

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Новосибирский государственный университет экономики и управления «НИНХ»**

**(ФГБОУ ВО «НГУЭУ», НГУЭУ)**

Кафедра информационных технологий

**РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА**

Дисциплина: Прикладные информационные технологии

Ф.И.О студента: Головач Елизавета Сергеевна

Направление: 09.03.02 Информационные системы и технологии

Номер группы: ИС101

Номер зачетной книжки: 212820

Номер варианта расчетно-графической работы: 10

Проверил: Ковригин Алексей Викторович

Новосибирск 2023

СОДЕРЖАНИЕ

[1.Задача 3](#_Toc124630007)

[2. Описание проекта 4](#_Toc124630008)

[3. Вид приложения в консоли 6](#_Toc124630009)

# 1.Задача

Создание программного приложения «Управления проектами»

Необходимо создать программу по управлению объектами данных по указанной теме с использованием ООП и реализовывать изученные паттерны.

Необходимо создать минимум 3 объекта сущности разных видов и указать взаимодействие между объектами. Реализовать интерфейсы ввод, вывода и редактирования сущностей. Реализовать ручной ввод сущностей, загрузку из файла и сохранения в файл, удаление выбранных сущностей.

Данные между запусками программы хранить в структурированном текстовом файле.

# 2. Описание проекта

**Описание проекта.**

По заданной теме проекта я решила создать аналог проектной программы, которая будет использоваться внутри компании.

Программа включает в себя класс компании, класс сотрудника, класс проекта.

**Описание классов.**

Class EmployeeBase является классом компании, содержит переменные длины списка сотрудников, динамический массив класса Employee – список сотрудников.

Содержит следующие методы:

* firstEmployee () – создание первого сотрудника;
* firstEmployee (firstEmployee (\*f) - перегруженный метод для создания первого сотрудника при чтении из файла
* newEmployee ()- внесение сотрудника в базу, когда сотрудников> 1
* init () - инициализация первого сотрудника, либо же нового (только по вызову пользователя, во время сессии)
* void enter () - метод входа в аккаунт сотрудника
* write\_to\_file () - метод записи данных в файл, вызывается в конце работы программы

Class Employee – класс сотрудника, содержит private login и password сотрудника, public name, section (отдел сотрудника) а также список проектов UserProject.

Содержит следующие методы:

* firstProject () - создание первого проекта
* firstProject (ifstream \*f) - перегруженный метод создания первого проекта, используется только пи чтении данных из файла
* newProject () - внесение проекта в базу, когда проектов> = 1
* newProject (ifstream \*f) - перегруженный метод создания списка проектов, когда проектов> 1, используется только пи чтении данных из файла
* removeProjectList (int const \*number) - метод удаления любого номера проекта из списка
* setInf () - метод, задающий параметры сотрудника
* setInf (ifstream \*f) - перегруженный метод задания параметров сотрудника, при чтении из файла
* check (string const \*l, string const \*p) - метод, отвечающий за проверку введеных имени, логина и пароля на корректность
* userAccount () - метод пользовательского аккаунта, который дает доступ в аккаунт сотрудника
* read\_file (ifstream \*f) - прикладной метод для чтения данных из файла, вызывается внутри конструктора класса EmployeeBase, работает только в начале программы

Class UserProject – класс, представляющий проекты сотрудника, их количество, содержание, название, имя и отдел создателя проекта. По умолчанию все поля “NULL”. Содержит методы:

* createProject (string const \*n, string const \*d) - метод создания проекта, он вызывается только в случае создания проекта, вызванного пользователем во время работы программы
* createProject (ifstream \*f, string const \*n, string const \*d) - перегруженный метод создания проекта, используется при запуске программы, читает файл
* getProjectList (int const \*number) - метод вывода информации об имеющемся проекте, number отвечает за номер проекта

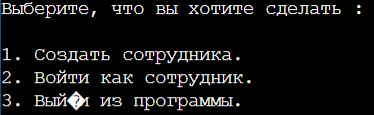
**Минусы проекта.**

* 1. Каждый при завершении работы вызывается. write\_to\_file (), который полностью перезаписывает файл. Необходимо создать отдельные методы/функции для оптимизированной записи данных. Но, возможно, это недостаток исключительно формата .txt. Возможно, в .xlsx удалось бы этого избежать;
  2. Много методов для создания сотрудника/проекта. Стоит сделать единый метод создания сотрудника/проекта вместо. first Project (), new Project () и т.д.;
  3. Загруженная система чтения данных из файла. Идет множественное вложение в различные методы других классов. Однако это обеспечивает взаимодействие и целостность каждого класса, с возможностью реализации protected и private данных и методов. В противном случае, пришлось бы все параметры классов делать public, что являлось бы небезопасным.

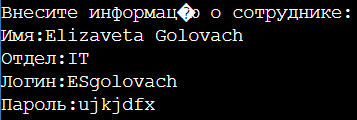
# 3. Вид приложения в консоли

**Скриншоты работы программы.**

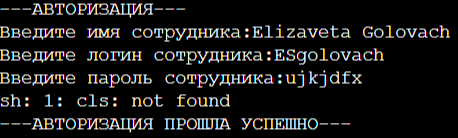
**Меню**

****

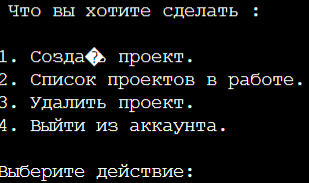
**Создание сотрудника**

****

**Вход в аккаунт**

****

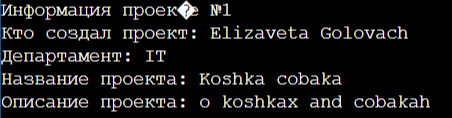
**Меню сотрудника**

****

**Создание проекта**

****

**Просмотр списка проектов, когда проекты есть**

****

**Удаление проекта, когда проекты есть**

****

**Удаление проекта, когда проектов нет**

****

**Просмотр списка проектов, когда проектов нет**

****