Министерство науки и высшего образования РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа «Компьютерных технологий и информационных систем»

ОТЧЕТ

по дисциплине «Практикум по программированию»

**Лабораторная работа № 4**

**Выполнил:**

Cтудент гр. 5130902/30002 С. С. Фибих

**Проверил**

Ст. преподаватель А.М. Журавская

Санкт-Петербург

2024 г.

## **Цель**

Создание десктопного приложения на Windows Form. Овладение навыком разделение проекта на файлы и его структуризации. Перенос классового проекта на С++ в Windows Form. Портирование консольного приложения в Windows Form.

## **Задание**

**Упражнение 1**

Вам необходимо взять любое написанное вами ранее консольное приложение (лабораторная работа 2 или 3) с классами и сделать их него многофайловый проект. Также вы можете создать небольшое приложение с классами с нуля, если не хотите привязываться ко второй и третьей лабораторной.

Для того, чтобы сделать многофайловый проект вы можете использовать свой текущий проект, добавив в него элементы через обозреватель решений (в Visual Studio) “ Добавить элемент” выбрать .cpp и .h.

.h -- заголовочные файлы,

.cpp -- файлы исходного кода. Файлы подключаются через #include.

Какую файловую структуру необходимо реализовать (названия можно придумать свои):

- Main.cpp -- в данном файле находится меню и точка входа main;

- Class.h -- объявление класса, геттеров и сеттеров.

- Class.cpp -- описание функций класса.

- Function.h -- объявление прототипов функций.

- Function.cpp -- реализация функций.

## **Код программы**

**Часть 1.**

**Main.cpp**

#include <iostream>

#include "Class.h"

#include "Function.h"

using namespace std;

int main() {

system("chcp 1251");

system("cls");

Airport Y;

string test, wntd;

float Tcost, Tdiscont;

do {

cout << "Список комманд:\n"

<< "1 - Добавить тариф с фиксированной скидкой\n"

<< "2 - Добавить тариф без фиксированной скидкой\n"

<< "3 - Вывести список тарифов\n"

<< "4 - Вывести тариф с максимальной стоимостью\n"

<< "5 - Завершить работу\n"

<< "Введите комманду: ";

cin >> wntd;

cout << "\n";

if (wntd == "1") {

try {

cout << "Введите стоимость билета" << endl;

inputFloat(Tcost);

cout << "Введите фиксированную скидку" << endl;

inputFloat(Tdiscont);

cout << "Введите название направления" << endl;

cin >> test;

cin.get();

}

catch (int code) {

if (code == -1) cout << "Значение не входит в диапазон [0; 100000]" << endl;

else if (code == -2) cout << "Значение некорректно" << endl;

continue;

}

if (Tcost < Tdiscont) {

cout << "Скидка не может быть больше стоимости" << endl;

continue;

}

Y.addTarif(new DiscontTariff(Tcost, test, Tdiscont));

}

else if (wntd == "2") {

try {

cout << "Введите стоимость билета" << endl;

inputFloat(Tcost);

cout << "Введите название направления" << endl;

cin >> test;

cin.get();

}

catch (int code) {

if (code == -1) cout << "Значение не входит в диапазон [0; 100000]" << endl;

else if (code == -2) cout << "Значение некорректно" << endl;

continue;

}

Y.addTarif(new ClassicTariff(Tcost, test));

}

else if (wntd == "3") {

Y.ShowTariff();

}

else if (wntd == "4") {

Y.GetMaxTariff();

}

} while (wntd != "5");

cout << "\n\n\n\tСПАСИБО ЗА ВНИМАНИЕ\n\n\n";

return 0;

}

**Class.h**

#pragma once

#include <iostream>

#include <string>

#include <vector>

using namespace std;

class ITariff {

protected:

float cost;

string TName;

public:

ITariff(float cost, string TName);

virtual ~ITariff();

virtual float GetCost() const = 0;

virtual string GetName();

};

class DiscontTariff : public ITariff {

private:

float discont;

public:

DiscontTariff(float cost, string TName, float discont);

float GetCost() const override;

};

class ClassicTariff : public ITariff {

public:

ClassicTariff(float cost, string TName);

float GetCost() const override;

};

class Airport {

private:

vector<ITariff\*> tariffs;

public:

~Airport();

void addTarif(ITariff\* tarif);

void ShowTariff();

void GetMaxTariff();

};

**Class.cpp**

#include "Class.h"

ITariff::ITariff(float cost, string TName) : cost(cost), TName(TName) {

cout << "the class has been CREATED" << endl;

}

ITariff::~ITariff() {

cout << "the class has been DESTROYED" << endl;

}

string ITariff::GetName() {

return TName;

}

DiscontTariff::DiscontTariff(float cost, string TName, float discont)

: ITariff(cost, TName), discont(discont) {

}

float DiscontTariff::GetCost() const {

return (cost - discont);

}

ClassicTariff::ClassicTariff(float cost, string TName)

: ITariff(cost, TName) {

}

float ClassicTariff::GetCost() const {

return cost;

}

Airport::~Airport() {

for (ITariff\* tarif : tariffs) delete tarif;

cout << "Airport was deleted" << endl;

}

void Airport::addTarif(ITariff\* tarif) {

tariffs.push\_back(tarif);

}

void Airport::ShowTariff() {

for (const auto& tarif : tariffs) {

cout << tarif->GetCost() << " " << tarif->GetName() << endl;

}

}

void Airport::GetMaxTariff() {

if (tariffs.empty()) {

cout << "Нет тарифов" << endl;

return;

}

ITariff\* max\_cost = tariffs[0];

for (const auto& tarif : tariffs) {

if (tarif->GetCost() > max\_cost->GetCost()) max\_cost = tarif;

}

cout << "Стоимость билета - " << max\_cost->GetCost()

<< " ; Название - " << max\_cost->GetName() << endl;

}

**Function.h**

#pragma once

#include <string>

using namespace std;

float inputFloat(float& f);

bool isFloat(const std::string& input);

**Function.cpp**

#include "Function.h"

#include <iostream>

#include <sstream>

using namespace std;

float inputFloat(float& f) {

string temp;

while (true) {

cin >> temp;

if (isFloat(temp)) {

if (stof(temp) < 0 || stof(temp) > 100000) {

throw (-1);

continue;

}

break;

}

else {

cin.get();

throw (-2);

break;

}

}

return f = stof(temp);

}

bool isFloat(const std::string& input) {

std::istringstream stream(input);

float f;

if (!(stream >> f)) {

return false;

}

return stream.eof();

}

**Пример работы программы**

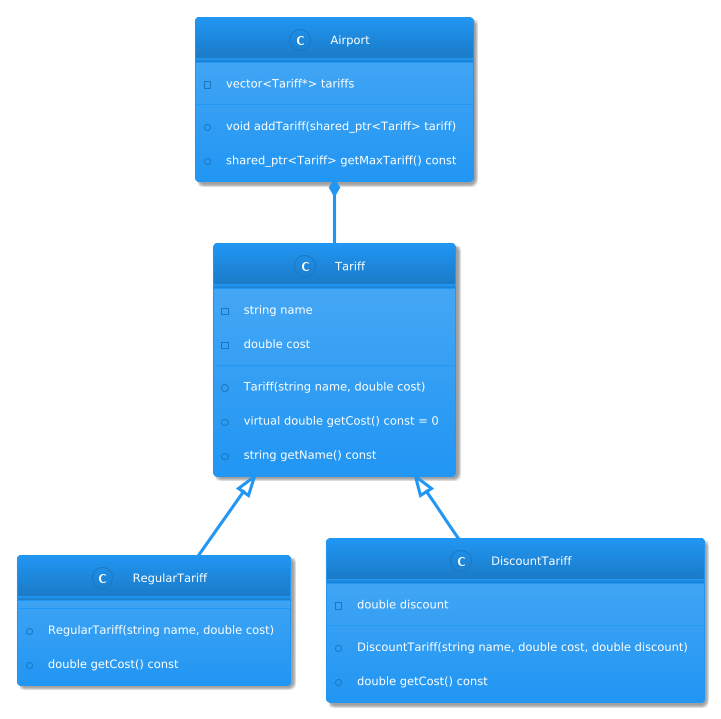
****

Рис. 1: Диаграмма классов

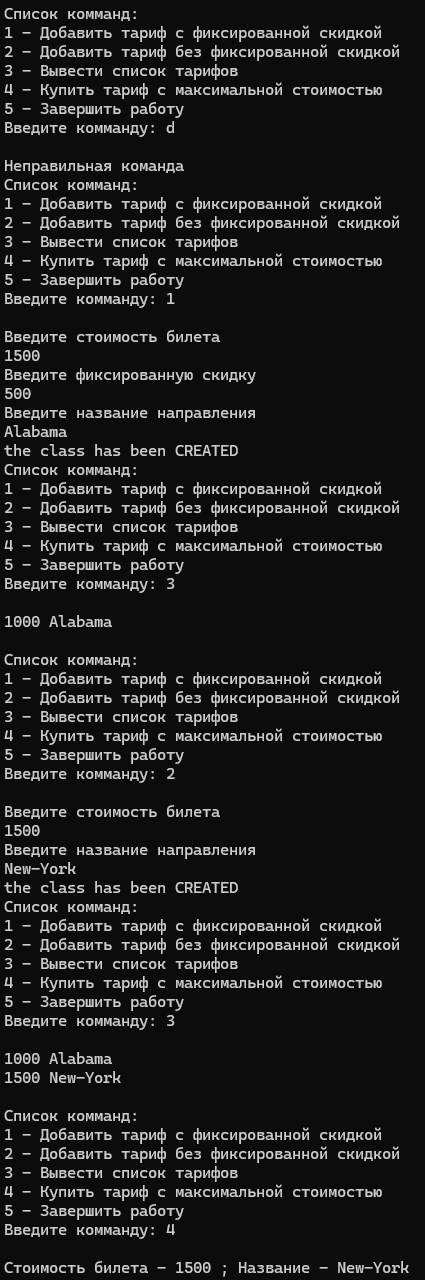


Рис. 2: Демонстрация работы

## **Вывод**

В ходе выполнения этой работы, я научился реализовывать шаблон проектирования классов Strategy, наследование, полиморфизм, а также парадигму интерфейсных классов.