

Содержание

Введение	2
1. Разработка проекта информационной системы	3
1.1. Анализ предметной области	3
1.2. Эскизное проектирование	4
1.3. Техническое задание	5
1.4. Выбор программных средств реализации	10
2. Проектирование системы	11
2.1. Проектирование пользовательского интерфейса	11
2.2. Хранение информации	16
Заключение	17
Список использованных источников	Error! Bookmark not defined.
Приложение	18

Введение

В последнее время все большее время уделяется хорошему опыту взаимодействия пользователя с функциями системы, что напрямую зависит от интерфейса.

Интерфейс человеко-машинный (интерфейс) — комплекс технических и информационно-программных средств, посредством которых осуществляется диалоговый режим взаимодействия человека-оператора и вычислительных средств компьютеризированного образца. [1]

Главная цель интерфейса – упростить и ускорить взаимодействие со сложными техническими объектами, а также выполнять задачу помощника, направляющего пользователя. Данная работа актуальна, поскольку данная область широко развивается и используется во всех направлениях.

Цель данной курсовой работы – создать приложение для комфортного взаимодействия со списками студентов и учета их успеваемости. Задачи, необходимые для достижения поставленной цели:

1. анализ предметной области;
2. определение функциональных требований;
3. проектирование интерфейса системы;
4. создание системы;

1. Разработка проекта информационной системы

1.1. Анализ предметной области

В целях ускорения и повышения эффективности управления учебным процессом система должна позволять взаимодействовать со списками студентов, просматривать и изменять информацию.

База данных содержит всю информацию о студентах и курсах одной кафедры. Информация о студентах включает в себя уникальный номер, ФИО, пол студента, дата рождения, номер группы, номер курса, список оценок и адрес прописки. Информация о группе содержит номер и ФИО куратора. Информация о курсе содержит номер и список предметов. Информация о предмете содержит название, ФИО преподавателя и количество часов.

Система должна позволять вошедшим в систему работникам деканата после выбора из списка курса получать список групп, а также просматривать списки студентов. Она также должна позволять выбрать студента из списка, чтобы отчислить или перевести его в другую группу или на другой курс.

Также система должна позволять управлять текущими и итоговыми оценками, вести учет успеваемости по итогу сданной студентом сессии. Позволяет вести информацию о предметах на курсе и менять ее.

Создание групп и занесение данных по студентам выполняется работниками деканата. Операция добавления студентов состоит из выбора курса и группы, и заполнения данных о новом студенте. Система также должна позволять менять информацию о группах и студентах.

1.2.Эскизное проектирование

Составление глоссария проекта:

Термин	Значение
Курс	Система занятий, рассчитанная на определенный период времени и состоящая из образовательных дисциплин.
Деканат	Организационный центр высшего учебного заведения по управлению факультетом, возглавляемый деканом
Кафедра	Обучающее или научное подразделение факультета, ведущее подготовку студентов или ведущее научную работу в рамках какой-то одной специализации
Оценка	Оценка, полученная студентом за конкретный предмет.
Табель успеваемости	Все оценки за все курсы, полученные студентом в данном семестре.
Студент	Личность, проходящая обучение в университете.
Дисциплина	Определенный предмет, который изучается в университете
Отчисление	Вариант исключения учащегося из образовательного заведения на конкретном основании
Перевод студента	Процедура по переходу на очередной этап обучения (следующий курс) или по группы (по инициативе учащегося и согласованию с администрации)
Группа	Определенное число лиц, изучающих одно и то же на протяжении одинакового для всех периода

1.3. Техническое задание

Действующие лица:

Работник деканата — заполняет данные и студентах, добавляет новых, изменяет информацию о имеющихся студентах, переводит их в другие группы или курсы.

Исходя из потребностей действующих лиц, выделяются следующие варианты использования:

- Войти в систему
- Добавить
- Перевод в другую группу
- Перевод на другой курс
- Просмотреть таблицу успеваемости.
- Проставить оценки
- Изменить

Вариант использования "Войти в систему"

Краткое описание:

Данный вариант использования описывает вход пользователя в систему “Деканат”.

Основной поток событий:

Данный вариант использования начинает выполняться, когда пользователь хочет войти в систему. Система запрашивает имя пользователя и пароль. Пользователь вводит имя и пароль. Система подтверждает имя и пароль, после чего открывается доступ в систему.

Альтернативные потоки:

Неправильное имя/пароль: если во время выполнения основного потока обнаружится, что пользователь ввел неправильное имя и/или пароль, система выводит сообщение об ошибке. Пользователь может вернуться к началу основного потока или отказаться от входа в систему, при этом выполнение варианта использования завершается.

Вариант использования “Добавить”

Краткое описание

Работник деканата вносит в базу данных информацию о новом студенте, курсе или группе

Основной поток событий:

Открыт список курсов

При выборе данного варианта использования система создает новый объект курс, предлагает пользователю заполнить номер. При сохранении курс добавляется в общий список. Если курсов в общей сложности стало равно 4, то возможность добавления курса пропадает.

Открыт список групп

При выборе данного варианта использования система создает новый объект группа, предлагает пользователю заполнить номер и ФИО куратора. При сохранении группа добавляется в список выбранного курса.

Открыт список студентов

При выборе данного варианта использования система создает новый объект студент, присваивает ему уникальный номер и предлагает пользователю заполнить все соответствующие данные: ФИО, пол студента, дата рождения, адрес прописки. Далее пользователь может сохранить введенную информацию. После этого студент появиться в общем списке.

Альтернативный поток событий:

Если какое-то из полей осталось пустым, выводится предупреждение о незаполненном поле.

Предусловия:

Перед началом выполнения данного варианта использования работник деканата должен войти в систему и открыть список, в который он хочет что-то добавить.

Вариант использования “Изменить”

Краткое описание:

Данный вариант использования описывает процесс изменения информации в системе.

Основной поток событий:

Открыт список курсов:

Система предлагает выбрать курс и показывает список дисциплин на курсе. Затем запрашивает требуемое действие (добавить новый предмет, удалить существующий). Когда работник указывает действие, выполняется один из подчиненных потоков.

Добавить дисциплину.

Система предлагает выбрать из списка предметов новый предмет. Если в списке нет подходящего предмета система позволяет добавить новый предмет. При этом работнику предлагается ввести ID и название предмета, ФИО преподавателя и количество часов. Или же изменить информацию о существующем предмете или удалить предмет из всей системы.

После выбора предмета, система добавляет предмет список предметов курса, сохраняет введенные данные и завершает подчиненный поток.

Удалить дисциплину

Система предлагает выбрать предмет, который нужно удалить из списка вместе с выставленными по нему оценками.

Из всей системы предмет не удаляется, а только из списка предметов выбранного курса

Открыт список групп:

Выводятся поля с информацией группы с возможностью изменения. После сохранения информация о группе меняется.

Открыт список студентов:

Выводятся поля с информацией о студенте с возможностью изменения, после сохранения информация о студенте меняется.

Завершить

Вариант использования завершается.

Предусловия:

Перед началом выполнения данного варианта использования работник деканата должен войти в систему, а также выбрать объект для изменения.

Вариант использования “ Перевод в другую группу”

Краткое описание

Система позволяет выбранного студента перевести в другую группу.

Основной поток событий:

Система позволяет выбрать другую группу того же курса. После сохранения, студент перемещается в список выбранной группы.

Альтернативный поток событий

Если другая группа не выбрана, студент останется в той же группе.

Предусловия:

Перед началом выполнения данного варианта использования работник деканата должен войти в систему. Работник деканата должен выбрать одного из студента в списке и запустить процесс изменения информации о студенте.

Вариант использования “ Перевод на другой курс”

Краткое описание

Система позволяет перевести все курсы на следующий.

Основной поток событий:

После выбора данного варианта использования система производит диагностику наличия студентов с незаполненными оценками. В случае получения подтверждения удаляет студентов с недостаточными баллами и переводит номер группы на следующий.

Альтернативный поток событий

Если подтверждения исключения студентов не получено, вариант использования завершится.

Предусловия:

Перед началом выполнения данного варианта использования работник деканата должен войти в систему.

Вариант использования “Просмотреть табель успеваемости.”

Краткое описание

Система показывает оценки за курс выбранного студента.

Основной поток событий:

При выполнении данного варианта использования показывается список предметов и список оценок за данные оценки.

Предусловия:

Перед началом выполнения данного варианта использования работник деканата должен войти в систему. Выбрать студента из списка.

Вариант использования “Проставить оценки”

Краткое описание:

Данный вариант использования описывает процесс выставление итоговых оценок студентам.

Основной поток событий:

Выводится список предметов студента и запрашивается ввод оценок. Затем данные сохраняются.

Предусловия:

Перед началом выполнения данного варианта использования работник деканата должен войти в систему, выбрать в таблице студента и запустить вариант использования “Просмотреть табель успеваемости.”.

1.4. Выбор программных средств реализации

Система реализована в среде IntelliJ IDEA 2021.3.2 (Ultimate Edition) на языке Java.

Java - один из самых важных и широко применяемых языков программирования в мире на протяжении многих лет, предназначенный для разработки веб-ориентированных приложений. [3]

Этот язык непрерывно адаптируется к изменениям в среде программирования и подходам к написанию программ, и поэтому продолжает оставаться столь успешным.

JavaFX — это библиотека для создания на языке Java приложений с графическим интерфейсом, насыщенных графикой, анимацией и интерактивными элементами.[2]

Компоновка элементов управления может быть осуществлена программным способом, как это сделано в библиотеке Swing, определена на декларативном, непроцедурном языке с помощью CSS и языка XML или выполнена с помощью визуального конструктора SceneBuilder.

Таким образом, возможности данной среды программирования полностью соответствуют требованиям при разработке программного комплекса (поддерживает язык Java, имеет полный набор инструментов для создания интерфейса, написания и отладки кода).

2. Проектирование системы

2.1. Проектирование пользовательского интерфейса

В приложении предполагается наличие окна авторизации, основного меню и побочных форм для заполнения и изменения информации. Основные цвета системы приведены на Рисунке 1.

Рис. 1. Основные цвета оформления системы “Деканат”

При открытии приложения отображается экран авторизации, где нужно ввести логин и пароль. (Рис. 2)

Рис. 2. Окно авторизации системы “Деканат”

Если процесс авторизации пройден успешно, на экран выводится главная форма с меню, таблицей и панелью с кнопками. (Рис. 3)

Рис. 3. Главное окно системы “Деканат”

Меню состоит из разделов "Файл" и "Помощь".

Раздел "Файл" состоит из следующих вариантов (Рис. 4):

Рис. 4. Открытый пункт меню “Файл”

- "Сохранить"- перезаписываются файлы с данными
- "Информация о курсах" - открывается изначальная таблица со списком всех курсов
- "Открыть" - при наведении выводятся кнопки перехода на конкретные группы и курсы. (Рис. 4)
- "Изменить" - открывается окно изменения выбранного объекта, если объект не выбран выводится предупреждение.
- "Добавить" - открывается окно внесения информации о новом объекте.
- "Закрыть" - приложение закрывается. За данным действием закреплено мнемоническое правило alt+X. При использовании запрашивает

подтверждения для выхода (Рис. 5) и сохранения внесенных изменений (Рис. 6).

Рис. 5. Всплывающее окно, запрашивающее подтверждением выхода из системы.

Рис. 6. Всплывающее окно, запрашивающее подтверждением сохранение внесенных изменений.

Вид добавляемого или изменяемого объекта определяется открытой в данный момент таблицей.

При наведении на вариант меню "Помощь" выводиться вариант "О программе". На Рисунке 7 показано содержание всплывающего окна.

Рис. 7. Всплывающее окно “Информация о программе”

Изначально открывается таблица с курсами, чтобы открыть список групп на курсе нужно выбрать необходимый курс и нажать кнопку "Открыть" на панели с кнопками, дважды щелкнуть на выбранный курс или выбрать нужный курс/группу в меню. Аналогично открывается список студентов в группе.

Панель с кнопками располагается внизу главной формы и содержит слева номер открытого(ой) курса/группы, а справа кнопки управления.

В зависимости от открытого раздела кнопки будут отличаться.

Если открыт общий список курсов будут доступны кнопки “Перевести на другой курс”, “Изменить”, “Добавить”, “Открыть”.

Если открыт список групп будут доступны кнопки “Назад”, “Перевести на другой курс”, “Изменить”, “Добавить”, “Открыть”. (Рис. 8)

Рис. 8 Таблица со списком групп одного из курсов

Если открыт список студентов будут доступны кнопки “Назад”, “Перевести на другой курс”, “Изменить”, “Добавить”, “Открыть”. (Рис. 9)

Рис. 9 Таблица со списком студентов одной из групп

Кнопка “Назад” открывает предыдущий раздел, родителя. В случае курсом – список курсов.

Кнопка “Перевести на другой курс” приводит к смене номеров курса на следующий, исключении студентов с недостаточным количеством баллов и удалением из системы выпускного курса (4 курс считается последним). Если при переводе система обнаружила студентов с 0 баллов, всплывает окно, запрашивающее подтверждение исключения данных студентов и дальнейшего перевода остальных на следующий курс. (Рис. 10)

Рис. 10. Всплывающее окно с подтверждением удаления студентов.

В случае, если перевод выполнен успешно выводится всплывающее окно, приведенное на Рисунке 11.

Рис. 11. Подтверждение завершения перевода на другой курс.

Кнопка “Просмотреть оценки”, доступная только при открытии списка студентов любой группы открывает форму (Рис. 12), показывающую список предметов с заблокированными для изменения оценками. На форме расположены кнопки “Сохранить” и “Отменить”, отвечающие состоянию введенных данных после закрытия, а также кнопка “Изменить”, которая разблокирует поля с оценками для изменения.

Рис. 12. Табель с оценками студента

Кнопки “Изменить” и “Добавить” выполняет аналогичное действие кнопок “Изменить” и “Добавить” в меню формы. Однако их действие в зависимости от открытой таблицы отличаются.

Если открыт список курсов, то кнопка “Добавить” позволяет внести новый номер курса.

Если открыт список курсов, то кнопка “Изменить” выведет форму с изменением списка предметов (Рис. 13), где можно удалить предмет из списка предметов данного курса вместе с выставленными по нему оценками, сохранить и отменить изменения, а также добавить в список новые предметы.

Рис. 13. Список предметов на курсе.

При добавлении нового предмета выводиться форма, изображенная на Рисунке 14 с выведенными преподаваемыми дисциплинами. Кнопка “Выбрать” закроет форму со всеми дисциплинами и добавит ее в список предметов курса, а в случае, если предмет не выбран, система вызовет всплывающее окно с предупреждением. Кнопка “Отмена” закроет форму без добавления нового предмета.

Рис. 14. Список всех дисциплин

Список всех дисциплин можно менять удаляя, добавляя новые, изменяя информацию о существующих. При изменении выводиться форма с заполненной информацией в случае с изменением и пустыми полями, если производиться добавление нового предмета. (Рис. 15)

Рис. 15. Форма изменения информации о предмете.

Если открыт список групп, выводиться форма с заполненной информацией в случае с изменением и пустыми полями, если производиться добавление новой группы. (Рис. 16)

Рис. 16. Форма изменения информации о группе.

Аналогично со списком студентов. (Рис. 17), но также есть дополнительная кнопка перевода студента в другую группу на том же курсе. Это можно сделать, разблокировав с помощью гипертекста “Изменить” ComboBox с номерами групп на курсе. Если группа изменена и нажата кнопка сохранить, студент перемещается в другой список.

Рис. 17. Форма изменения информации о студенте.

Если при добавлении или изменении остались незаполненные или неправильно заполненные поля, система не даст сохранить введенные данные и выдаст предупреждение.

Рис. 18. Предупреждение о некорректности внесенных данных.

2.2.Хранение информации

В качестве способа хранения данных для системы используются стандартный текстовый документ. На Рисунках 19 – 24 приведено, в каком виде данные хранятся.

В системе информация при каждом запуске переноситься из файлов и заполняется во внутренние списки с помощью вложенной иерархии. Приложение 1 – 5 демонстрирует классы с полями, хранящие информацию во время использования приложения.

Рис. 19. Пример хранения данных о дисциплинах на курсах

Рис. 20. Пример хранения данных о дисциплинах на курсах

Рис. 21. Пример хранения данных о дисциплинах на курсах

Рис. 22. Пример хранения данных о группах на курсах

Рис. 23. Пример хранения данных о студентах

Рис. 24. Пример хранения данных об оценках студентов

Заключение

Подводя итог, было создано приложение "Деканат" для более удобного взаимодействия с большим количеством информации о студентах, группах, курсах и преподаваемых дисциплинах.

В приложении реализован полностью корректно реализован перечисленный в техническом задании функционал, а именно просмотр, изменение, удаление и добавление курсов, групп, студентов, а также изменение списка дисциплин и заполнение табеля с оценками.

Приложение

Приложение 1

```
public class User {  
  
    private final String login;  
  
    private final String password;  
  
    static final String file = "Users.txt";}
```

Приложение 2

```
public class Course {  
  
    private Integer number;  
  
    private ObservableList<Discipline> disciplines;  
  
    private ObservableList<Group> groups;  
  
    private String allDisciplines;}
```

Приложение 3

```
public class Discipline {  
  
    private Integer id;  
  
    private Integer hours;  
  
    private String disciplineName;  
  
    private String FIO_teacher;}
```

Приложение 4

```
public class Group {
```

```
private String groupNumber;  
  
private String groupTutor;  
  
private ObservableList<Student> students;}
```

Приложение 5

```
public class Student{  
  
    private Course course;  
  
    private Integer id;  
  
    private String FIO;  
  
    private String gender;  
  
    private LocalDate dateOfBirthday;  
  
    private String address;  
  
    private ObservableMap<Discipline, Integer> marks;}
```