### 

Выполнил

студент группы КТбо1-2 А. А. Кубанов

Принял

доцент ИКТИБ В. С. Лапшин

Таганрог 2024

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

(ФГАОУ ВО «ЮФУ»)

Институт компьютерных технологий и информационной безопасности

Кафедра системного анализа и телекоммуникаций

**ЛАБОРАТОРНАЯ РАБОТА № 4**

по дисциплине «Основы алгоритмизации и программирования»

Вариант 6

на тему:

«Разработка приложений на С++ в объектном стиле»

СОДЕРЖАНИЕ

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 4](#_Toc166196369)

[ЦЕЛЬ РАБОТЫ 4](#_Toc166196370)

[ЗАДАНИЕ 4](#_Toc166196371)

[ХОД РАБОТЫ 6](#_Toc166196372)

[АЛГОРИТМ 6](#_Toc166196373)

[ПРИМЕР РАБОТЫ ПРОГРАММЫ 7](#_Toc166196374)

[ВЫВОД 9](#_Toc166196375)

[ЛИСТИНГ 10](#_Toc166196376)

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ЗАДАНИЕ

Разработать классы для реализации индивидуального задания и отладить их в консольном приложении для Windows, используя библиотеку .NET.

**Вариант задания:**  
**Что дано:** Список операционных систем и наименований смартфонов, на которых эти ОС устанавливаются.

**Что получить:** Для заданной совокупности наименований смартфонов отобрать наименования ОС, в которых они могут устанавливаться.

# ХОД РАБОТЫ

## АЛГОРИТМ

1) Создаём проект в Visual studio и начинаем работу;

2) Создаём проект консольное приложение CLR (.NET Framework) на C++;

3) В меню VS нажимаешь проект и в свойствах проекта настраиваем точку входа main, а также ставим подсистему Windows;

4) Через обозреватель решений проекта добавляем два класса с .h и .cpp c названием Smartphone и OperatingSystem;

5) В Smartphone.h мы объявляем класс с инициализацией переменной name переданным значением и функцием для добавления операционной системы в operatingSystems. А также две функции для получения названия смартфона и вектора операционных систем;

6) В OperatingSystem.h мы объявляем класс с инициализацией переменной name для operationSystems и публичный метод getName, который возвращает переданное значение name;

7) В Smartphone.cpp мы описываем метод для добавления операционной системы в конец вектора. Эот метод позволяет объекту Smartphone хранить список операционных систем, которые он поддерживает или на которых он работает;

8) В OperationSystems.cpp мы просто объявляем заголовочный файл OperatingSystem.h, который содержит объявления класса;

9) В главном cpp файле мы создаём список смартфонов, у меня это iPhone, Samsung, Nothing\_Phone и Nokia. Также создаём список операционных систем: ios, Android и WindowsPhone. Добавляем список поддерживаемых операционных систем для смартфонов через addOperatingSystem. Затем даём возможность пользователю добавить данные в виде названия смартфона. Реализуем поиск поддерживаемых смартфонами операционных систем. Перебираются все названия смартфонов и проверяется совпадают ли они с тем, что ввёл пользователь, далее мы получаем список всех операционных систем и выводятся та операционная система, которая поддерживается на данном смартфоне. После того, как ОС найдено, программа завершается.

## ПРИМЕР РАБОТЫ ПРОГРАММЫ

Пример работы программы представлен ниже (Рис. 1).

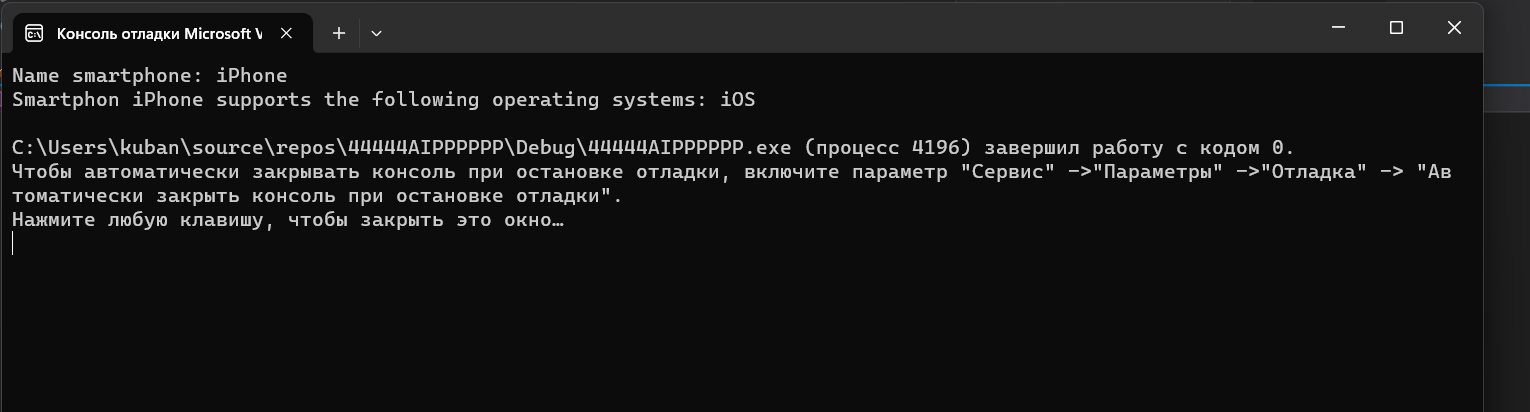


Рисунок 1

# ВЫВОД

По итогу лабораторной работы, я разработал классы для реализации индивидуального задания и отладить их в консольном приложении для Windows, используя библиотеку .NET.

# ЛИСТИНГ

44444AIPPPPPP.cpp:

#include "pch.h"

#include "Smartphone.h"

#include "OperatingSystem.h"

#include <iostream>

#include <vector>

using namespace std;

int main() {

// Создание списка операционных систем

vector<OperatingSystem> operatingSystems = {

OperatingSystem("iOS"),

OperatingSystem("Android"),

OperatingSystem("WindowsPhone")

};

// Создание списка смартфонов

vector<Smartphone> smartphones = {

Smartphone("iPhone"),

Smartphone("Samsung"),

Smartphone("Nothing\_Phone"),

Smartphone("Nokia")

};

// Добавление поддерживаемых операционных систем к смартфонам

smartphones[0].addOperatingSystem("iOS");

smartphones[1].addOperatingSystem("Android");

smartphones[1].addOperatingSystem("WindowsPhone");

smartphones[2].addOperatingSystem("Android");

smartphones[3].addOperatingSystem("WindowsPhone");

// Ввод данных пользователем

string phoneName;

cout << "Name smartphone: ";

cin >> phoneName;

// Поиск и вывод операционных систем для заданного смартфона

bool found = false;

for (const auto& phone : smartphones) {

if (phone.getName() == phoneName) {

found = true;

vector<string> osList = phone.getOperatingSystems();

cout << "Smartphon " << phoneName << " supports the following operating systems: ";

for (const auto& os : osList) {

cout << os << " ";

}

cout << endl;

break;

}

}

if (!found) {

cout << "A smartphone with the name " << phoneName << " not found." << endl;

}

return 0;

}

OperatingSystem.cpp:

#include "pch.h"

#include "OperatingSystem.h"

Smartphone.cpp:

#include "pch.h"

#include "Smartphone.h"

using namespace std;

void Smartphone::addOperatingSystem(const string& os) {

operatingSystems.push\_back(os);

}

OperatingSystem.h:

#include <string>

using namespace std;

class OperatingSystem

{

private:

string name;

public:

OperatingSystem(const string& name) : name(name) {}

string getName() const

{

return name;

}

};

Smartphone.h: