МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ИИТ

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине СПП

Тема: «Классы в программах на языке программирования Java»

                                                                       Выполнил

студент группы ПО-5

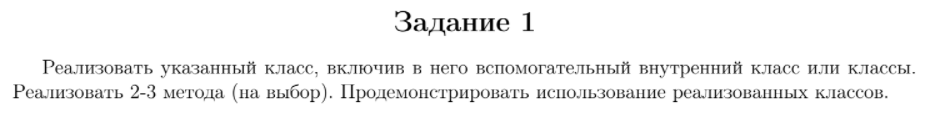
                                                                       Романюк В. А.

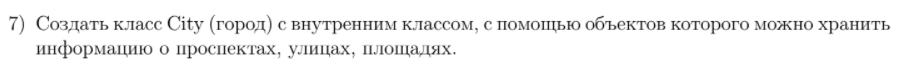
                                                                       Проверил: преподаватель

                                                                       Крощенко А. А.

Брест, 2021

Цель работы: приобрести практические навыки в области объектно-ориентированного проектирования.

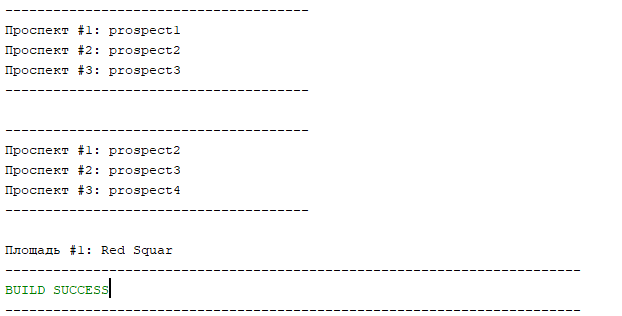


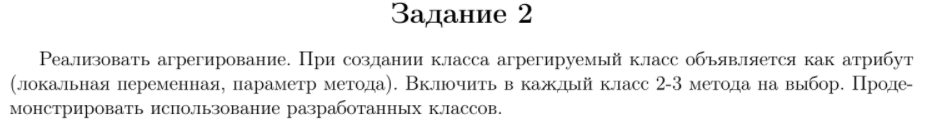


Код программы

package com.company;  
  
import java.util.\*;  
  
public class Lab4Task1 {  
 public static void main(String[] args) {  
 City test = new City(new String[] {"prospect1", "prospect2", "prospect3"},  
 new String[] {"street1", "street2"},  
 new String[] {"area1"});  
  
 test.printProspects();  
  
 test.addProspect("prospect4");  
 test.removeProspect("prospect1");  
  
 test.printProspects();  
  
 test.setAreas(new String[] {"Red Squar"});  
  
 test.printAreas();  
 }  
}  
  
class City {  
 public class CityChild {  
 private ArrayList<String> prospects;  
 private ArrayList<String> streets;  
 private ArrayList<String> areas;  
  
 public CityChild(String[] prospects, String[] streets, String[] areas) {  
 this.prospects = new ArrayList<>(Arrays.asList(prospects));  
 this.streets = new ArrayList<>(Arrays.asList(streets));  
 this.areas = new ArrayList<>(Arrays.asList(areas));  
 }  
 }  
  
 private CityChild obj;  
  
 public City(String[] prospects, String[] streets, String[] areas) {  
 obj = new CityChild(prospects, streets, areas);  
 }  
  
 public void printProspects() {  
 int step = 1;  
  
 System.out.println("--------------------------------------");  
  
 for (String prospect : obj.prospects) {  
 System.out.println("Проспект #" + step + ": " + prospect);  
 step++;  
 }  
  
 System.out.println("--------------------------------------\n");  
 }  
  
 public void printStreets() {  
 int step = 1;  
  
 for (String street : obj.streets) {  
 System.out.println("Улица #" + step + ": " + street);  
 step++;  
 }  
 }  
  
 public void printAreas() {  
 int step = 1;  
  
 for (String area : obj.areas) {  
 System.out.println("Площадь #" + step + ": " + area);  
 step++;  
 }  
 }  
  
 public void addProspect(String prospect) {  
 obj.prospects.add(prospect);  
 }  
  
 public void addStreet(String street) {  
 obj.streets.add(street);  
 }  
  
 public void addArea(String area) {  
 obj.areas.add(area);  
 }  
  
 public void removeProspect(String prospect) {  
 obj.prospects.remove(prospect);  
 }  
  
 public void removeStreet(String street) {  
 obj.streets.remove(street);  
 }  
  
 public void removeArea(String area) {  
 obj.areas.remove(area);  
 }  
  
 public void setProspects(String[] prospects) {  
 obj.prospects = new ArrayList<>(Arrays.asList(prospects));  
 }  
  
 public void setStreets(String[] streets) {  
 obj.streets = new ArrayList<>(Arrays.asList(streets));  
 }  
  
 public void setAreas(String[] areas) {  
 obj.areas = new ArrayList<>(Arrays.asList(areas));  
 }  
}

Тестирование



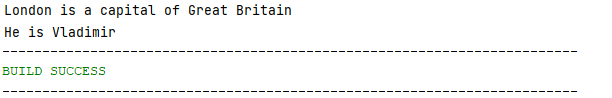


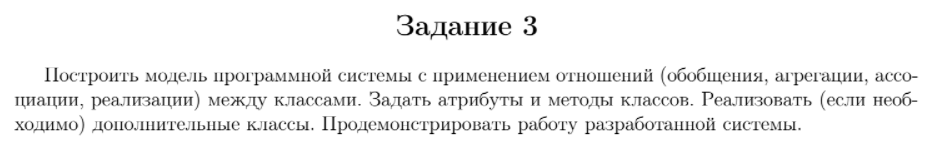


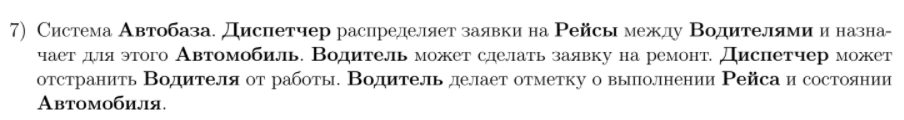
Код программы

package com.company;  
  
import java.util.\*;  
  
public class Lab4Task2 {  
 public static void main(String[] args) {  
 Word london\_ = new Word("London");  
 Word is\_ = new Word("is");  
 Word a\_ = new Word("a");  
 Word capital\_ = new Word("capital");  
 Word of\_ = new Word("of");  
 Word great\_ = new Word("Great");  
 Word britain\_ = new Word("Britain");  
  
 Word he\_ = new Word("He");  
 Word artyom\_ = new Word("Vladimir");  
  
 Line liacogb\_\_ = new Line(new Word[] {london\_, is\_, a\_, capital\_, of\_, great\_, britain\_});  
 Line hia\_\_ = new Line(new Word[] {he\_, is\_, artyom\_});  
  
 Page liacogb\_hia\_\_\_ = new Page(new Line[] {liacogb\_\_, hia\_\_});  
  
 liacogb\_hia\_\_\_.printPage();  
 }  
}  
  
class Word {  
 String word;  
  
 public Word(String word) {  
 this.word = word;  
 }  
  
 public void printWord() {  
 System.out.println(this.word);  
 }  
}  
  
class Line {  
 String line;  
  
 public Line(Word[] words) {  
 line = "";  
  
 for (Word word : words) {  
 line += word.word;  
 line += " ";  
 }  
  
 line = line.trim();  
 line += ".";  
 }  
  
 public void printLine() {  
 System.out.println(this.line);  
 }  
}  
  
class Page {  
 String page;  
  
 public Page(Line[] lines) {  
 page = "";  
  
 for (Line line : lines) {  
 page += line.line;  
 page += "\n";  
 }  
  
 page = page.trim();  
 }  
  
 public void printPage() {  
 System.out.println(this.page);  
 }  
}

Тестирование



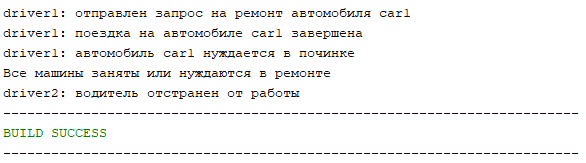




Код программы

package com.company;  
  
import java.util.\*;  
  
public class Lab4Task3 {  
 public static void main(String[] args) {  
 Trip trip1 = new Trip("trip1");  
 Trip trip2 = new Trip("trip2");  
 Trip trip3 = new Trip("trip3");  
 Trip trip4 = new Trip("trip4");  
 Trip trip5 = new Trip("trip5");  
 Trip trip6 = new Trip("trip6");  
  
 Driver driver1 = new Driver("driver1");  
 Driver driver2 = new Driver("driver2");  
 Driver driver3 = new Driver("driver3");  
 Driver driver4 = new Driver("driver4");  
 Driver driver5 = new Driver("driver5");  
  
 Car car1 = new Car("car1");  
 Car car2 = new Car("car2");  
 Car car3 = new Car("car3");  
 Car car4 = new Car("car4");  
 Car car5 = new Car("car5");  
  
 Dispatcher dispatcher = new Dispatcher(new Trip[] {trip1, trip2, trip3, trip4, trip5},  
 new Driver[] {driver1, driver2, driver3, driver4, driver5},  
 new Car[] {car1, car2, car3, car4, car5});  
  
 dispatcher.scheduleTrips();  
 driver1.repairRequest();  
 driver1.finishTrip();  
 dispatcher.scheduleTrips();  
 driver1.finishTrip();  
 dispatcher.removeDriver(driver2);  
 driver2.report();  
 }  
}  
  
interface Interface {  
 void repairRequest();  
}  
  
class Dispatcher {  
 ArrayList<Trip> trips;  
 ArrayList<Driver> drivers;  
 ArrayList<Car> cars;  
  
 Dispatcher(Trip[] trips, Driver[] drivers, Car[] cars) {  
 this.trips = new ArrayList<>(Arrays.asList(trips));  
 this.drivers = new ArrayList<>(Arrays.asList(drivers));  
 this.cars = new ArrayList<>(Arrays.asList(cars));  
 }  
  
 public void addTrip(Trip trip) {  
 trips.add(trip);  
 }  
  
 public void addDriver(Driver driver) {  
 drivers.add(driver);  
 }  
  
 public void addCar(Car car) {  
 cars.add(car);  
 }  
  
 public void scheduleTrips() {  
 point1:  
 for (Trip trip : trips) {  
 point2:  
 if (!trip.scheduled) {  
 for (Driver driver : drivers) {  
 if (!driver.tripStarted && !driver.removed) {  
 for (Car car : cars) {  
 if (!car.taken && !car.needRepair) {  
 trip.scheduled = true;  
 driver.tripStarted = true;  
 car.taken = true;  
 driver.car = car;  
 driver.trip = trip;  
 break point2;  
 }  
 }  
  
 System.out.println("Все машины заняты или нуждаются в ремонте");  
 break point1;  
 }  
 }  
  
 System.out.println("Все водители заняты или отстранены от работы");  
 }  
 }  
 }  
  
 public void removeDriver(Driver driver) {  
 driver.removed = true;  
 driver.tripStarted = false;  
 driver.tripFinished = false;  
 driver.car.taken = false;  
 driver.trip.scheduled = false;  
 driver.car = null;  
 driver.trip = null;  
 }  
}  
  
class Trip {  
 protected String name;  
 protected boolean scheduled;  
  
 Trip(String name) {  
 this.name = name;  
 }  
}  
  
class Driver implements Interface {  
 protected String name;  
 protected Car car;  
 protected Trip trip;  
 protected boolean tripStarted;  
 protected boolean tripFinished;  
 protected boolean removed;  
  
 Driver(String name) {  
 this.name = name;  
 tripStarted = false;  
 tripFinished = false;  
 removed = false;  
 car = null;  
 trip = null;  
 }  
  
 @Override  
 public void repairRequest() {  
 car.needRepair = true;  
 System.out.println(name + ": отправлен запрос на ремонт автомобиля " + car.name);  
 }  
  
 public void finishTrip() {  
 if (tripStarted && !tripFinished) {  
 tripStarted = false;  
 tripFinished = true;  
 report();  
 car.taken = false;  
 trip.scheduled = false;  
 tripFinished = false;  
 car = null;  
 trip = null;  
 }  
 }  
  
 public void report() {  
 if (removed) {  
 System.out.println(name + ": водитель отстранен от работы");  
 }  
 else {  
 if (tripStarted) {  
 System.out.println(name + ": поездка на автомобиле " + car.name + " не завершена");  
 }  
 else if (!tripStarted && tripFinished) {  
 System.out.println(name + ": поездка на автомобиле " + car.name + " завершена");  
 }  
 else {  
 System.out.println(name + ": поездка на автомобиле не назначена");  
 }  
  
 if (car != null) {  
 if (car.needRepair) {  
 System.out.println(name + ": автомобиль " + car.name + " нуждается в починке");  
 }  
 else {  
 System.out.println(name + ": автомобиль " + car.name + " не нуждается в починке");  
 }  
 }  
 }  
 }  
}  
  
class Car {  
 protected String name;  
 protected boolean needRepair;  
 protected boolean taken;  
  
 Car(String name) {  
 this.name = name;  
 needRepair = false;  
 taken = false;  
 }  
}

Тестирование



Вывод: приобрел практические навыки в области объектно-ориентированного проектирования.