Учреждение образования Федерации профсоюзов Беларуси

«Международный университет «МИТСО»

Рег. № \_\_\_\_\_\_\_\_\_\_ Кафедра информационных систем и технологий

Дата \_\_\_\_\_\_\_\_\_2020

**КУРСОВАЯ РАБОТА**

на тему \_ Учет клиентов музыкальной студии

по дисциплине \_Название\_\_Основы конструирования программ\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Основные замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Отметка о допуске курсовой работы к  защите:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_  Подпись научного руководителя:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  (Ф.И.О. – полностью)  Курс \_\_1\_\_, группа \_\_\_2020\_\_\_\_  Факультет \_\_экономический\_\_\_  Специальность \_\_Информационные системы и технологии\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Научный руководитель:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность, ученая степень, ученое звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Ф.И.О. – полностью) |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc72055182)

[1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc72055183)

[2.ТРЕБОВАНИЕ К ПРОГРАММЕ 5](#_Toc72055184)

[3.КОНСТРУИРОВАНИЕ ПРОГРАММЫ 6](#_Toc72055185)

[4.ОПИСАНИЕ ИНТЕРФЕЙСА 7](#_Toc72055186)

[5.ВЫБОР СПОСОБА ОРГАНИЗАЦИИ ДАННЫХ 9](#_Toc72055187)

[6.РАЗРАБОТКА ПЕРЕЧНЯ ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ ПРОГРАММЫ 10](#_Toc72055188)

[7.ПРОГРАММНАЯ ЧАСТЬ 16](#_Toc72055191)

[ЗАКЛЮЧЕНИЕ 17](#_Toc72055192)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 17](#_Toc72055193)

[ПРИЛОЖЕНИЕ 19](#_Toc72055194)

# **ВВЕДЕНИЕ**

Была поставлена задача разработать программный модуль «Информационные процессы поликлиники», предназначенный для использования работников регистратуры. В базе содержатся сведения о пациентах (фамилия, имя, отчество, история болезней). При возникновения нового пациента или возникновение болезни, данные заноситься в базу.

Программа создана для систематизации больных, быстрого просмотра/изменения и добавления новых студентов.

Цель разработки обеспечить работы деканата быстро и максимально эффективно, удовлетворять запросы работников регистратуры, а также должен быть возможность пополнять базу, для оптимизации процессов поликлиники. И неотъемлемой частью данной программы должна высокая оптимизация, чтобы пользователь мог запустить даже на самом слабом устройстве.

Задачи данного программного продукта: быстро и гибко редактироваться со стороны пользователя. С точки зрения использования пользователя программа должна иметь красивый, понятный и интуитивный интерфейс чтобы было владельцу приятно пользоваться программой. Так же требуется предусмотреть возможность добавления новых функций в программе.

Для разработки программы был использован текстовый файл и язык программирования C++ разработанный в среде Visual Studio компаний Microsoft.

Visual Studio-линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой Windows Forms, а также веб-сайты, веб-приложения, веб-службы, как в родном, так и в управляемом кодах для всех платформ, поддерживающих Windows, Windows Mobile, Windows Ce, .Net Framework, Xbox, Windows Phone .Net Compact Framework и Silverling.

# **1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

Была поставлена задача разработать программный модуль «Информационные процессы поликлиники», предназначенный для использования работников регистратуры. В базе содержатся сведения о пациентах (фамилия, имя, отчество, история болезней). При возникновения нового пациента или возникновение болезни, данные заноситься в базу.

Поликлиника имеет здание (помещение), в которых работают врачи различной специализации. Для каждого больного выписываются карточки, куда заноситься история болезни. Первоначально больной приходит к терапевту. Если есть необходимость терапевт направляет больного к врачам специалистам.

Из-за роста населения и городов возникла необходимость роста количества и размера поликлиник. Из-за этого и возникла потребность в данном программном продукте. Поэтому разрабатываемая программа должна выдержать весьма серьёзную конкуренцию.

Функции программного продукта: содержатся сведения о пациентах (фамилия, имя, отчество, история болезни) должны иметь понятные данные и возможность обрабатывать их (добавлять, просматривать, изменять).

Данный программный продукт выгодно отличается от других подобных тем что, пользуясь им, пользователь может не только хранить данные пациентов, но производить различные манипуляции с ними.

# **2.ТРЕБОВАНИЕ К ПРОГРАММЕ**

Исходя из задач, описанных выше главными требованиями к языку, на котором будет написана программы — это тонкая настройка и возможность использованием минимум вычислительных ресурсов добиваться максимум результата.

Для выполнения данных задач очень подходит такой язык программирования как с++. Так как его из-за того, что он является весьма низкоуровневым языком программирования, а значит он больше других приближен к машинному коду, из-за этого с++ дает возможность оптимизировать проект так что даже на очень старых устройствах он будет работать.

Windows 10 – операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT. После Windows 8.1 система получила номер 10, минуя 9. Серверный аналог Windows 10 – Windows Server 2016. Система призвана стать единой для разных устройств, таких как персональные компьютеры, планшеты, смартфоны, консоли Xbox One и пр. Доступна единая платформа разработки и единый магазин универсальных приложений, совместимых со всеми поддерживаемыми устройствами. Windows 10 поставляется в качестве услуги с выпуском обновлений на протяжении всего цикла поддержки. В течение первого года после выхода системы пользователи могли бесплатно обновиться до Windows 10 на устройствах под управлением лицензионных копий Windows 7, Windows 8.1 и Windows Phone 8.1. Среди значимых нововведений – голосовая помощница Кортана, возможность создания и переключения нескольких рабочих столов и др. Windows 10 – последняя «коробочная» версия Windows, все последующие версии будут распространяться исключительно в цифровом виде.

Среда разработки Microsoft Visual Studio. Вид приложения – консольное. Программа построена на базе объектно-ориентированного программирования. Способ организации данных: классы и очередь. Каждая логическая завершенная данных подзадача программы реализована в виде методов. К защите курсовой предоставляется: консольное приложение и пояснительная записка.

# **3.КОНСТРУИРОВАНИЕ ПРОГРАММЫ**



В текущей главе представлено описание алгоритма, который имеет начало алгоритма и конец, внутри алгоритма имеются циклы и иные операций.

Первым пунктом идёт выбор операции, за счет введение цифры от 1 до 6 и последующей за вводом проверки.

Следующий шаг — это работа непосредственно с программой в зависимости от выбранных данных. Добавление данных, вывод данных,изменение данных.

# **4.ОПИСАНИЕ ИНТЕРФЕЙСА**

При старте программы происходит выбор желаемой операции (рис.4.1)

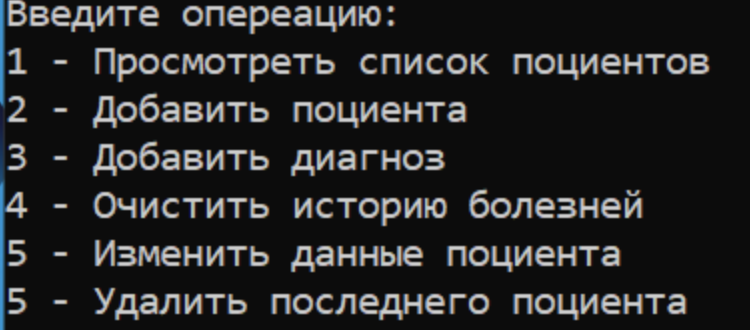


Рис. 4.1

При выборе первого пункта происходит вывод всех пациентов и историю болезни на консоль (рис 4.2).

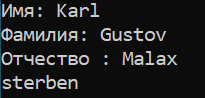


Рис 4.2

При выборе второго пункта в консоль необходимо ввести данные пациента, последовательно (рис. 4.3).

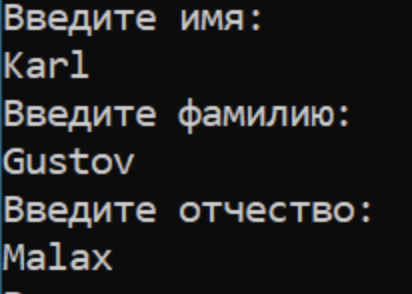


Рис.4.3

При выборе 3-го пункта необходимо вначале необходимо ввести номер пациента. А затем ввести название болезни (4.4).

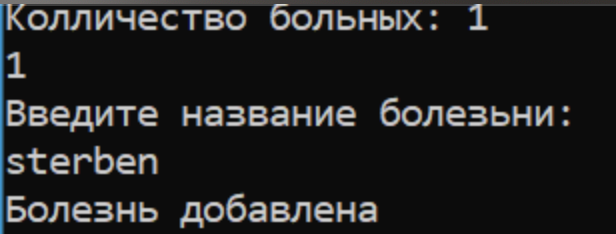


Рис. 4.4

При выборе четвертого пункта происходит очистка истории болезни конкретного пациента. Путём ввода номера пациента (рис 4.5).

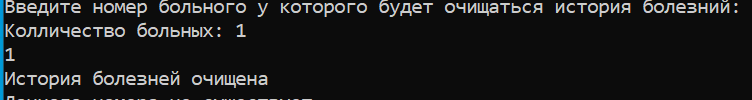


Рис.4.5

При выборе пятого пункта происходит изменение данных конкретного студента. Путём ввода номера пациента и последующее его изменение (рис 4.6).

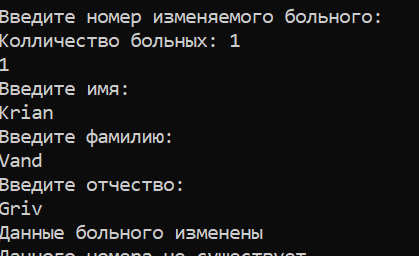


Рис 4.6

При выборе шестого пункта происходит завершение программы.

# **5.ВЫБОР СПОСОБА ОРГАНИЗАЦИИ ДАННЫХ**

Класс **Person** (рис 5.1) – класс, который представляет собой пациента поликлиники.

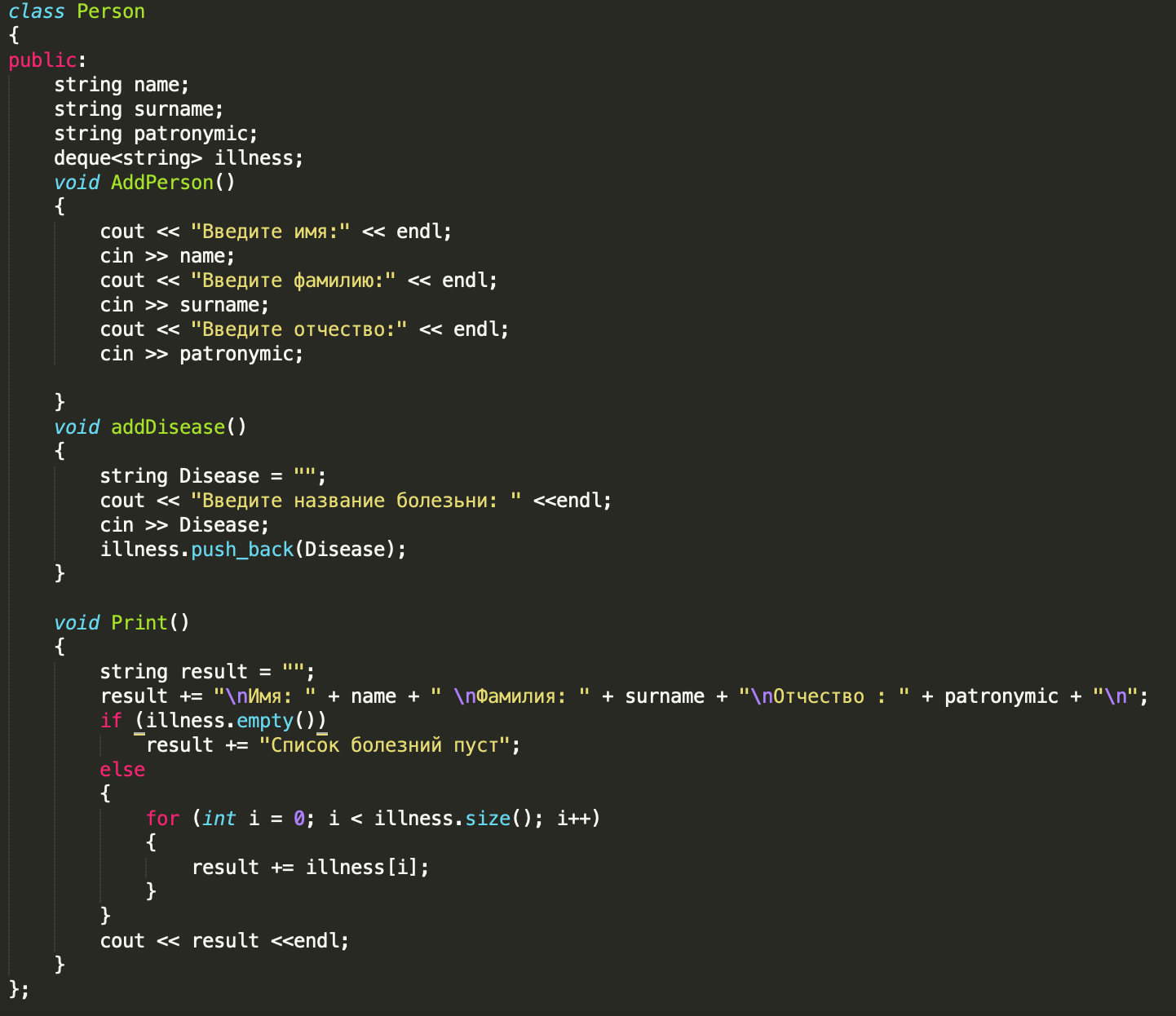
****

рис 5.1

Переменные:

**name** – имя пациента.

**surname** – фамилия пациента.

**patronymic –** отчество пациента.

**illness** – история болезней.

Методы:

void **AddPerson**() – добавление пациента данных.

void **addDisease**() – добавление болезни к истории болезни.

void **Print**() – вывод данных класса на консоль.

# **6.РАЗРАБОТКА ПЕРЕЧНЯ ПОЛЬЗОВАТЕЛЬСКИХ МЕТОДОВ ПРОГРАММЫ**

void **AddPerson**(рис 6.1) – добавление пациента данных. Путём введения данных поочерёдно данных экземпляра класса на консоль.

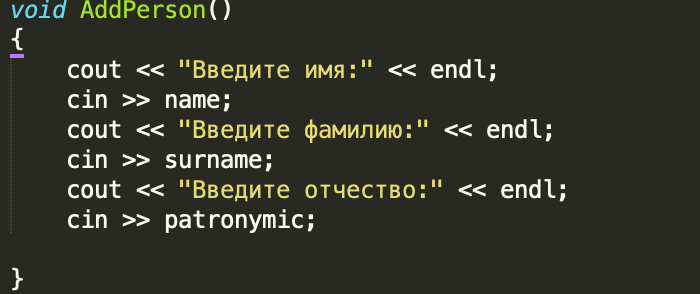


Рис 6.1

void **addDisease**(рис 6.2) – добавление болезни к истории болезни. Путём ввода данных в консоль.

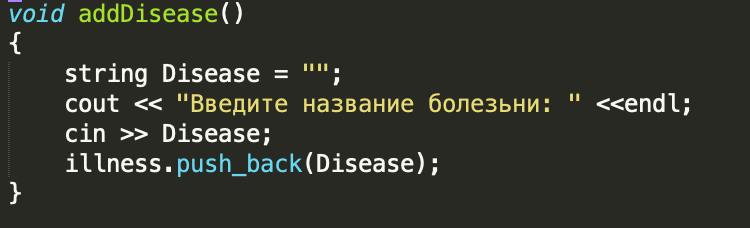


рис 6.2

void **Print**(Рис 6.3) – вывод данных экземпляра класса на консоль. Вначале создаётся происходит инициализация переменной result, в которую в дальнейшем будет присваиваться строка с информацией о классе.

Вначале происходит присвоение данных пациента. А потом добавление всей истории болезни.

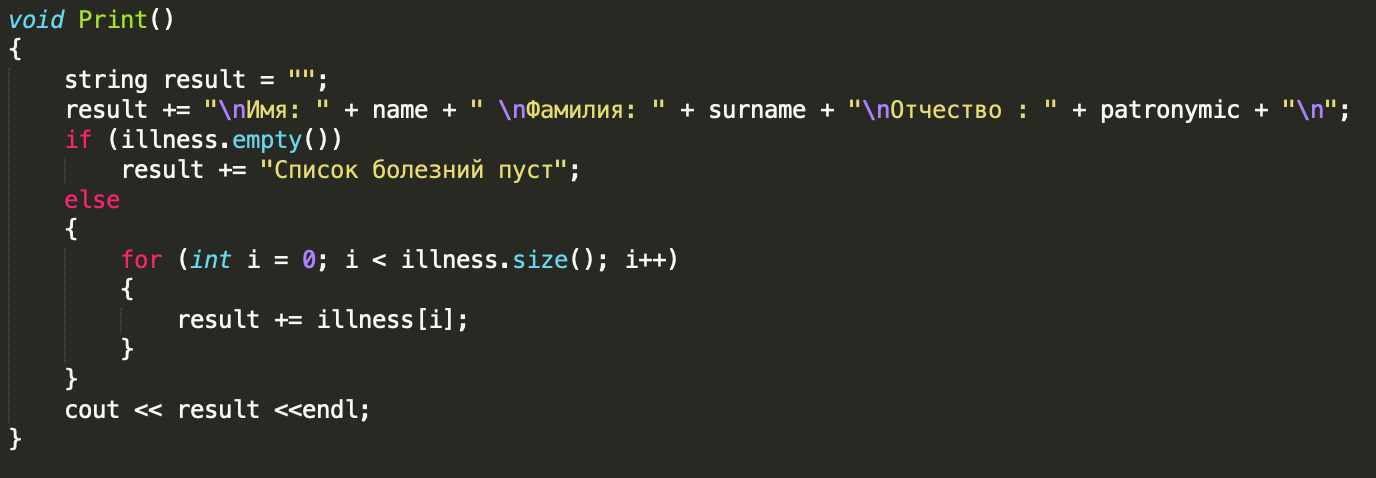


рис 6.3

# **7.ПРОГРАММНАЯ ЧАСТЬ**

Вначале происходит выбор операции:

1. Просмотреть список пациентов
2. Добавление пациента
3. Добавить диагноз
4. Очистка истории болезней
5. Изменить данные пациента
6. Удалить последнего поциента

Далее при помощи конструкции switch происходит переход к выбранной операции.

1: Происходит перебор коллекции Persons и при каждой итерации цикла происходит вызов метода класса Person Print(). Который выводит данные на объекта класса на консоль.

2: Происходит вызов метода AddPerson. Его суть заключается в добавление пациента данных. Путём введения данных поочерёдно данных экземпляра класса на консоль.

3: Происходит вызов метода addDisease. Его суть заключается в добавление болезни к истории болезни. Путём ввода данных в консоль.

4: Происходит ввод переменной которая отвечает за конкретный элемент коллекции. При верном вводе данных происходит очистка коллекции illness для конкретного элемента коллекции

5: Происходит ввод переменной которая отвечает за конкретный элемент коллекции. При верном вводе данных происходит вызов метода addPerson суть которого заключается в поочерёдном вводе данных пациента с консоли.

6: К коллекции illness происходит вызов метода pop\_back().

# **ЗАКЛЮЧЕНИЕ**

Исходя из поставленных целей и задач было сконструировано, разработана и протестирован программный модуль «Информационные процессы поликлиники», предназначенный для использования работников регистратуры. В базе содержатся сведения о пациентах (фамилия, имя, отчество, история болезней). При возникновения нового пациента или возникновение болезни, данные заноситься в базу.

1)Быстро и гибко редактироваться со стороны работников регистратуры.

2)Программный продукт имеет красивый, понятный и интуитивный интерфейс чтобы пользователю было приятно пользоваться программой.

3)Так же предусмотрено возможность добавления новых функций в программе.

Цель была выполнены, картотека имеет следующие преимущества:

1)Работа поликлиники происходит быстро и максимально эффективно.

2)Удовлетворяет запросы поликлиники.

2)Также реализована возможность пополнять базу, для оптимизации процессов поликлиники.

3)Реализована высокая оптимизация, что-то привело к возможности пользователем запускать приложение даже на самом слабом устройстве.

Данный программный продукт выгодно отличается от других подобных тем что, пользуясь им, пользователь может не только хранить данные пациентов, но и взаимодействовать с их личными делами.

Таким образом программа работает без сбоев и все задачи были выполнены.

# **СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

1. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирование / Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. – СПб: Питер, 2015. – 368с.
2. Коплиен Дж. Мультипарадигменное проектирование для С++ / Коплиен Дж. – Питер, 2005.
3. Обзор обновлений и новых функций Windows 10 [Электронный ресурс]. – Microsoft, 2020. – Режим доступа: https://www.microsoft.com/ru-ru/windows/features. – Дата доступа: 22.05.2020.
4. Объектно-ориентированный анализ и проектирование с примерами приложений / Гради Буч [и др.]. – 3-е изд. – М.: ООО «И.Д. Вильямс», 2008. – 720 с.
5. Стивен Прата Язык программирования C++ / Стивен Прата – Вильямс, 2012.
6. Страуструп Б. Программирование. Принципы и практика с использованием C++ / Страуструп Б. – 2-е изд. – Вильямс, 2016.
7. Model-View-Presenter [Электронный ресурс]. – Википедия, 2020. – Режим доступа: https://ru.wikipedia.org/wiki/Model-View-Presenter. – Дата доступа: 22.04.2020.
8. SFML [Электронный ресурс]. – SFML, 2020. – Режим доступа: https://www.sfml-dev.org. – Дата доступа: 22.04.2020.
9. SFML [Электронный ресурс]. – Википедия, 2020. – Режим доступа: https://ru.wikipedia.org/wiki/SFML. – Дата доступа: 22.04.2020.

# **ПРИЛОЖЕНИЕ**

#include<iostream>

#include<string>

#include<conio.h>

#include<fstream>

#include<iomanip>//Для пробелов между словами и символами

#include<Windows.h>

#define N 1

using namespace std;

struct OT

{

int shifr\_studenta;

char imja[20 + 1];

char familia[20 + 1];

char otchestvo[20 + 1];

char facultet[10 + 1];

int shifr\_gruppi;

int matematika;

int angl;

int history;

int count;

};

bool add(int& count, OT\* ot);

void drawLineW();

void drawLineN();

void print(int count, OT\* ot);

void Menu(int count, OT\* ot);

void horosho(int count, OT\* ot);

void ploho(int count, OT\* ot);

void search\_student(int count, OT\* ot);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

system("cls");

int count = N;

OT\* ot = new OT[count];

Menu(count, ot);

\_getch();

return 0;

}

void Menu(int count, OT\* ot)

{

system("cls");

int numberSearch;

do

{

cout << "Здравствуйте, пожалуйста выбирайте поле в Меню" << endl;

cout << "1- Добавление студента " << endl;

cout << "2- Печать студентов" << endl;

cout << "3- Печать студентов со средним баллом 4 и выше" << endl;

cout << "4- Печать студентов со средним баллом 3 ниже " << endl;

cout << "5- вывести информацию о студенте" << endl;

cout << "6- Выход из программы" << endl;

cin >> numberSearch;

switch (numberSearch)

{

case 1: add(count, ot); break;

case 2: print(count, ot); break;

case 3: horosho(count, ot); break;

case 4: ploho(count, ot); break;

case 5: search\_student(count, ot); break;

case 6: exit(0); break;

}

} while (numberSearch != 6);

system("cls");

}

bool add(int& count, OT\* ot)

{

system("cls");

cout << setw(50) << "Добавленние студента" << endl;

for (int i = 0; i < count; i++)

{

system("cls");

cout << "Номер группы: " << endl;

cin.ignore(4, '\n') >> setw(6) >> ot[i].shifr\_gruppi;//удалить разделитель строк,// пропускает указанное число элементов от текущей позиции чтения

cout << "Фамилия студента " << endl;

cin.ignore(20, '\n') >> setw(20) >> ot[i].familia;

cout << "Отчество студента " << endl;

cin.ignore(20, '\n') >> setw(20) >> ot[i].otchestvo;

cout << "Имя студента " << endl;

cin.ignore(20, '\n') >> setw(20) >> ot[i].imja;

cout << "Номер студента: " << endl;

cin.ignore(4, '\n') >> setw(6) >> ot[i].shifr\_studenta;

cout << "Оценка по математике " << endl;

cin.ignore(4, '\n') >> setw(1) >> ot[i].matematika;

cout << "Оценка по английскому " << endl;

cin.ignore(4, '\n') >> setw(1) >> ot[i].angl;

cout << "Оценка по история " << endl;

cin.ignore(4, '\n') >> setw(1) >> ot[i].history;

}

system("cls");

for (int i = 1; i < count; i++)

for (int j = 0; j < count; j++)

if (ot[i].shifr\_gruppi == ot[j].shifr\_gruppi &&

ot[i].familia == ot[j].familia &&

ot[i].otchestvo == ot[i].otchestvo &&

ot[i].imja == ot[i].imja &&

ot[i].shifr\_studenta == ot[i].shifr\_studenta &&

ot[i].matematika == ot[i].matematika &&

ot[i].angl == ot[i].angl &&

ot[i].history == ot[i].history

)

{

ot[j].count += ot[i].count;

return true;

}

return false;

char choice;

do {

cout << "\n Возвращение в меню кнопка -y; выход из Меню любая другая кнопка... " << endl;

cin >> choice;

if (choice == 'y')

Menu(count, ot);

else return exit;

} while (1);

system("cls");

}

void drawLineW()

{

for (int i = 0; i < 80; ++i)

cout << "\_";

}

void drawLineN()

{

for (int i = 0; i < 80; ++i)

cout << "-";

}

void print(int count, OT\* ot)

{

system("cls");

setlocale(LC\_ALL, "Russian");

cout << setw(40) << "Учёт успеваемости студента" << endl;

drawLineW();

cout << setiosflags(ios::left) << setw(6) << "\n| Шифр студента: " << setw(20) << "| Фамилия студента: |"

<< setw(20) << "| Имя студента: |"

<< setw(20) << "| Отечество студента: |" << setw(6) << " шифр группы: |" << setw(1) << " оценка по английскому: |"

<< setw(1) << " оценка по математике: |" << setw(1) << " оценка по истории: |"

<< setw(1) << "Среднее: |" << endl;

drawLineN();

for (int i = 0; i < count; ++i)

{

float srednee = (ot[i].matematika + ot[i].angl + ot[i].history) / 3;

cout << setiosflags(std::ios::left) << "\n" << "| "

<< setw(6) << ot[i].shifr\_studenta << " | "

<< setw(20) << ot[i].familia << " | "

<< setw(20) << ot[i].imja << " | "

<< setw(20) << ot[i].otchestvo << " | "

<< setw(6) << ot[i].shifr\_gruppi << " | "

<< setw(1) << ot[i].angl << " | "

<< setw(1) << ot[i].matematika << " | "

<< setw(1) << ot[i].history << " | "

<< setw(1) << srednee << " | "

<< endl;

drawLineN();

}

char choice;

do {

cout << "\n Возвращение в меню кнопка -y; выход из Меню любая другая кнопка... " << endl;

cin >> choice;

if (choice == 'y')

Menu(count, ot);

else return exit(1);

} while (1);

}

void horosho(int count, OT\* ot)

{

system("cls");// Получение из файла кол-ва записей в б.д.

float srednee;

cout << setw(40) << "Список студентов с баллом 4 и выше" << endl;

drawLineW();

cout << setiosflags(ios::left) << setw(6) << "\n| Шифр студента: " << setw(6) << "| Шифр группы: |"

<< setw(20) << "Фамилия |" << setw(20) << " Имя |" << setw(20) << "Отчество |" << setw(1) << " Оценка по английскому: |" << setw(1) << " Оценка по математике: |"

<< setw(1) << " Оценка по истории : |"

<< setw(1) << "Среднее: |"

<< endl;

drawLineN();

for (int i = 0; i < count; ++i)

{

srednee = (ot[i].matematika + ot[i].angl + ot[i].history) / 3;

if (srednee >= 4) {

srednee = (ot[i].matematika + ot[i].angl + ot[i].history) / 3;

cout << setw(6) << ot[i].shifr\_studenta << " | "

<< setw(6) << ot[i].shifr\_gruppi << " | "

<< setw(20) << ot[i].familia << " | "

<< setw(20) << ot[i].imja << " | "

<< setw(20) << ot[i].otchestvo << " | "

<< setw(1) << ot[i].angl << " | "

<< setw(1) << ot[i].matematika << " | "

<< setw(1) << ot[i].history << " | "

<< setw(1) << srednee << " | "

<< endl;

drawLineN();

}

}

char choice;

do {

cout << "\n Возвращение в меню кнопка -y; выход из Меню любая другая кнопка... " << endl;

cin >> choice;

if (choice == 'y')

Menu(count, ot);

else return exit(1);

} while (1);

}

void ploho(int count, OT\* ot) {

system("cls");// Получение из файла кол-ва записей в б.д

float srednee;

cout << setw(40) << "Список студентов с баллом ниже 4" << endl;

drawLineW();

cout << setiosflags(ios::left) << setw(6) << "\n| Шифр студента: " << setw(6) << "| Шифр группы: |"

<< setw(20) << "Фамилия |" << setw(20) << " Имя |" << setw(20) << "Отчество |" << setw(1) << " Оценка по английскому: |" << setw(1) << " Оценка по математике: |"

<< setw(1) << " Оценка по истории : |"

<< setw(1) << "Среднее: |"

<< endl;

drawLineN();

for (int i = 0; i < count; ++i)

{

srednee = (ot[i].matematika + ot[i].angl + ot[i].history) / 3;

if (srednee < 3) {

srednee = (ot[i].matematika + ot[i].angl + ot[i].history) / 3;

cout << setw(6) << ot[i].shifr\_studenta << " | "

<< setw(6) << ot[i].shifr\_gruppi << " | "

<< setw(20) << ot[i].familia << " | "

<< setw(20) << ot[i].imja << " | "

<< setw(20) << ot[i].otchestvo << " | "

<< setw(1) << ot[i].angl << " | "

<< setw(1) << ot[i].matematika << " | "

<< setw(1) << ot[i].history << " | "

<< setw(1) << srednee << " | "

<< endl;

drawLineN();

}

}

char choice;

do {

cout << "\n Возвращение в меню кнопка -y; выход из Меню любая другая кнопка... " << endl;

cin >> choice;

if (choice == 'y')

Menu(count, ot);

else return exit(1);

} while (1);

}

void search\_student(int count, OT\* ot) {

system("cls");

for (int i = 0; i < count; ++i) {

float srednee = (ot[i].matematika + ot[i].angl + ot[i].history) / 3;

cout << setw(60) << "Введите код студента" << endl;

cin >> ot[i].shifr\_studenta;

cout << setw(60) << "Студент" << endl;

drawLineW();

cout << setiosflags(ios::left) << setw(6) << "\n| Шифр студента: " << setw(6) << "| Шифр группы: |"

<< setw(20) << "Фамилия |" << setw(20) << " Имя |" << setw(20) << "Отчество |" << setw(1) << " Оценка по английскому: |" << setw(1) << " Оценка по математике: |"

<< setw(1) << " Оценка по истории : |"

<< setw(1) << "Среднее: |"

<< endl;

drawLineN();

if (ot[i].shifr\_studenta) {

cout << "\n" << "| "

<< setw(6) << ot[i].shifr\_studenta << " | "

<< setw(6) << ot[i].shifr\_gruppi << " | "

<< setw(20) << ot[i].familia << " | "

<< setw(20) << ot[i].imja << " | "

<< setw(20) << ot[i].otchestvo << " | "

<< setw(1) << ot[i].angl << " | "

<< setw(1) << ot[i].matematika << " | "

<< setw(1) << ot[i].history << " | "

<< setw(1) << srednee << " | "

<< endl;

drawLineN();

}

}

char choice;

do {

cout << "\n Возвращение в меню кнопка -y; выход из Меню любая другая кнопка... " << endl;

cin >> choice;

if (choice == 'y')

Menu(count, ot);

else return exit(1);

} while (1);

}