Министерство образования и науки РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа «Компьютерных технологий и информационных систем»

ОТЧЕТ

по дисциплине «Практика по программированию»

**Лабораторная работа № 1**

**Выполнил:**

Cтудент гр. 5130902/30003 Садчев Р.В

**Проверил**

Ст. преподаватель А.М. Журавская

Санкт-Петербург

2021 г.

## **Задание**

Часть 1:

## 1. Выбрать предметную область согласно варианту индивидуальногозадания.

## 2. Спроектировать класс для выбранной предметной области.

## 3. Нарисовать диаграмму спроектированного класса.

## 4. Предусмотреть наличие у объекта полей, методов и свойств

## 5. Предусмотреть наличие свойств только для записи.

Часть 2:  
1. Реализовать тот же класс, что был в первой части на языке Python в colab.google.

1. Разместить код либо из 1 части, либо из второй на https://github.com/ (можно и тот и другой).

Вариант 14(4)

## **Код программы/Листинг программы**

**Часть 1:**

#include <iostream>

#include <string>

#include <cstdlib> // для system("cls") / system("clear")

#include <limits>

using namespace std;

class Airport {

private:

string name; // Название аэропорта

double ticketPrice; // Стоимость билета

int totalSeats; // Общее количество мест во всех самолетах

int soldTickets; // Количество проданных билетов

public:

Airport(const string& airportName, double price, int seats)

: name(airportName), ticketPrice(price), totalSeats(seats), soldTickets(0) {}

void sellTickets(int number) {

if (number <= (totalSeats - soldTickets)) {

soldTickets += number;

cout << "Sold " << number << " tickets." << endl;

}

else {

cout << "There are not enough available places for sale " << number << " tickets." << endl;

}

}

double calculateTotalRevenue() const {

return soldTickets \* ticketPrice;

}

void displayInfo() const {

cout << "Airport: " << name << endl;

cout << "Ticket cost: " << ticketPrice << endl;

cout << "Total number of seats: " << totalSeats << endl;

cout << "Tickets sold: " << soldTickets << endl;

cout << "Total cost of tickets sold: " << calculateTotalRevenue() << endl;

}

};

void clearConsole() {

#ifdef \_WIN32

system("cls"); // Очистка консоли для Windows

#else

system("clear"); // Очистка консоли для Linux/Unix

#endif

}

bool isValidNumber(const string& str) {

if (str.empty()) return false;

for (char c : str) {

if (!isdigit(c)) return false;

}

return true;

}

// Функция для безопасного ввода числа с проверкой на некорректный ввод

template <typename T>

T getValidInput(const string& prompt) {

string input;

while (true) {

cout << prompt;

cin >> input;

if (isValidNumber(input)) {

try {

if constexpr (is\_same<T, int>::value) {

return stoi(input);

}

else if constexpr (is\_same<T, double>::value) {

return stod(input);

}

}

catch (const invalid\_argument& e) {

cout << "Invalid input. Please enter a valid number." << endl;

}

}

else {

cout << "Invalid input. Please enter only numbers." << endl;

}

}

}

int main() {

Airport\* airport = nullptr;

bool OnDisplay = true;

short int choice;

while (OnDisplay) {

clearConsole();

cout << "Database Administrator Console:\n";

cout << "1. To add a new airport to the database.\n";

cout << "2. Current information about the airport.\n";

cout << "3. Calculating total revenue.\n";

cout << "4. Sell tickets.\n";

cout << "5. Exit.\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1: {

clearConsole();

string airportname;

double airportprice;

int airportseats;

cout << "Enter the name of the airport: ";

cin >> airportname;

airportprice = getValidInput<double>("Enter the ticket price: ");

airportseats = getValidInput<int>("Enter the total number of seats: ");

// Создаем новый аэропорт

airport = new Airport(airportname, airportprice, airportseats);

cout << "Airport added successfully!" << endl;

break;

}

case 2: {

clearConsole();

if (airport) {

airport->displayInfo();

}

else {

cout << "No airport data available. Please add an airport first." << endl;

}

break;

}

case 3: {

clearConsole();

if (airport) {

cout << "Total revenue: " << airport->calculateTotalRevenue() << endl;

}

else {

cout << "No airport data available. Please add an airport first." << endl;

}

break;

}

case 4: {

clearConsole();

if (airport) {

int ticketsToSell = getValidInput<int>("Enter the number of tickets to sell: "); // безопасный ввод

airport->sellTickets(ticketsToSell);

}

else {

cout << "No airport data available. Please add an airport first." << endl;

}

break;

}

case 5: {

clearConsole();

cout << "Exiting the program..." << endl;

OnDisplay = false;

break;

}

default: {

clearConsole();

cout << "Invalid choice. Please try again." << endl;

}

}

if (OnDisplay) {

cout << "Press Enter to continue...";

cin.ignore(); // Игнорируем символ новой строки после ввода числа

cin.get(); // Ожидаем нажатия клавиши

}

}

if (airport) {

delete airport;

}

return 0;

}

Часть 2:

import os

​

class Airport:

def \_\_init\_\_(self, name, ticket\_price, total\_seats):

self.name = name

self.ticket\_price = ticket\_price

self.total\_seats = total\_seats

self.sold\_tickets = 0

​

def sell\_tickets(self, number):

if number <= (self.total\_seats - self.sold\_tickets):

self.sold\_tickets += number

print(f"Sold {number} tickets.")

else:

print(f"There are not enough available places for sale {number} tickets.")

​

def calculate\_total\_revenue(self):

return self.sold\_tickets \* self.ticket\_price

​

def display\_info(self):

print(f"Airport: {self.name}")

print(f"Ticket cost: {self.ticket\_price}")

print(f"Total number of seats: {self.total\_seats}")

print(f"Tickets sold: {self.sold\_tickets}")

print(f"Total revenue: {self.calculate\_total\_revenue()}")

​

def clear\_console():

os.system('cls' if os.name == 'nt' else 'clear')

​

def is\_valid\_number(input\_str):

return input\_str.isdigit()

​

def get\_valid\_input(prompt, input\_type=int):

while True:

input\_str = input(prompt)

if is\_valid\_number(input\_str):

try:

return input\_type(input\_str)

except ValueError:

print("Invalid input. Please enter a valid number.")

else:

print("Invalid input. Please enter only numbers.")

​

def main():

airport = None

on\_display = True

​

while on\_display:

clear\_console()

print("Database Administrator Console:")

print("1. To add a new airport to the database.")

print("2. Current information about the airport.")

print("3. Calculating total revenue.")

print("4. Sell tickets.")

print("5. Exit.")

​

choice = get\_valid\_input("Enter your choice: ", int)

​

if choice == 1:

clear\_console()

airport\_name = input("Enter the name of the airport: ")

airport\_price = get\_valid\_input("Enter the ticket price: ", float)

airport\_seats = get\_valid\_input("Enter the total number of seats: ", int)

airport = Airport(airport\_name, airport\_price, airport\_seats)

print("Airport added successfully!")

elif choice == 2:

clear\_console()

if airport:

airport.display\_info()

else:

print("No airport data available. Please add an airport first.")

elif choice == 3:

clear\_console()

if airport:

print(f"Total revenue: {airport.calculate\_total\_revenue()}")

else:

print("No airport data available. Please add an airport first.")

elif choice == 4:

clear\_console()

if airport:

tickets\_to\_sell = get\_valid\_input("Enter the number of tickets to sell: ", int)

airport.sell\_tickets(tickets\_to\_sell)

else:

print("No airport data available. Please add an airport first.")

elif choice == 5:

clear\_console()

print("Exiting the program...")

on\_display = False

else:

clear\_console()

print("Invalid choice. Please try again.")

​

if on\_display:

input("Press Enter to continue...")

​

main()

**Пример работы программы**

Часть 1:

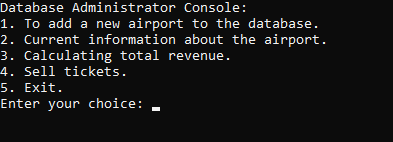


Рисунок 1 – Правильная работа программы

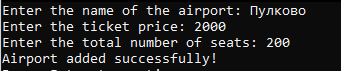


Рисунок 2 –Правильная работа программы

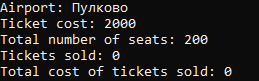


Рисунок 3 –Правильная работа программы



Рисунок 4 – Правильная работа программы



Рисунок 5 – Правильная работа программы

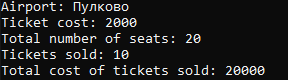


Рисунок 6 – Правильная работа программы



Рисунок 7 – Правильная работа программы при вводе неверных данных

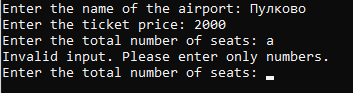


Рисунок 8 – Правильная работа программы при вводе неверных данных

Часть 2:

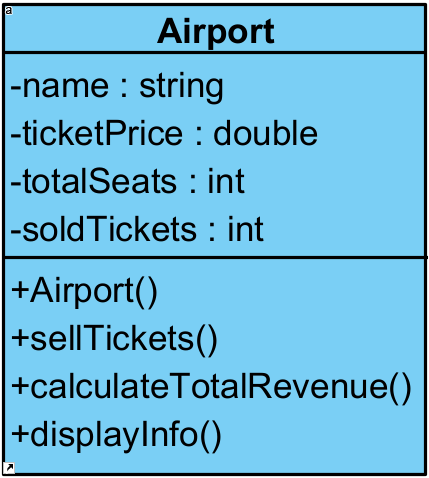


Рисунок 9 – Диаграмма класса Airport (Построена в программе Visual Paradigm)

## **Вывод**

В ходе выполнения лабораторной работы все цели и задачи были выполнены, были получены навыки работы с функциями, циклами, условным оператором, с классами C++, а также классами в языке Python и составлением диаграмм класса. В результате выполнения программы была получена программа для заполнения информации о рейсах в определённом аэропорту с использованием технологи объектно-ориентированного программирования.