Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

Лабораторная работа № 4-6

«Проектирование приложений с использованием UML»  
По дисциплине: «Объектное моделирование интеллектуальных систем»

Выполнил:  
Студент 3 курса  
Группы ИИ-23  
Романюк А.П.

Проверила:  
Демидович А.Г.

Брест 2024

**Цель работы:** 1. Изучить содержание типовых этапов объектно-ориентированной разработки программ. 2. Изучить использование диаграмм UML при разработке программ. 3. Выполнить объектно-ориентированный анализ с элементами проектирования и реализации приложения с документированием результатов на языке UML.

**Описание задания:**

1. Тип ЗАПИСЕЙ: г – год, расходы, доходы.

2. Тип ХРАНЕНИЯ СПИСКА записей: б – как “системный” файл.

3. Тип СПИСКА: б – упорядоченный (отсортированный) по заданному полю (полям) записей.

4. Тип КАРКАСА приложения (интерфейса): а – на базе окна типа главное с управлением задачами (прецедентами) с помощью меню;

5. Тип организации АВТОРИЗАЦИИ пользователей: в – с регистрацией и авторизацией пользователей и хранением перечня логинов во внутреннем списке приложения;

6. Типы ОПЕРАЦИЙ со СПИСКОМ: а – создание нового списка с ”автоматическим” заполнением начальным константным содержимым; г – вывод содержимого списка в виде перечня записей;

7. Типы ОПЕРАЦИЙ с ЗАПИСЬЮ: б – редактирование выбранной записи (изменение полей с последующим сохранением или отменой изменений); в – удаление записи(-ей).

**Вариант задания:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Атрибуты варианта** | | | | | | |
| **1** | **2** | **3** | **4\*** | **5** | **6** | **7** |
| 9 | г | б (системный файл) | б | а | в | а, г | б, в |

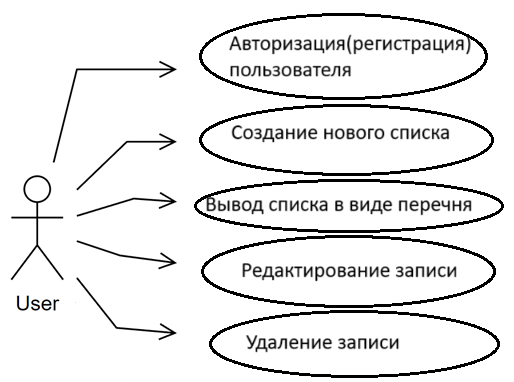
**Перечень этапов:**

|  |
| --- |
| **Выполняемые работы** |
| **ЭТАП 1 – Объектно-ориентированный анализ предметной области** |
| 1. Описание предметной области, автоматизируемых задач (результаты интервью) |
| 2. Выявление и описание прецедентов (построение диаграммы и описание потоков) |
| 3. Прототипирование приложения:  - первоначальное выявление интерфейсных форм (сопутствующих классов);  - дизайн форм;  - описание диаграммы состояний приложения;  - описание диаграмм состояний для прецедентов;  - описание диаграмм состояний для отдельных форм |
| 4. Выявление, первоначальное описание классов, классов предметной области, диаграммы классов предметной области на уровне “первичных” отношений |
| 5. Первичное описание структуры приложения и обработчиков |
| **ЭТАП 2 - Объектно-ориентированное проектирование** |
| 6. Описание диаграмм последовательностей, видов деятельности прецедентов |
| 7. Уточненное описание структуры классов, диаграммы классов предметной области |
| 8. Описание диаграмм видов деятельности методов |
| Реализация интерфейса приложения (макетирование) |
| **ЭТАП 3 – Объектно-ориентированное проектирование и реализация** |
| 10. Описание структуры приложения – диаграммы пакетов, компонентов |
| 11. Описание диаграммы развертывания приложения |
| 12. Реализация приложения, тестирование, документирование |

**1.**

Список записей заданного типа с возможностью осуществления операций как с самим списком, так и с составляющими его записями. Варианты организации списка, типы записей, а также варианты работы с ними приведены выше. Для работы со списком может потребоваться регистрация и авторизация пользователей. Информация (сам список, логины) может храниться в “системном” или произвольных файлах.

**2.**



|  |  |
| --- | --- |
| **Поток** | **Описание** |
| Авторизация(регистрация)  пользователя | Пользователь заходит в программу и для начала работы с ней регистрируется или входит в свою учетную запись. |
| Создание нового списка | Пользователь выбирает опцию создания нового списка. Система создает список заполненный константным содержимым. |
| Вывод списка в виде перечня | Пользователь выбирает опцию вывода содержимого списка. Система отображает содержимое списка в виде перечня записей, упорядоченных по заданному полю (если список был отсортирован). |
| Редактирование записи | Пользователь выбирает опцию редактирования записи списка. Пользователь указывает запись для редактирования. Система выводит выбранную запись в окне для редактирования. |
| Удаление записи(-ей) | Пользователь выбирает опцию удаления записи(-ей). Пользователь указывает, какую запись нужно удалить  Система удаляет указанную(-ые) запись(-и) из списка. |

**3.**

Используем язык программирования C# и Windows Forms для создания оконного интерфейса.

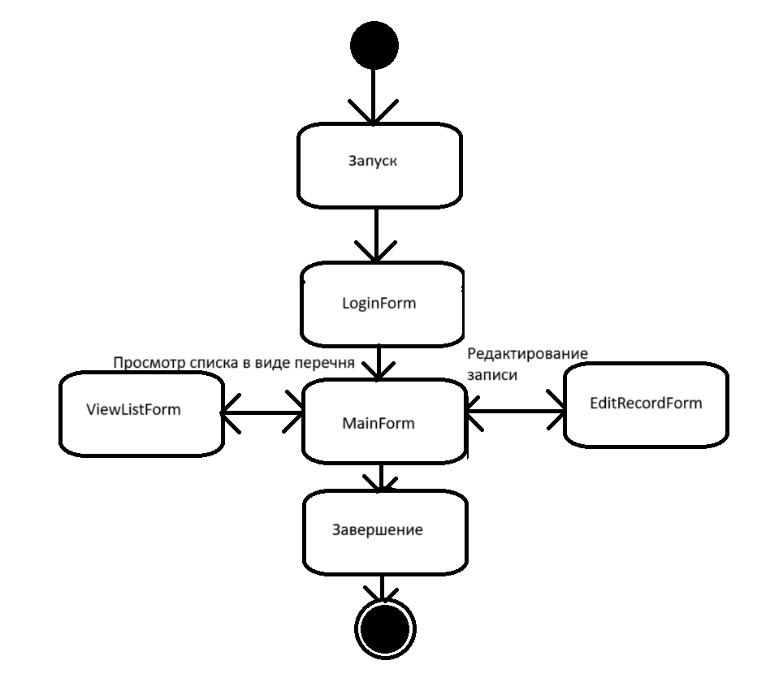
**Интерфейсные формы и сопутствующие классы:**

* + Форма "Главное меню" (MainForm): Отображает выбранный список и содержит кнопки для взаимодействия с ним.
  + Форма "Просмотр записей" (ViewListForm): позволяет пользователю увидеть списоу в виде перечня записей.
  + Форма "Изменение записи" (EditRecordForm): позволяет пользователю изменить выбранную запись.
  + Форма "Логин" (LoginForm): позволяет пользователю зарегистрировать или войти в учетную запись.

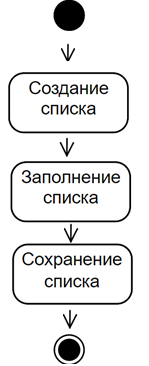
**Дизайн форм:**

* + Форма "Главное меню" (MainForm): содержит список и кнопки для взаимодействия с ним
  + Форма "Просмотр записей" (ViewListForm): Содержит поле в виде текстового вывода.
  + Форма "Изменение записи" (EditRecordForm): Содержит поля текстового ввода для редактирования записи.
  + Форма "Логин" (LoginForm): содержит поля ввода логина и пароля, а также кнопки войти или зарегистрироваться.

**Диаграмма состояний приложения:**



**Диаграммы состояний для прецедентов:**

****

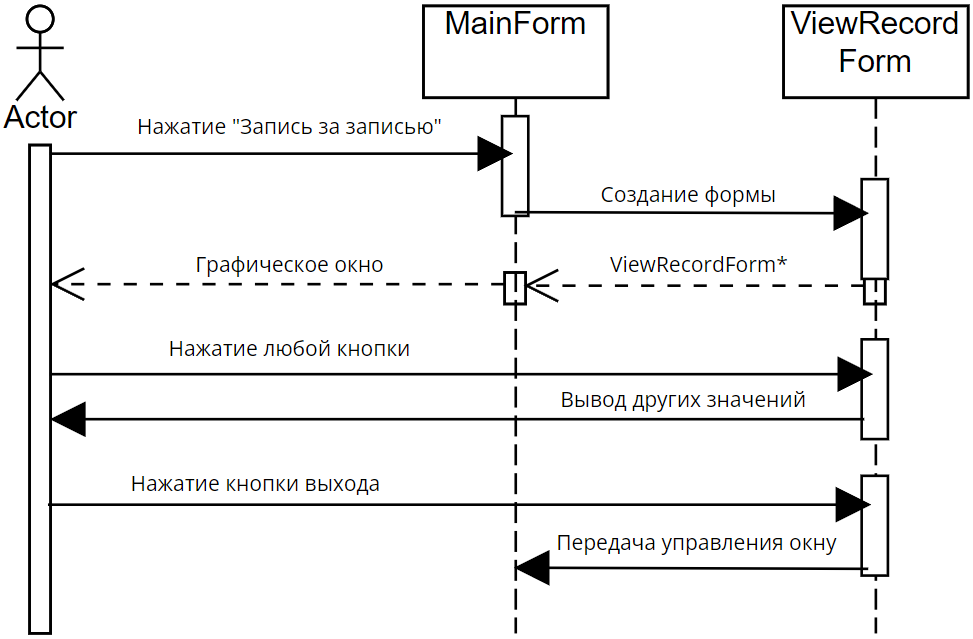
**4.**

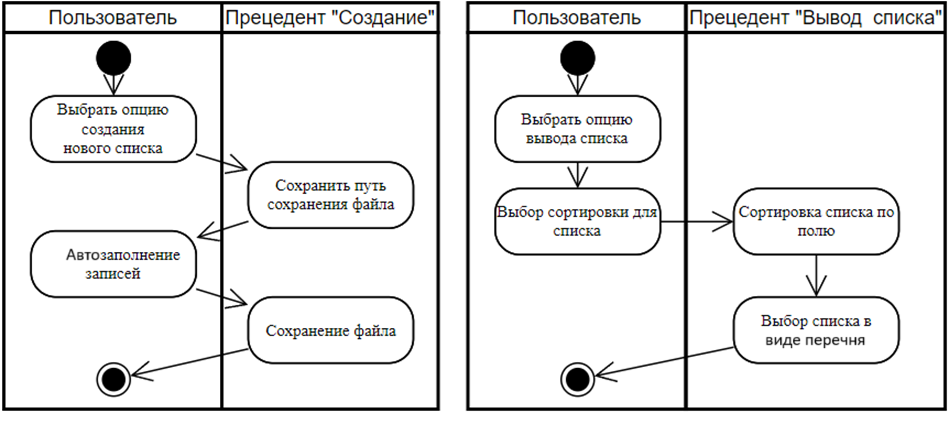
**Первичное описание структуры приложения и обработчиков:**

* MainForm: отображает основное меню с возможностью выбора задач, например, "Создание списка", "Вывод списка", "Просмотр записей", "Поиск записи", "Удаление записи".
* CreateListForm: пользователь выбирает тип записей и тип хранения списка. После выбора происходит создание объекта RecordList.
* FillListForm: пользователь вводит данные для каждой группы, после чего новая запись добавляется в объект RecordList.
* DisplayListForm: отображает список записей в виде перечня, упорядоченного по выбранному полю из объекта RecordList.
* ViewRecordForm: поочередно отображает каждую запись из объекта RecordList, позволяя пользователю просмотреть детали каждой группы.
* SearchRecordForm: пользователь указывает номер или шаблон для поиска записи. Поиск происходит в объекте RecordList, и первая найденная запись отображается на форме.
* DeleteRecordForm: пользователь выбирает метод удаления записи (с конца списка, с начала списка, указанную запись). Запись удаляется из объекта RecordList.

**5.**

**Описание диаграмм последовательностей и видов деятельности прецедентов:**

****

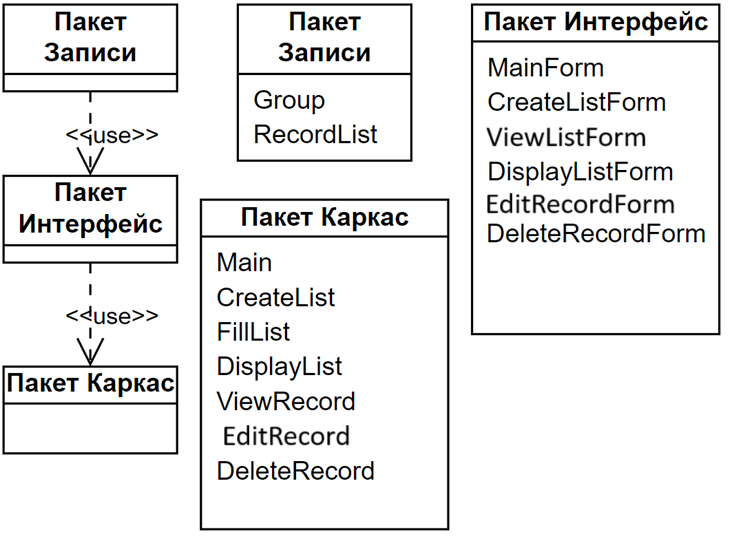
****

**6.**

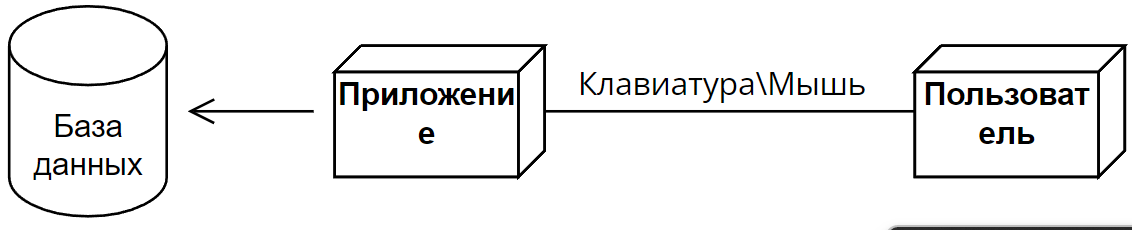
**Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание**

**8.**

****

**9.**

****

**10.**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Автоматически созданное описание\**

**Вывод:** 1. Изучить содержание типовых этапов объектно-ориентированной разработки программ. 2. Изучить использование диаграмм UML при разработке программ. 3. Выполнить объектно-ориентированный анализ с элементами проектирования и реализации приложения с документированием результатов на языке UML.