**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

по дисциплине

«Введение в разработка программного обеспечения»

**Развитие представлений о разработке программ. Объектно-ориентированное программирование в Java**

Выполнил студент группы ИТП-21

Дешев А. П.

Проверил преподаватель

Башаримов Ю. С.

Гомель 2024

Цель работы: исследовать развитие представлений о разработке

программного обеспечения, сосредоточив внимание на объектно-ориентированном программировании (ООП) в *Java*.

**Задание.**

Подготовить отчет, который должен содержать:

* Название работы.
* Цель работы.
* Условие задачи.
* Разработать *UML*‒диаграмму.
* Листинг программы.
* Вывод.

**Ход работы**

**Задание:**

Составить и блок‒схему алгоритма и программу для вычисления значений переменных в соответствии с условием. Вариант задания приведён в таблице 1. Результат выполнения задания представлен на рисунке 1, *UML*‒диаграмма на рисунке 2. Код программы представлен в приложении А.

Таблица 1 – Условие к заданию (Вариант 7)

|  |  |
| --- | --- |
| Вариант | Условие задачи |
| 7 | 1.1 Создать иерархию классов для учёта самолетов в аэропорту.  1.2 Создать 3 класса самолетов и минимум 3 экземпляра. 1.3 Классы должны содержать поля: количество пассажиров, количество топлива, название рейса.  1.4 Рассчитать расход топлива.  1.5 Вывести все самолеты, которые содержать количество пассажиров меньше заданного.  1.6 Вывести все самолеты, название которых начинается с букву заданную букву. |

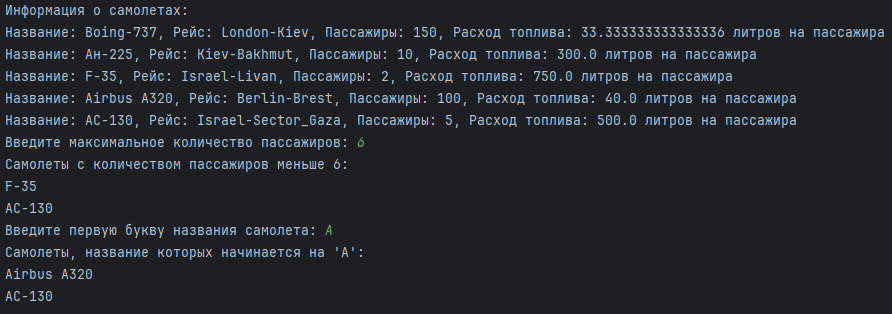




Рисунок 1 ‒ Результат выполнения задания



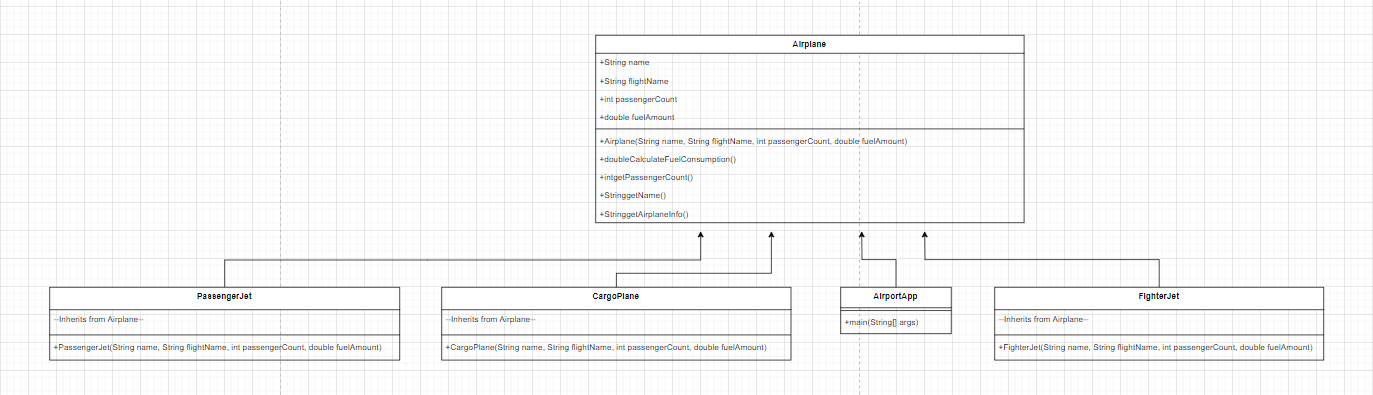


Рисунок 2 ‒ *UML*‒диаграмма

Вывод: в результате выполнения лабораторной работы изучили развитие

представлений о разработке программного обеспечения, сосредоточив внимание на объектно-ориентированном программировании (ООП) в *Java*.

**ПРИЛОЖЕНИЕ А**

**Текст программы**

**AirportApp.java:**

package com.airport;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
public class AirportApp {  
  
 public static void main(String[] args) {  
 // Список самолетов  
 List<Airplane> airplanes = new ArrayList<>();  
  
 // Добавление самолетов с фиксированными параметрами  
 airplanes.add(new PassengerJet("Boing-737", "London-Kiev", 150, 5000));  
 airplanes.add(new CargoPlane("Ан-225", "Kiev-Bakhmut", 10, 3000));  
 airplanes.add(new FighterJet("F-35", "Israel-Livan", 2, 1500));  
 airplanes.add(new PassengerJet("Airbus A320", "Berlin-Brest", 100, 4000));  
 airplanes.add(new FighterJet("AC-130", "Israel-Sector\_Gaza", 5, 2500));  
  
 // Вывод всей информации о самолетах  
 System.*out*.println("Информация о самолетах:");  
 for (Airplane airplane : airplanes) {  
 System.*out*.println(airplane.getAirplaneInfo());  
 }  
  
 // Запрос количества пассажиров  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Введите максимальное количество пассажиров: ");  
 int maxPassengers = scanner.nextInt();  
  
 System.*out*.println("Самолеты с количеством пассажиров меньше " + maxPassengers + ":");  
 for (Airplane airplane : airplanes) {  
 if (airplane.getPassengerCount() < maxPassengers) {  
 System.*out*.println(airplane.getName());  
 }  
 }  
  
 // Запрос буквы для фильтрации по названию  
 System.*out*.print("Введите первую букву названия самолета: ");  
 char initial = scanner.next().charAt(0);  
  
 System.*out*.println("Самолеты, название которых начинается на '" + initial + "':");  
 for (Airplane airplane : airplanes) {  
 if (airplane.getName().charAt(0) == initial) {  
 System.*out*.println(airplane.getName());  
 }  
 }  
  
 scanner.close();  
 }  
}

**Airplane.java:**

package com.airport;  
  
public abstract class Airplane {  
 public String name; // Название самолета  
 public String flightName; // Рейс  
 public int passengerCount; // Количество пассажиров  
 public double fuelAmount; // Количество топлива  
  
 public Airplane(String name, String flightName, int passengerCount, double fuelAmount) {  
 this.name = name;  
 this.flightName = flightName;  
 this.passengerCount = passengerCount;  
 this.fuelAmount = fuelAmount;  
 }  
  
 public double calculateFuelConsumption() {  
 return fuelAmount / passengerCount;  
 }  
  
 public int getPassengerCount() {  
 return passengerCount;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public String getAirplaneInfo() {  
 return "Название: " + name + ", Рейс: " + flightName + ", Пассажиры: " + passengerCount +  
 ", Расход топлива: " + calculateFuelConsumption() + " литров на пассажира";  
 }  
}

**PassengerJet.java:**

package com.airport;  
  
public class PassengerJet extends Airplane {  
 public PassengerJet(String name, String flightName, int passengerCount, double fuelAmount) {  
 super(name, flightName, passengerCount, fuelAmount);  
 }  
}

**CargoPlane.java:**

package com.airport;  
  
public class CargoPlane extends Airplane {  
 public CargoPlane(String name, String flightName, int passengerCount, double fuelAmount) {  
 super(name, flightName, passengerCount, fuelAmount);  
 }  
}

**FighterJet.java:**

package com.airport;  
  
public class FighterJet extends Airplane {  
 public FighterJet(String name, String flightName, int passengerCount, double fuelAmount) {  
 super(name, flightName, passengerCount, fuelAmount);  
 }  
}