Учреждение образования Федерации профсоюзов Беларуси

«Международный университет «МИТСО»

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

Дата \_\_\_\_\_\_\_\_\_2020

**КУРСОВАЯ РАБОТА**

на тему \_\_\_\_Название\_\_\_Разработать программный модуль «Картотека недвижимости»\_\_\_\_\_\_\_\_

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

|  |  |
| --- | --- |
| Основные замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Отметка о допуске курсовой работы к  защите:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_  Подпись научного руководителя:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  Урбанович Владислав Сергеевич (Ф.И.О. – полностью)  Курс \_\_1\_\_, группа \_\_\_1920\_\_\_\_  Факультет \_\_экономический\_\_\_  Специальность \_\_Информационные системы и технологии\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Научный руководитель:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность, ученая степень, ученое звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Ф.И.О. – полностью) |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc70868457)

[1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc70868458)

[**1.1. Общие положения 4**](#_Toc70868459)

[**1.2. Постановка задачи 4**](#_Toc70868460)

[**1.3. Спецификация входных и выходных данных 4**](#_Toc70868461)

[**1.4. Выбор средства 5**](#_Toc70868462)

[**1.5.Описывается требуемое оборудование и программное обеспечение 6**](#_Toc70868463)

[2.ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ 7](#_Toc70868464)

[**2.1. Алгоритм решения 7**](#_Toc70868465)

[**2.2. Описание программы 8**](#_Toc70868466)

[**2.3. Результаты работы программы 11**](#_Toc70868467)

[3.РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 12](#_Toc70868468)

[ЗАКЛЮЧЕНИЕ 13](#_Toc70868469)

[СПИСОК ЛИТЕРАТУРЫ 14](#_Toc70868470)

[ПРИЛОЖЕНИЕ 15](#_Toc70868471)

# ВВЕДЕНИЕ

Была поставлена задача разработать программный продукт «Учет наличия комнат, их площадей̆ и прописанных в них жильцов», предназначенный для использования работниками общежития и деканата.

Программа создана для автоматизации процесса заселения студентов в общежития. Учёта студентов в общежитии. Определение студентов, нарушающих устав общежития.

Цель разработки обеспечить работу программного продукта быстро и максимально эффективно, удовлетворять работников, а также должен быть возможность пополнять базу данных, для учёта студентов. Основная же задача стоит в высокой оптимизация, чтобы пользователь мог запустить даже на самом слабом устройстве.

Задачи данного программного продукта: быстро и гибко редактироваться со стороны пользователя программного продукта. С точки зрения использования пользователя программа должна иметь красивый, понятный и интуитивный интерфейс чтобы было клиенту 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

## 1.1. Общие положения

Расселение студентов в общежитие крайне трудоёмкий и сложный процесс. Он требует участие нескольких квалифицированных специалистов. Так же процесс заселения требует учёта студентов, что является трудной задачей, при которой возникают большое количество ошибок. Именно для решения данных задачи был реализован данный программный продукт.

Была поставлена задача разработать программный продукт «Учет наличия комнат, их площадей̆ и прописанных в них жильцов», предназначенный для использования работниками общежития и деканата.

Программа создана для автоматизации процесса заселения студентов в общежития. Учёта студентов в общежитии. Определение студентов, нарушающих устав общежития.

Цель разработки обеспечить работу программного продукта быстро и максимально эффективно, удовлетворять работников, а также должен быть возможность пополнять базу данных, для учёта студентов. Основная же задача стоит в высокой оптимизация, чтобы пользователь мог запустить даже на самом слабом устройстве.

Данный программный продукт выгодно отличается от других подобных тем что, пользуясь им, пользователь может не хранить данные студентов, проводить различные операции (удаление/изменение) .

## 1.2. Постановка задачи

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.

## 1.3. Спецификация входных и выходных данных

В данном программном продукте используются следующие типы данных типа данных: sting, double, int bool. А также присутствуют пользовательские типы данных: Person, Room.

boll – используется только внутри программы, для проверки, подсчёта, иннерваций и так далее. Пользователь на данный этого типа влиять и пользоваться не может.

String – используется в двух случаях. Либо для наименования данных (присвоение данным название удобного пользователя). Во втором же случае данный тип данных используется для вывода всех данных на экран.

Int – используется для указание возраста студентов, номер комнаты, пощади комнаты, количество жильцов

В данном проекте реализовано два класса Person, Room.

Person – хранит данные студентов (имя, фамилия, возраст).

Room – вокруг данного класса строиться весь функционал программы. Данный класс содержит данные комнаты и студентов заселённые в комнату.

Так же в данном классе реализован конструктор присваивания и есть метод.

## 

## 1.4. Выбор средства

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.

Среда разработки Microsoft Visual Studio. Вид приложения – консольное. Программа построена на базе объектно-ориентированного программирования. Способ организации данных поля классов. Способ хранения данных – динамические массивы. Каждая логическая завершенная данных подзадача программы реализована в виде методов. Построение программного кода соответствует с <<C++ Code Convention>>. К защите курсовой предоставляется: консольное приложение и пояснительная записка

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.

## 1.5.Описывается требуемое оборудование и программное обеспечение

Для полноценного функционирования разработанного программного приложения, качественного выполнения всех поставленных задач, требуется ПК следующей конфигурации:

* процессор Intel Core i3;
* оперативная память DDR3 512 МБ или больше;
* минимально свободного места на жёстком диске 512 МБ;
* мышь;
* операционная система Windows 7/8/10.

# 2.ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ

## 2.1. Алгоритм решения

Алгоритм данного программного продукта строиться на условном операторе switch. Данный оператор имеет ряд плюсов и минусов. При его использовании значительно ухудшается читаемость кода, но зато он крайне удобен с точки зрения построения алгоритма.

Вначале идёт инициализация коллекции Rooms. Она хранит все комнаты в общежитии. Данная коллекция является основной в данном программном продукте.

Далеепроисходит переопределение кодировки консоли, путём использования следующих функций:

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "Russian");

Весь алгоритм описываемый далее помещён в бесконечный цикл, чтобы иметь возможность пользоваться приложением неоднократно, без его перезапуска.

Далее пользователем программного продукта производиться выбор желаемой операции путём ввода числового значения (рис 2.1.1)

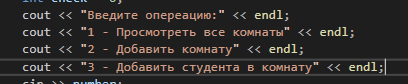


рис 2.1.1

Происходит перебор коллекции rooms и при каждой инерции происходит вызов метода Print. Данный алгоритм вынесен в отдельную функции PrintDeque(рис 2.1.2).

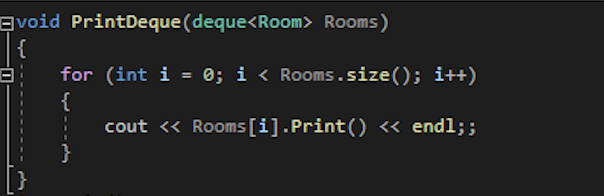


рис 2.1.2

На данном участке кода реализован добавления комнаты в базу при(рис 2.1.3) помощи методa addRoom класса Room(рис 2.1.4). И потом данные комната записывается в список комнат.



рис 2.1.3

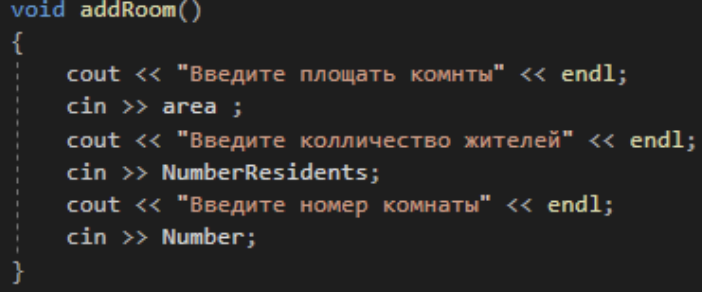


рис 2.1.4

На данном участке кода реализован добавления студента в комнату (рис 2.1.5).Вначале необходимо ввести номер комнаты в которую будет заселяться студент. Потом происходит проверка существует ли данная комната в списке. При нахождении совпадений вызывается метод addPerson() – который добавляет данные студента(рис 2.1.6). В обратном же случае на консоль вводиться сообщение: “Комната не существует”.

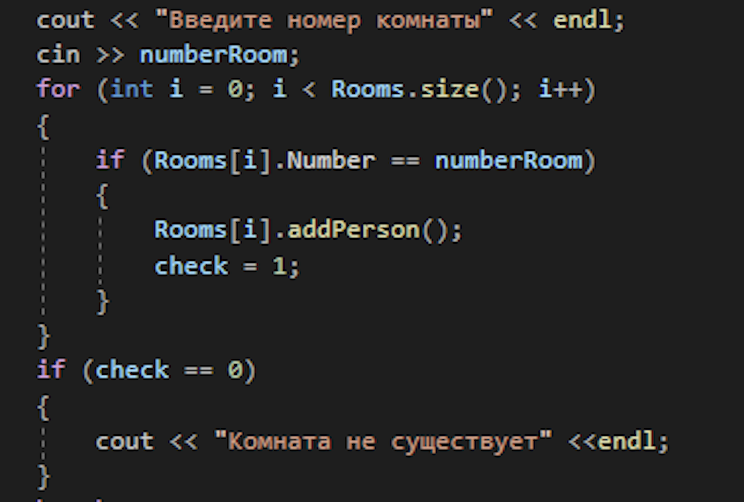


Рис 2.1.5

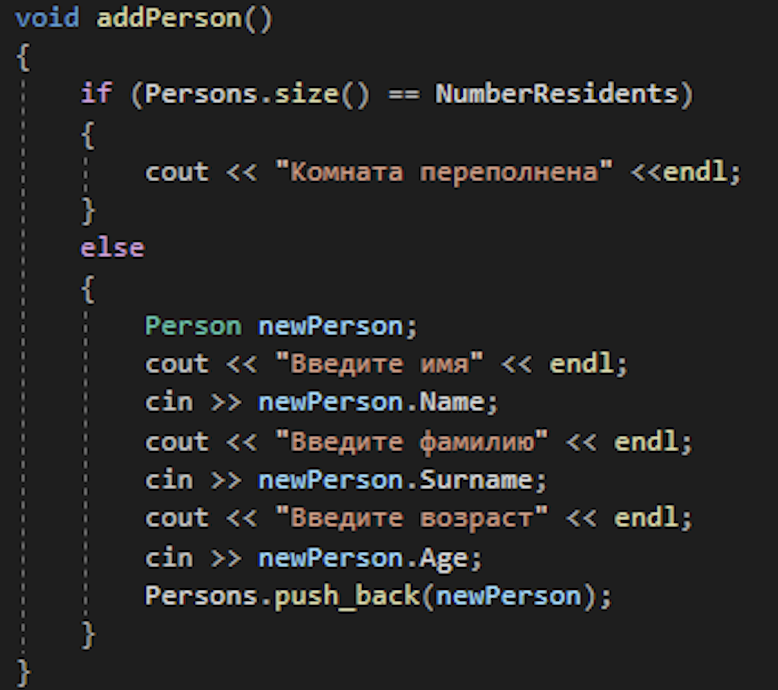


Рис 2.1.6

## 2.2. Описание программы

В данной программы используются следующие библиотеки:

#include <iostream> - основная библиотека, которая реализует основные возможность языка С++.

#include <string> - библиотека для возможности работы с типом данных string.

#include <random> - данная библиотека необходима для реализации работы с генератором случайных числах.

#include <ctime> - данная библиотека необходима для реализации работы с датой и времени.

#include <deque> - данная библиотека необходима для реализации работы с двухсторонней очередью.

В данном проекте реализовано два класса Person, Room.

Person – хранит данные студентов (имя, фамилия, возраст)(рис 2.2.1).

## 

## Рис 2.2.1

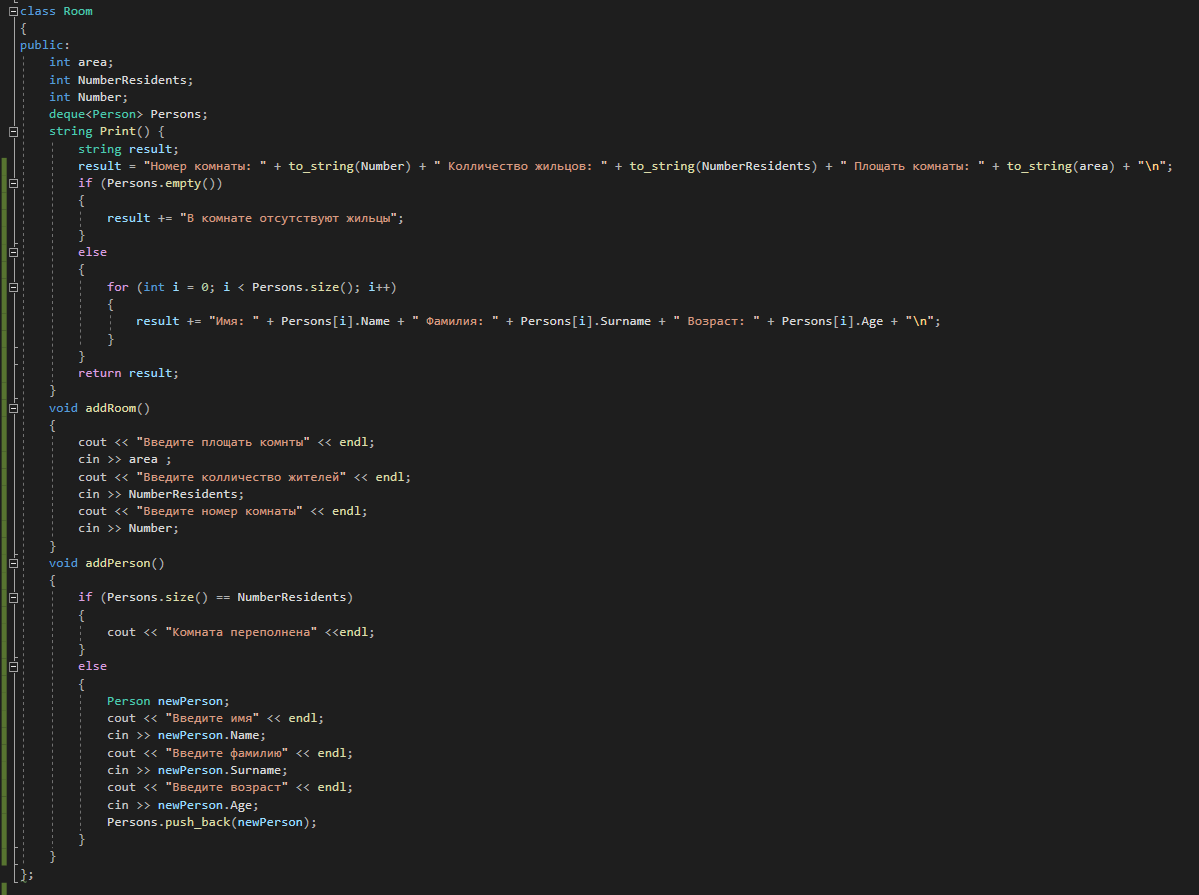
Room – вокруг данного класса строиться весь функционал программы. Данный класс содержит данные комнаты и студентов заселённые в комнату(рис 2.2.2).

В данном классе отсутствуют конструкторы.

Существуют содержит 4 переменных. С помощью трёх из их используются для получения информации о комнате. Четвертая переменная — это список студентов в конкретной комнате.

В данном классе реализована 3 метода.

1. **String** Print() – выводит данные объекта класса на консоль.
2. **Void** addRoom() – присваивает комнате значения.
3. **Void** addPerson() – добавляет студента в комнату.



Метод string **Print:**

Данный метод возвращает строковое значение, в котором содержаться все данные, содержащиеся в объекте класса, для последующего вывода данных на консоль.

## 2.3. Результаты работы программы

Результат работы данной программы являться либо вывод ранее введённых данных на консоль. Либо вывод тех же данных только в изменённом виде.

# 3.РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Вначале пользователю необходимо выбрать пункт, который будет выполняться. Выбор происходит за счёт ввода пользователем в консоль значения от 1 до 3(рис 3.1).

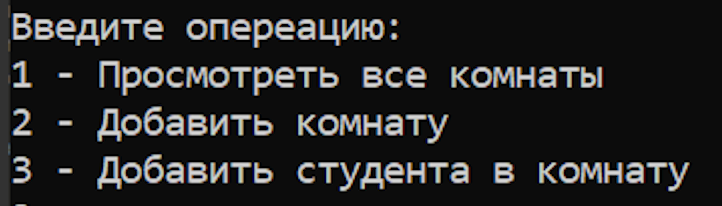


Рис 3.1

При выборе пункта №1 на консоль выводиться данные комнаты (Номер комнаты, площадь, количество мест) и выводит данные студентов, заселённых в комнату (рис 3.2).

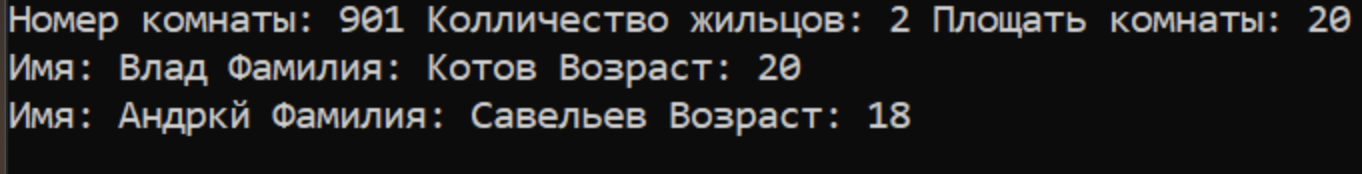


Рис 3.2

При выборе пункта №2 происходит добавление комнаты в список. Необходима ввести данные комнаты (площадь, количество жителей, номер комнаты) (рис 3.3)

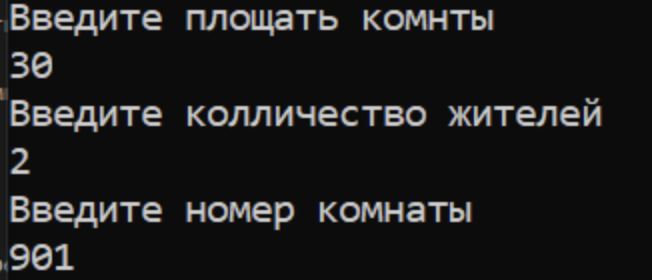


Рис 3.3

При выборе пункта №3 происходит добавление в комнату студента. Вначале происходит выбор комнаты, в которую будет происходит добавление студента. После выбора комнаты происходит ввод данных студента, заселяемого в комнату (рис 3.4). В случае если данной комнаты не существует то на консоль выводиться надпись “Комната переполнена”.

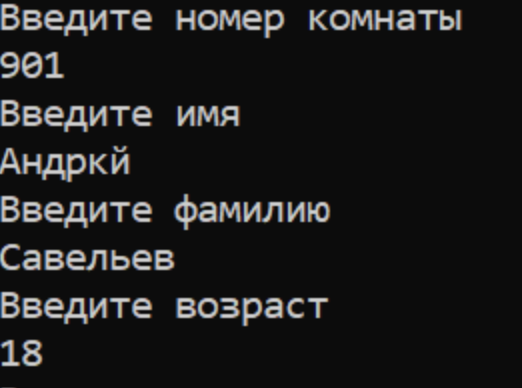


Рис 3.4

# ЗАКЛЮЧЕНИЕ

Была поставлена задача разработать программный продукт «Учет наличия комнат, их площадей̆ и прописанных в них жильцов», предназначенный для использования работниками общежития и деканата.

Были реализованы задачи:

1. Взаимодействие с базой студентов.
2. Добавление комнат и студентов.
3. Учёт студентов в комнату.

Цель была выполнены, программный продукт имеет следующие преимущества:

1. Работа программного продукта происходит быстро и максимально эффективно. Программа удовлетворяет запросам пользователя.
2. Имеет возможность пополнять базу данных, для учёта студентов в комнату.
3. Так же программы имеет высокую оптимизацию. Из-за этого программа работает весьма быстра и запускается даже на самом слабом устройстве.

Так же данный программный продукт выгодно отличается от других представленных на рынке, возможностью не только просматривать, но и добавлять студентов и комнаты, а так же в программном продукте интуитивным интерфейс.

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

# СПИСОК ЛИТЕРАТУРЫ

1. Коплиен Дж. Мультипарадигменное проектирование для С++ / Коплиен Дж. – Питер, 2005.
2. 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.
3. Объектно-ориентированный анализ и проектирование с примерами приложений / Гради Буч [и др.]. – 3-е изд. – М.: ООО «И.Д. Вильямс», 2008. – 720 с.
4. Стивен Прата Язык программирования C++ / Стивен Прата – Вильямс, 2012.
5. Страуструп Б. Программирование. Принципы и практика с использованием C++ / Страуструп Б. – 2-е изд. – Вильямс, 2016.
6. Model-View-Presenter [Электронный ресурс]. – Википедия, 2020. – Режим доступа: https://ru.wikipedia.org/wiki/Model-View-Presenter. – Дата доступа: 22.04.2020.
7. SFML [Электронный ресурс]. – SFML, 2020. – Режим доступа: https://www.sfml-dev.org. – Дата доступа: 22.04.2020.

# ПРИЛОЖЕНИЕ

#include <string>

#include <deque>

#include <iostream>

#include <Windows.h>

using namespace std;

class Person

{

public:

string Name;

string Surname;

string Age;

};

class Room

{

public:

int area;

int NumberResidents;

int Number;

deque<Person> Persons;

string Print() {

string result;

result = "Номер комнаты: " + to\_string(Number) + " Колличество жильцов: " + to\_string(NumberResidents) + " Площать комнаты: " + to\_string(area) + "\n";

if (Persons.empty())

{

result += "В комнате отсутствуют жильцы";

}

else

{

for (int i = 0; i < Persons.size(); i++)

{

result += "Имя: " + Persons[i].Name + " Фамилия: " + Persons[i].Surname + " Возраст: " + Persons[i].Age + "\n";

}

}

return result;

}

void addRoom()

{

cout << "Введите площать комнты" << endl;

cin >> area ;

cout << "Введите колличество жителей" << endl;

cin >> NumberResidents;

cout << "Введите номер комнаты" << endl;

cin >> Number;

}

void addPerson()

{

if (Persons.size() == NumberResidents)

{

cout << "Комната переполнена" <<endl;

}

else

{

Person newPerson;

cout << "Введите имя" << endl;

cin >> newPerson.Name;

cout << "Введите фамилию" << endl;

cin >> newPerson.Surname;

cout << "Введите возраст" << endl;

cin >> newPerson.Age;

Persons.push\_back(newPerson);

}

}

};

void PrintDeque(deque<Room> Rooms)

{

for (int i = 0; i < Rooms.size(); i++)

{

cout << Rooms[i].Print() << endl;;

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "Russian");

deque<Room> Rooms;

while (true)

{

Room room;

int number;

int numberRoom;

int numberRoomDel;

int check = 0;

cout << "Введите опереацию:" << endl;

cout << "1 - Просмотреть все комнаты" << endl;

cout << "2 - Добавить комнату" << endl;

cout << "3 - Добавить студента в комнату" << endl;

cin >> number;

switch (number)

{

case 1:

PrintDeque(Rooms);

break;

case 2:

room.addRoom();

Rooms.push\_back(room);

break;

case 3:

cout << "Введите номер комнаты" << endl;

cin >> numberRoom;

for (int i = 0; i < Rooms.size(); i++)

{

if (Rooms[i].Number == numberRoom)

{

Rooms[i].addPerson();

check = 1;

}

}

if (check == 0)

{

cout << "Комната не существует" <<endl;

}

break;

default:

cout << "Неверно выедена операция";

break;

}

}

}