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-21  
Карагодин Д.Л.

Проверила:  
Демидович А.Г.

Брест 2023

**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: е – группа, староста, численность студентов.

2. Тип ХРАНЕНИЯ СПИСКА записей: в – как произвольный 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: а – без регистрации и авторизации пользователей;

6. Типы ОПЕРАЦИЙ со СПИСКОМ: в – создание нового списка с ”ручным” заполнением начальным содержимым; г – вывод содержимого списка в виде перечня записей; д – просмотр содержимого списка ”запись за записью”; е – поиск первой записи списка (по заданному номеру, по шаблону);

7. Типы ОПЕРАЦИЙ с ЗАПИСЬЮ: в – удаление записи (с конца списка, с начала списка, указанной (-ых) записей, в том числе найденной (-ых) записей).

**Вариант задания:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Атрибуты варианта** | | | | | | |
| **1** | **2** | **3** | **4\*** | **5** | **6** | **7** |
| 5 | е | в (произвольный файл) | б | а | а (без авторизации) | в, д, г, е | в |

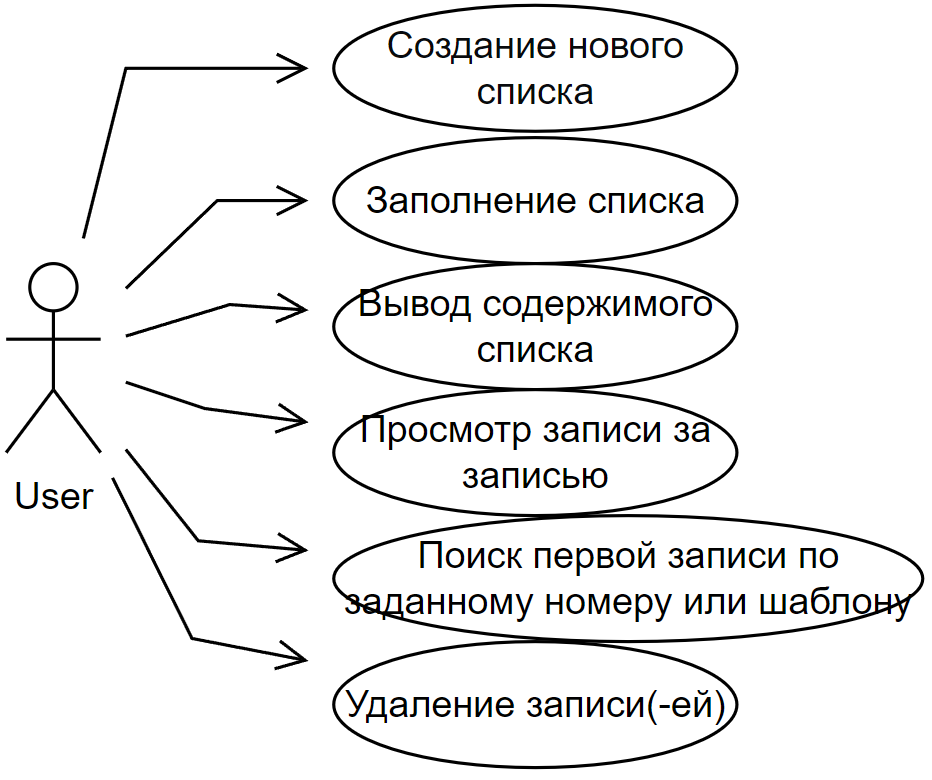
**Перечень этапов:**

|  |
| --- |
| **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** |
| Создание нового списка | Пользователь выбирает опцию создания нового списка. Система запрашивает у пользователя тип записей (группа, староста, численность студентов). В случае выбора произвольного файла в качестве типа хранения, система запрашивает у пользователя путь к файлу. Система создает новый список с указанными характеристиками. |
| Заполнение списка | Пользователь выбирает опцию "Ручное заполнение" списка. Система запрашивает у пользователя данные для каждой записи (группа, староста, численность студентов). Система добавляет новую запись в список. Пользователь может повторять этот шаг для добавления нескольких записей. |
| Вывод содержимого списка | Пользователь выбирает опцию вывода содержимого списка. Система отображает содержимое списка в виде перечня записей, упорядоченных по заданному полю (если список был отсортирован). |
| Просмотр записи за записью | Пользователь выбирает опцию просмотра содержимого списка "запись за записью". Система последовательно отображает каждую запись списка. |
| Поиск первой записи по заданному номеру или шаблону | Пользователь выбирает опцию поиска первой записи списка. Пользователь указывает номер или шаблон для поиска. Система ищет первую запись, соответствующую заданным критериям, и отображает её. |
| Удаление записи(-ей) | Пользователь выбирает опцию удаления записи(-ей). Пользователь указывает, какую запись нужно удалить (с конца списка, с начала списка или указанные записи). Система удаляет указанную(-ые) запись(-и) из списка. |

**3.**

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

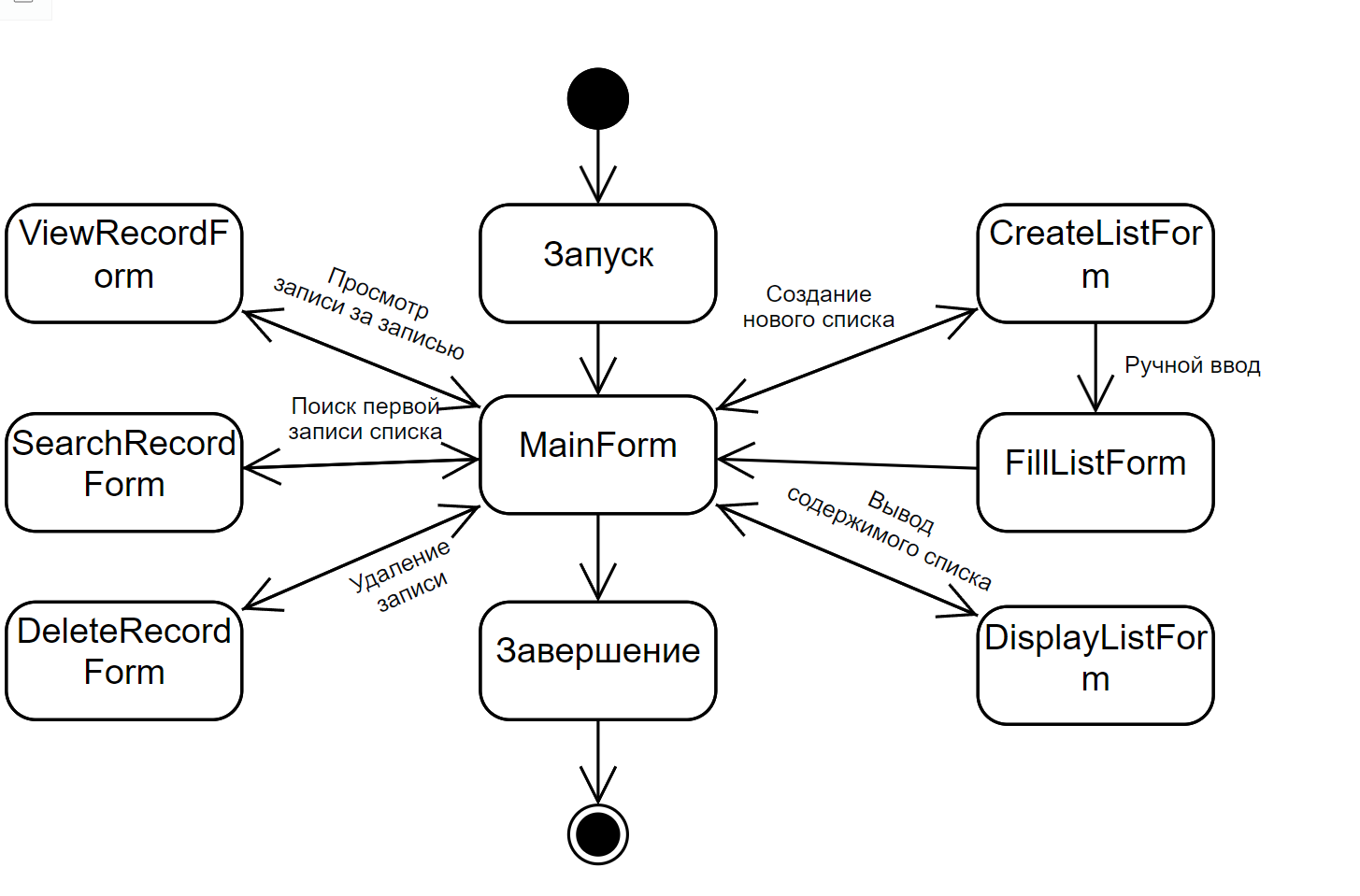
**Интерфейсные формы и сопутствующие классы:**

* + Форма "Главное меню" (MainForm): отображает список доступных задач и предоставляет кнопки для выбора операций с записями.
  + Форма "Создание списка" (CreateListForm): позволяет пользователю выбрать тип записей и тип хранения списка.
  + Форма "Заполнение списка" (FillListForm): позволяет пользователю вводить данные для каждой записи и добавлять их в список.
  + Форма "Вывод списка" (DisplayListForm): отображает содержимое списка в виде перечня записей, упорядоченных по выбранному полю.
  + Форма "Просмотр записей" (ViewRecordForm): позволяет пользователю последовательно просматривать каждую запись списка.
  + Форма "Поиск записи" (SearchRecordForm): позволяет пользователю указать номер или шаблон для поиска и отображает первую найденную запись.
  + Форма "Удаление записи" (DeleteRecordForm): позволяет пользователю выбрать метод удаления записи (с конца списка, с начала списка, указанную запись) и удалить ее.

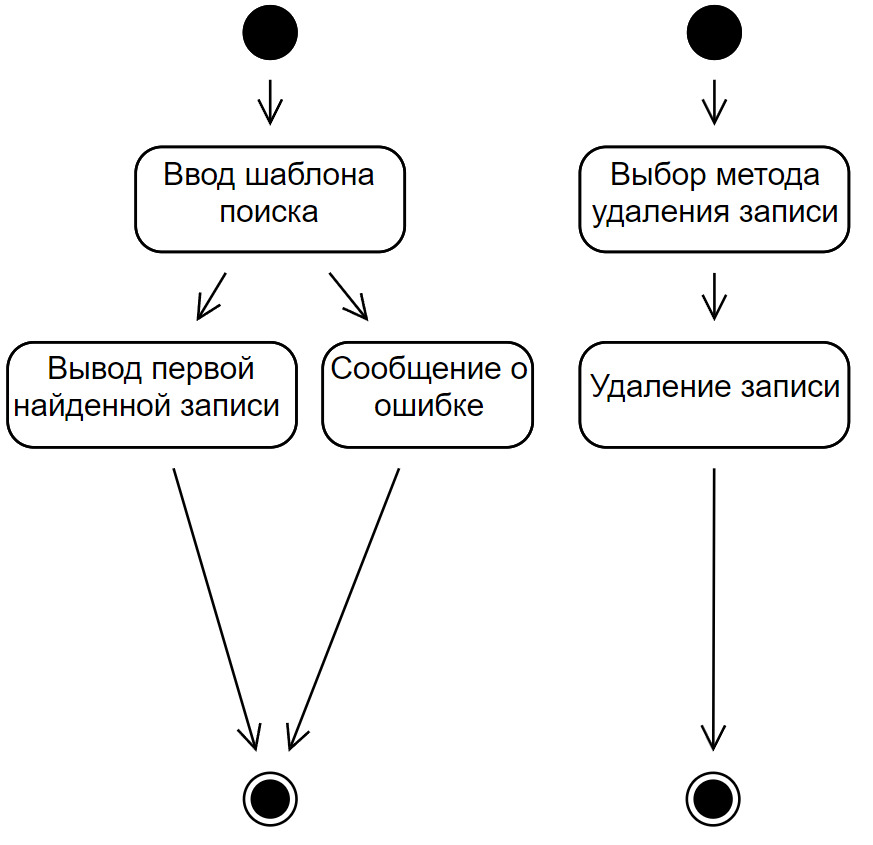
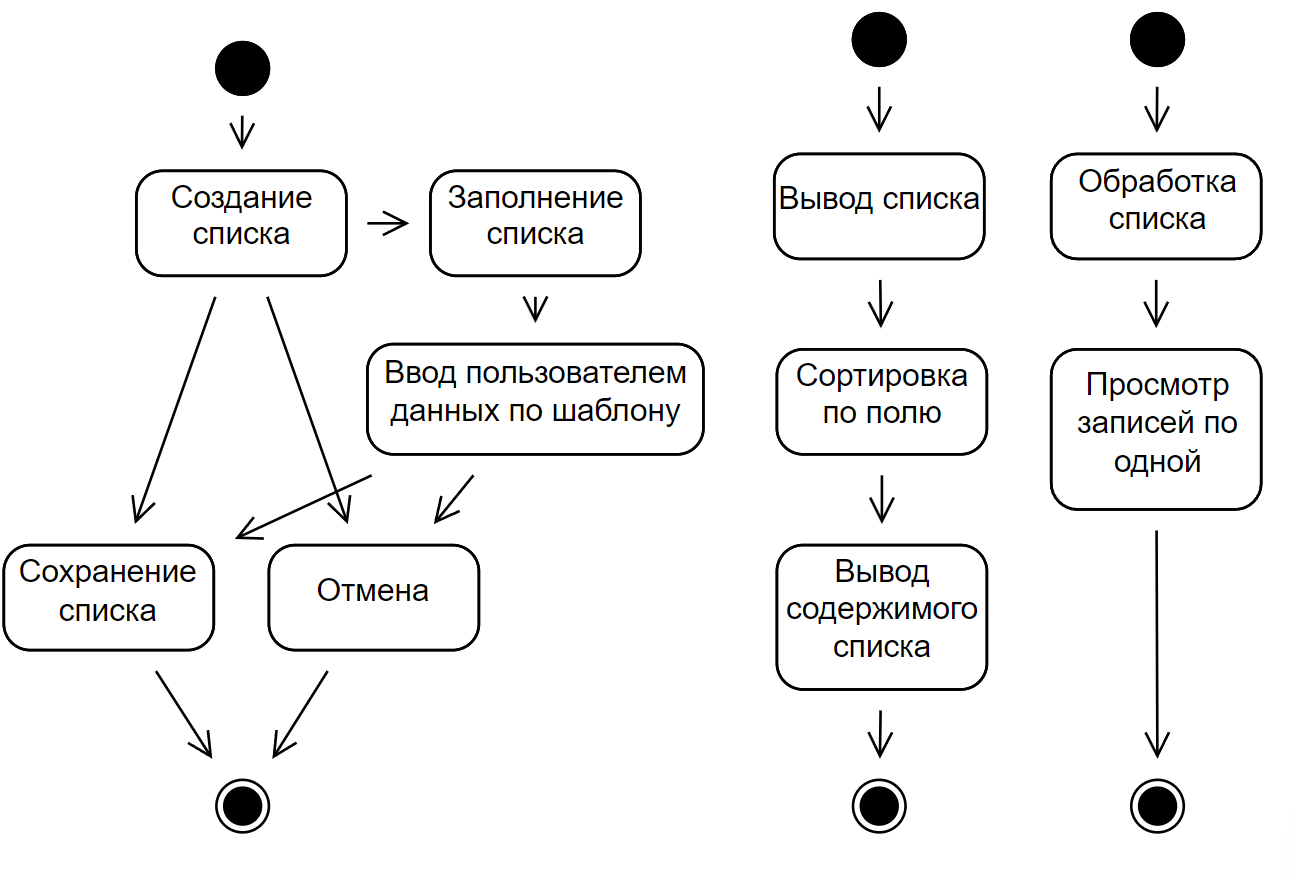
**Дизайн форм:**

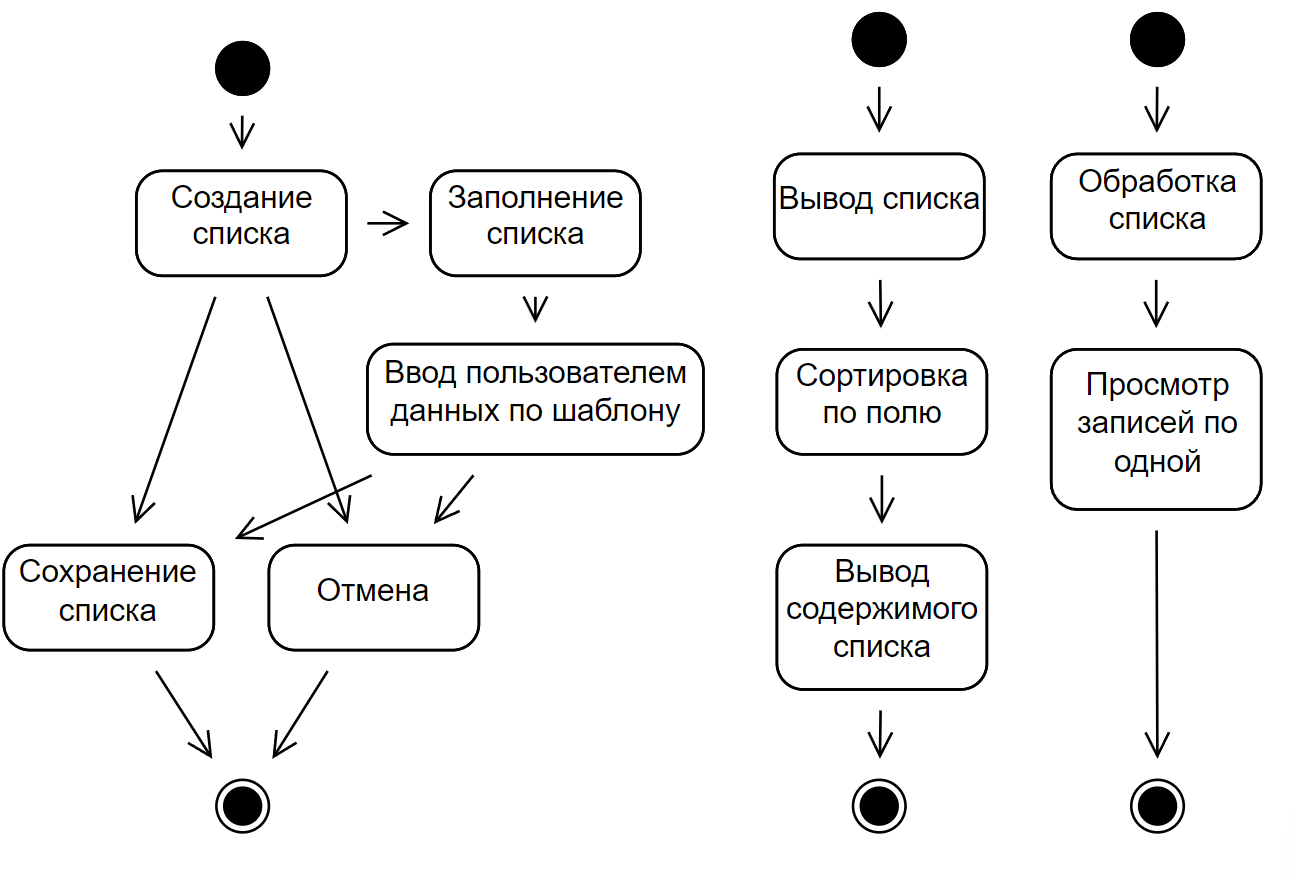
* + Форма MainForm: содержит меню с кнопками для выбора задач и операций со списком записей.
  + Форма CreateListForm: содержит элементы управления для выбора типа записей и типа хранения списка.
  + Форма FillListForm: содержит текстовые поля для ввода данных записей и кнопку для добавления их в список.
  + Форма DisplayListForm: отображает содержимое списка в виде ListBox или DataGridView, с возможностью выбора поля для сортировки.
  + Форма ViewRecordForm: содержит элементы управления для просмотра каждой записи списка.
  + Форма SearchRecordForm: содержит поле ввода номера или шаблона для поиска, кнопку для выполнения поиска и отображения найденной записи.
  + Форма DeleteRecordForm: содержит элементы управления для выбора метода удаления записи и кнопку удаления.

**Диаграмма состояний приложения:**



**Диаграммы состояний для прецедентов:**

**** 

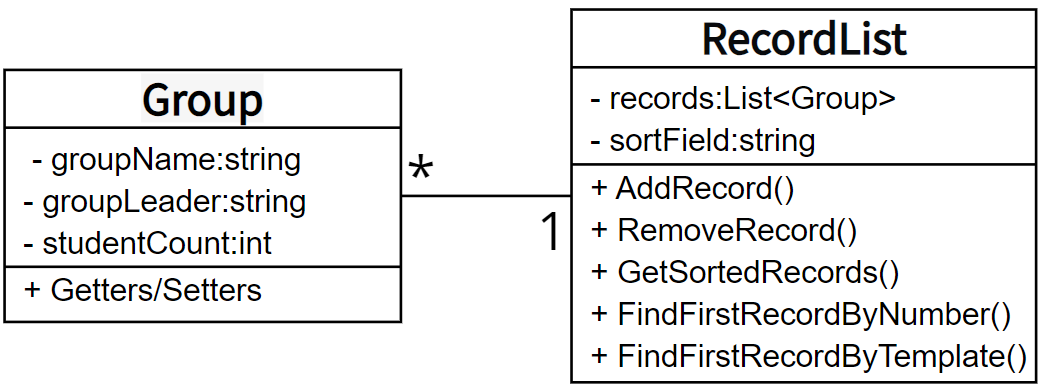


**4.**

**Выявление и первоначальное описание классов предметной области:**

1. Класс "Group" (Группа):
   * Поля:
     + string groupName: название группы
     + string groupLeader: имя и фамилия старосты
     + int studentCount: численность студентов в группе
     + Getters/Setters: геттеры и сеттеры для переменных
2. Класс "RecordList" (Список записей):
   * Поля:
     + List records: список записей типа "Группа"
     + string sortField: поле, по которому осуществляется сортировка списка
   * Методы:
     + void AddRecord(Group group): добавляет новую запись группы в список
     + void RemoveRecord(Group group): удаляет указанную запись группы из списка
3. Классы интерфейсных форм и сопутствующие классы:
   * MainForm: форма главного меню
   * CreateListForm: форма создания списка записей
   * FillListForm: форма заполнения списка записей
   * DisplayListForm: форма вывода списка записей
   * ViewRecordForm: форма просмотра записей
   * SearchRecordForm: форма поиска записей
   * DeleteRecordForm: форма удаления записей

**Диаграммы классов предметной области на уровне “первичных” отношений:**

****

**5.**

**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.

**6.**

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

1) Прецедент "Создание списка":

* Пользователь выбирает тип записей и тип хранения списка на форме "Создание списка" (CreateListForm).
* При выборе вида хранения файла в виде произвольного файла (в), система создает новый файл, с которым будет работать.

2) Прецедент "Заполнение списка":

* Пользователь вводит данные для каждой записи и добавляет их в список на форме "Заполнение списка" (FillListForm).
* Данные записи сохраняются в соответствующем хранилище (файл).
* После добавления записи, пользователь может продолжить вводить данные для следующей записи или завершить процесс заполнения списка.

3) Прецедент "Вывод списка":

* Пользователь выбирает поле, по которому нужно упорядочить записи, на форме "Вывод списка" (DisplayListForm).
* Система сортирует записи по указанному полю и отображает их в виде перечня на форме.
* Пользователь может выполнять дополнительные действия с отображенным списком, например, редактировать или удалять записи.

4) Прецедент "Просмотр записей":

* Пользователь просматривает записи списка последовательно на форме "Просмотр записей" (ViewRecordForm).
* Система отображает информацию о текущей записи, позволяя пользователю перемещаться между записями.

5) Прецедент "Поиск записи":

* Пользователь указывает номер или шаблон для поиска на форме "Поиск записи" (SearchRecordForm).
* Система выполняет поиск в списке по указанному номеру или шаблону и отображает первую найденную запись.

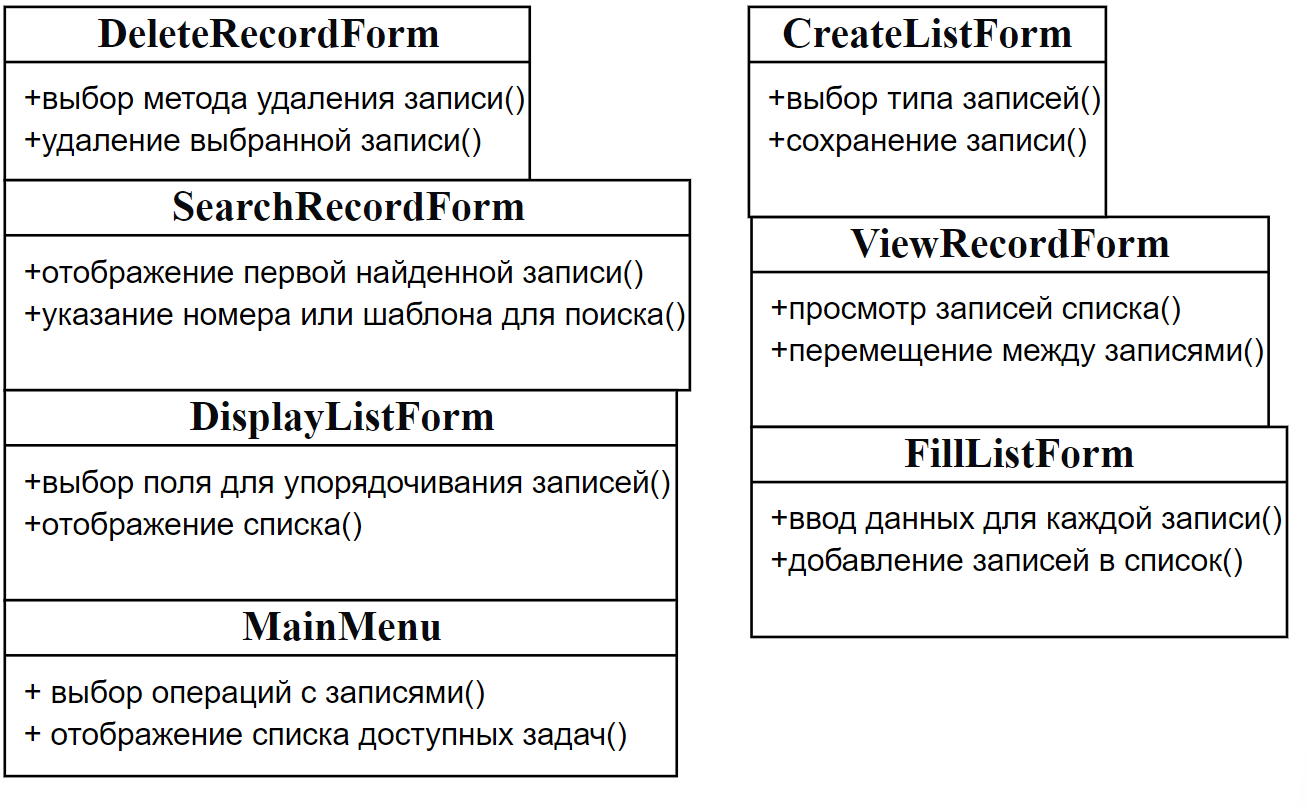
6) Прецедент "Удаление записи":

* Пользователь выбирает метод удаления записи (с конца списка, с начала списка или указанную запись) на форме "Удаление записи" (DeleteRecordForm).
* Система выполняет удаление выбранной записи из списка.
* Если выбран метод удаления с конца или с начала списка, то система будет продолжать удалять записи до тех пор, пока не будет выполнено условие остановки.

**7.**

Уточненное описание классов "Group" (Группа), "RecordList" (Список записей) присутствует в пункте 4.

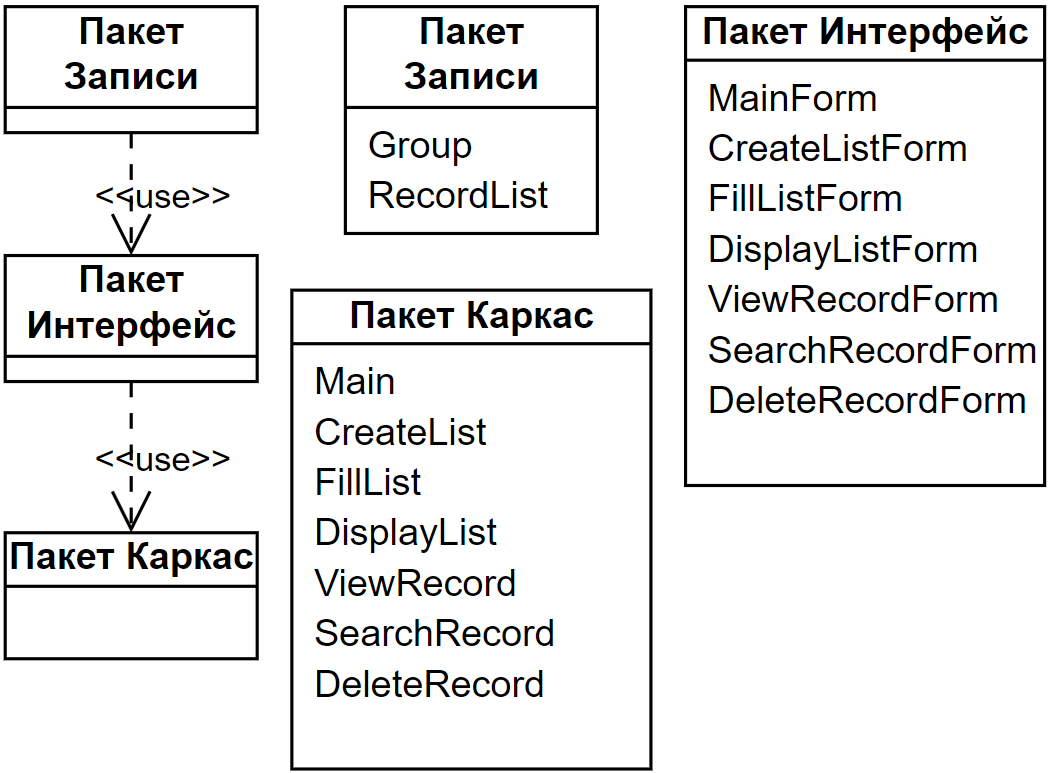
* Класс "Главное меню" (MainMenu):
  + Методы: отображение списка доступных задач, выбор операций с записями.
* Класс "Окно создания списка" (CreateListForm):
  + Методы: выбор типа записей, сохранение записи.
* Класс "Окно заполнения списка" (FillListForm):
  + Методы: ввод данных для каждой записи, добавление записей в список.
* Класс "Окно вывода списка" (DisplayListForm):
  + Методы: выбор поля для упорядочивания записей, отображение списка.
* Класс "Окно просмотра записей" (ViewRecordForm):
  + Методы: просмотр записей списка, перемещение между записями.
* Класс "Окно поиска записи" (SearchRecordForm):
  + Методы: указание номера или шаблона для поиска, отображение первой найденной записи.
* Класс "Окно удаления записи" (DeleteRecordForm):
  + Методы: выбор метода удаления записи, удаление выбранной записи.

****

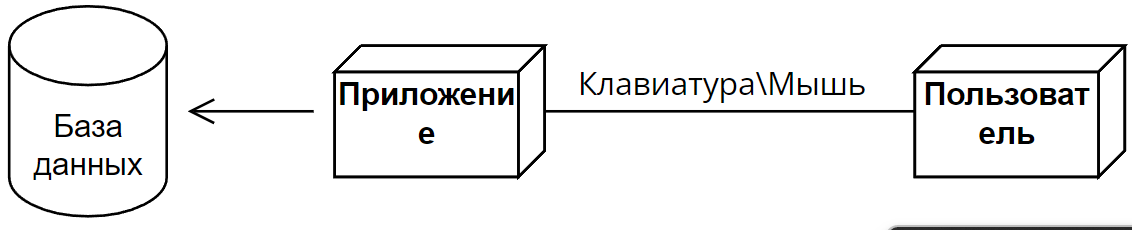
**8.**

1. Диаграмма активности для метода AddRecord(Group group):
   * Пользователь вызывает операцию "Добавить запись" в главном меню
   * Открывается окно создания списка (CreateListForm)
   * Пользователь выбирает тип записей и сохраняет список
   * Открывается окно заполнения списка (FillListForm)
   * Пользователь вводит данные для каждой записи и добавляет их в список
   * Метод AddRecord(Group group) вызывается в классе RecordList для добавления новой записи группы в список
2. Диаграмма активности для метода RemoveRecord(Group group):
   * Пользователь вызывает операцию "Удалить запись" в главном меню
   * Открывается окно удаления записи (DeleteRecordForm)
   * Пользователь выбирает метод удаления записи (с конца списка, с начала списка, указанной)
   * Пользователь удаляет выбранную запись
   * Метод RemoveRecord(Group group) вызывается в классе RecordList для удаления указанной записи группы из списка
3. Диаграмма активности для метода GetSortedRecords():
   * Пользователь вызывает операцию "Отобразить список" в главном меню
   * Открывается окно вывода списка (DisplayListForm)
   * Пользователь выбирает поле для упорядочивания записей
   * Метод GetSortedRecords() вызывается в классе RecordList для получения упорядоченного списка записей
   * Отображается список записей, упорядоченных по выбранному полю
4. Диаграмма активности для метода FindFirstRecordByNumber(int number):
   * Пользователь вызывает операцию "Поиск записи" в главном меню
   * Открывается окно поиска записи (SearchRecordForm)
   * Пользователь вводит номер записи, которую хочет найти
   * Метод FindFirstRecordByNumber(int number) вызывается в классе RecordList для поиска первой записи с заданным номером
   * Отображается найденная запись или сообщение, если запись не найдена
5. Диаграмма активности для метода FindFirstRecordByTemplate(string template):
   * Пользователь вызывает операцию "Поиск записи" в главном меню
   * Открывается окно поиска записи (SearchRecordForm)
   * Пользователь вводит шаблон, по которому хочет найти запись
   * Метод FindFirstRecordByTemplate(string template) вызывается в классе RecordList для поиска первой записи по заданному шаблону
   * Отображается найденная запись или сообщение, если запись не найдена
6. Диаграмма активности для метода ViewRecordForm():
   * Пользователь вызывает операцию "Просмотр записей" в главном меню
   * Открывается окно просмотра записей (ViewRecordForm)
   * Пользователь просматривает записи списка и может перемещаться между записями
7. Диаграмма активности для метода RemoveRecord(Group group):
   * Пользователь вызывает операцию "Удалить запись" в главном меню
   * Открывается окно удаления записи (DeleteRecordForm)
   * Пользователь выбирает метод удаления записи
   * Пользователь удаляет выбранную запись
   * Метод RemoveRecord(Group group) вызывается в классе RecordList для удаления выбранной записи
8. Диаграмма активности для метода отображения списка доступных задач в классе "Главное меню" (MainMenu):
   * Пользователь вызывает метод "отображение списка доступных задач"
   * Метод отображает список доступных задач на экране
9. Диаграмма активности для метода выбора операций с записями в классе "Главное меню" (MainMenu):
   * Пользователь вызывает метод "выбор операций с записями"
   * Метод открывает подменю операций с записями
   * Пользователь выбирает одну из операций (например, добавление, удаление, просмотр или поиск записей)
10. Диаграмма активности для метода выбора типа записей в классе "Окно создания списка" (CreateListForm):
    * Пользователь вызывает метод "выбор типа записей"
    * Метод открывает список доступных типов записей
    * Пользователь выбирает один из типов записей (например, группа, студент или событие)
11. Диаграмма активности для метода сохранения списка в классе "Окно создания списка" (CreateListForm):
    * Пользователь вызывает метод "сохранение списка"
    * Метод сохраняет текущий список записей
12. Диаграмма активности для метода ввода данных для каждой записи в классе "Окно заполнения списка" (FillListForm):
    * Пользователь вызывает метод "ввод данных для каждой записи"
    * Метод открывает форму для ввода данных
    * Пользователь вводит данные для каждой записи (например, название группы, имя и фамилию старосты, численность студентов)
13. Диаграмма активности для метода добавления записей в список в классе "Окно заполнения списка" (FillListForm):
    * Пользователь вызывает метод "добавление записей в список"
    * Метод добавляет введенные пользователем данные в список записей
14. Диаграмма активности для метода выбора поля для упорядочивания записей в классе "Окно вывода списка" (DisplayListForm):
    * Пользователь вызывает метод "выбор поля для упорядочивания записей"
    * Метод открывает список доступных полей для упорядочивания (например, название группы, имя старосты, численность студентов)
    * Пользователь выбирает одно из полей для упорядочивания списка записей
15. Диаграмма активности для метода отображения списка в классе "Окно вывода списка" (DisplayListForm):
    * Пользователь вызывает метод "отображение списка"
    * Метод отображает список записей на экране, упорядоченный по выбранному полю
16. Диаграмма активности для метода просмотра записей списка в классе "Окно просмотра записей" (ViewRecordForm):
    * Пользователь вызывает метод "просмотр записей списка"
    * Метод открывает окно просмотра записей
    * Пользователь может просматривать каждую запись и перемещаться между ними
17. Диаграмма активности для метода указания номера или шаблона для поиска в классе "Окно поиска записи" (SearchRecordForm):
    * Пользователь вызывает метод "указание номера или шаблона для поиска"
    * Метод открывает окно поиска записи
    * Пользователь вводит номер или шаблон для поиска записи
18. Диаграмма активности для метода отображения первой найденной записи в классе "Окно поиска записи" (SearchRecordForm):
    * Пользователь вызывает метод "отображение первой найденной записи"
    * Метод выполняет поиск по указанному номеру или шаблону записи
    * Если запись найдена, метод отображает ее на экране
    * Если запись не найдена, метод выводит сообщение о ненаходе
19. Диаграмма активности для метода выбора метода удаления записи в классе "Окно удаления записи" (DeleteRecordForm):
    * Пользователь вызывает метод "выбор метода удаления записи"
    * Метод открывает список доступных методов удаления (например, удаление с конца списка, удаление с начала списка, удаление выбранной записи)
    * Пользователь выбирает один из методов удаления записи
20. Диаграмма активности для метода удаления выбранной записи в классе "Окно удаления записи" (DeleteRecordForm):
    * Пользователь вызывает метод "удаление выбранной записи"
    * Метод удаляет выбранную пользователем запись из списка

**10.**

****

**11.**

****

**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.