existPath(){  
 update();  
 return (start<end)&&(start!=-1)&&(end!=-1);  
 }  
  
 int getCost(){  
 int cost = 0;  
 for(int i = start + 1; i<=end; ++i){  
 cost += path.get(i).cost;  
 }  
 return cost;  
 }  
  
 String getTimes(){  
 return path.get(start).time + " – " + path.get(end).time;  
 }  
  
 String getExtremePoints(){  
 return path.get(0).point + " – " + path.get(path.size()-1).point;  
 }  
  
 String getWeekdays(){  
 ArrayList<String> days = new ArrayList<>();  
 for (DayOfWeek day : weekdays){  
 days.add(switch (day){  
 case MONDAY -> "Понедельник";  
 case TUESDAY -> "Вторник";  
 case WEDNESDAY -> "Среда";  
 case THURSDAY -> "Четверг";  
 case FRIDAY -> "Пятница";  
 case SATURDAY -> "Суббота";  
 case SUNDAY -> "Воскресение";  
 });  
 }  
 if(days.size()==7) {  
 return "Ежедневно";  
 }  
 return String.join(", ", days);  
 }  
  
 @Override  
 public String toString() {  
 return path.stream()  
 .map(ArrivalPoint::toString)  
 .collect(Collectors.joining(" – "));  
 }  
 }  
  
 class TransInfoBuilder {  
 private ArrayList<DayOfWeek> weekdays;  
 private ArrayList<ArrivalPoint> path;  
  
 private TransInfoBuilder(String startPoint, LocalTime sendTime){  
 weekdays = new ArrayList<>();  
 path = new ArrayList<>();  
 path.add(new ArrivalPoint(startPoint, sendTime, 0));  
 }  
  
 TransInfoBuilder addToPath(String point, LocalTime time, int cost){  
 path.add(new ArrivalPoint(point, time, cost));  
 return this;  
 }  
  
 TransInfoBuilder setWeekDays(DayOfWeek... days){  
 Collections.addAll(weekdays, days);  
 return this;  
 }  
  
 TransInfo apply(){  
 return new TransInfo(weekdays, path);  
 }  
 }  
}

Основная программа

import java.time.DayOfWeek;  
import java.time.LocalDate;  
import java.time.LocalTime;  
  
public class App {  
 public static void main(String[] args) {  
 TransportRefService service = getService();  
 System.out.println("Вывод всех маршрутов:");  
 service.printAllPaths();  
 System.out.println();  
 System.out.println("Автобусы маршрута Рязань – Тума:");  
 service.printBus("Рязань", "Тума");  
 System.out.println("Автобусы маршрута Гусь-Железный – Спас-Клепики:");  
 service.printBus("Гусь-Железный", "Спас-Клепики");  
 System.out.println("Автобусы маршрута Спас-Клепики – Тума на 15.04.2022:");  
 service.printBus("Спас-Клепики", "Тума", LocalDate.of(2022, 4, 22));  
 }  
  
 static TransportRefService getService(){  
 TransportRefService service = new TransportRefService();  
 service.addBus(service.buildTransInfo("Рязань", LocalTime.of(18, 25))  
 .addToPath("Спас-Клепики", LocalTime.of(19, 55), 222)  
 .addToPath("Тума", LocalTime.of(20, 40), 97)  
 .addToPath("Гусь-Железный", LocalTime.of(21, 29), 128)  
 .addToPath("Касимов", LocalTime.of(22, 5), 64)  
 .setWeekDays(DayOfWeek.values())  
 .apply());  
 service.addBus(service.buildTransInfo("Касимов", LocalTime.of(11, 10))  
 .addToPath("Гусь-Железный", LocalTime.of(11, 40), 64)  
 .addToPath("Тума", LocalTime.of(12, 25), 128)  
 .addToPath("Спас-Клепики", LocalTime.of(12, 55), 97)  
 .addToPath("Москва", LocalTime.of(16, 40), 760)  
 .setWeekDays(DayOfWeek.values())  
 .apply());  
 service.addBus(service.buildTransInfo("Рязань", LocalTime.of(8, 10))  
 .addToPath("Спас-Клепики", LocalTime.of(9, 25), 222)  
 .addToPath("Тума", LocalTime.of(9, 48), 97)  
 .addToPath("Гусь-Хрустальный", LocalTime.of(10, 50), 170)  
 .addToPath("Владимир", LocalTime.of(13, 0), 200)  
 .setWeekDays(DayOfWeek.SATURDAY, DayOfWeek.SUNDAY)  
 .apply());  
 return service;  
 }  
}

Окно вывода:





**Вывод:** мы изучили внутренние классы, приобрели навыки использования inner классов в Java-программах.