

ТЕХНИЧЕСКОЕ ЗАДАНИЕ ПО ПРОЕКТУ ОБЩЕЖИТИЕ БГУИР v1.0

Шинкевич Г. С.
гр. 653501

Обзор программы

Программа Общежитие БГУИР предназначена для автоматизации управления заселением студентов в общежитие. Программа позволяет сохранять всю информацию о жильцах в файл, а в дальнейшем читать из него.

Программа рассчитана на пользователя, владеющего русским языком. Программа имеет интуитивно понятный интерфейс и может быть использована даже вашей бабушкой.

Структура данных

1. Класс для хранения информации о жильце (студенте):

```
class Student
{
    string name; //Имя
    string surname; // Фамилия
    string patronymic; // Отчество
    int faculty; // Факультет
    int course; // Курс
    int group; // Группа
    vector<string> violations; // Нарушения
    int room; // Комната
    int id; // ID

public:
    Student(); // Конструктор
    void SetName(string); // Метод установки имени
    void SetSurname(string); // Метод установки фамилии
    void SetPatronymic(string); // Метод установки отчества
    void SetFaculty(int); // Метод установки номера факультета
    string FacultyName(int); // Метод трансляции номера факультета в название
    void SetCourse(int); // Метод установки курса
    void SetGroup(int); // Метод установки группы
    void SetViolations(vector<string>); // Метод установки нарушений
    void SetRoom(int); // Метод установки комнаты
    void SetId(int); // Метод установки ID
    string GetName() { return name; } // Метод получения имени
    string GetSurname() { return surname; } // Метод получения фамилии
    string GetPatronymic() { return patronymic; } // Метод получения отчества
    int GetFaculty() { return faculty; } // Метод получения факультета
    int GetCourse() { return course; } // Метод получения курса
    int GetGroup() { return group; } // Метод получения группы
    vector<string> GetViolations() { return violations; } // Метод получения нарушений
    int GetRoom() { return room; } // Метод получения комнаты
    int GetId() { return id; } // Метод получения ID
};

Student::Student()
{
```

```

        name = "";
        surname = "";
        patronymic = "";
        faculty = 0;
        course = 0;
        group = 0;
        violations = {};
        room = 0;
        id = 0;
    }

    string Student::FacultyName(int faculty)
    {
        switch (faculty)
        {
            case 1: return "ФКП"; break;
            case 2: return "ФИТУ"; break;
            case 3: return "ФРЭ"; break;
            case 4: return "ФКСиС"; break;
            case 5: return "ФТК"; break;
            case 6: return "ИЭФ"; break;
            default: return "Военка"; break;
        }
    }

    void Student::SetName(string Name)
    {
        name = Name;
    }

    void Student::SetSurname(string Surname)
    {
        surname = Surname;
    }

    void Student::SetPatronymic(string Patronymic)
    {
        patronymic = Patronymic;
    }

    void Student::SetFaculty(int Faculty)
    {
        faculty = Faculty;
    }

    void Student::SetCourse(int Course)
    {
        course = Course;
    }

    void Student::SetGroup(int Group)
    {
        group = Group;
    }

    void Student::SetViolations(vector<string> Violations)

```

```

{
    violations = Violations;
}

void Student::SetRoom(int Room)
{
    room = Room;
}

void Student::SetId(int Id)
{
    id = Id;
}

```

Архитектура системы

Работа с программой организована в виде навигации по различным пунктам меню.

Карта меню:

- 1) Прочитать из файла.
После выбора данного пункта меню пользователю предлагается ввести имя файла. Если файл с данным именем существует и соответствует нужному формату, он будет открыт.
- 2) Заселить.
После выбора данного пункта меню пользователю предлагается заполнить данные о новом жильце (имя, фамилия, отчество, факультет (определяет доступные комнаты), курс, группа, комната (доступность комнат определяется факультетом)).
После заполнения данных жильцу автоматически будет присвоен ID и жилец будет заселен.
- 3) Выселить.
После выбора данного пункта меню пользователю предлагается ввести ID жильца, которого он хочет выселить. После подтверждения жилец будет выселен.
- 4) Изменить данные.
После выбора данного пункта меню пользователю предлагается ввести ID жильца, данные которого он хочет изменить. После ввода ID пользователю предлагается выбрать поле для изменения (имя, фамилия, отчество, факультет, курс, группа, комната, нарушения).
При выборе поля «Нарушения» пользователь получает возможность просмотреть список нарушений, добавить или удалить нарушение. Если количество нарушений больше 3, программа выводит предупреждение о необходимости выселения.
После ввода новых данных, редактирование будет завершено.
- 5) Поиск.
После выбора данного пункта меню пользователю предлагается выбрать поле, по которому будет осуществлён поиск (ID, фамилия, комната, факультет, курс, группа). После пользователь должен ввести данные соответствующего поля.
При поиске по первым трём полям, результаты будут отображены в виде карточек, при поиске по последним трём полям — в табличном виде.

- 6) Список жильцов.
После выбора данного пункта меню отобразится полный список жильцов в табличном виде.
- 7) Отчёт.
После выбора данного пункта меню отобразится общее количество жильцов и свободных мест.
У пользователя есть возможность просмотреть отчёт для каждого факультета.
- 8) Список служащих.
После выбора данного пункта меню отобразится список служащих и телефоны для связи.
- 9) Записать в файл.
После выбора данного пункта меню пользователю предлагается ввести имя файла.
Если запись в этот файл разрешена, он будет записан.
- 10) Выход.
После выбора данного пункта меню программа завершает свою работу.

Системные требования

Минимальные требования: процессор — 3 ГГц, оперативная память — 8 Гб, 50 Гб свободного пространства на диске (желательно, SSD). Операционные системы: Windows 10 и новее.