Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 16

По дисциплине «Объектноориентироавнное технологии программирования и стандарты проектирования»

Выполнил:

Студент 1 курса 10 группы

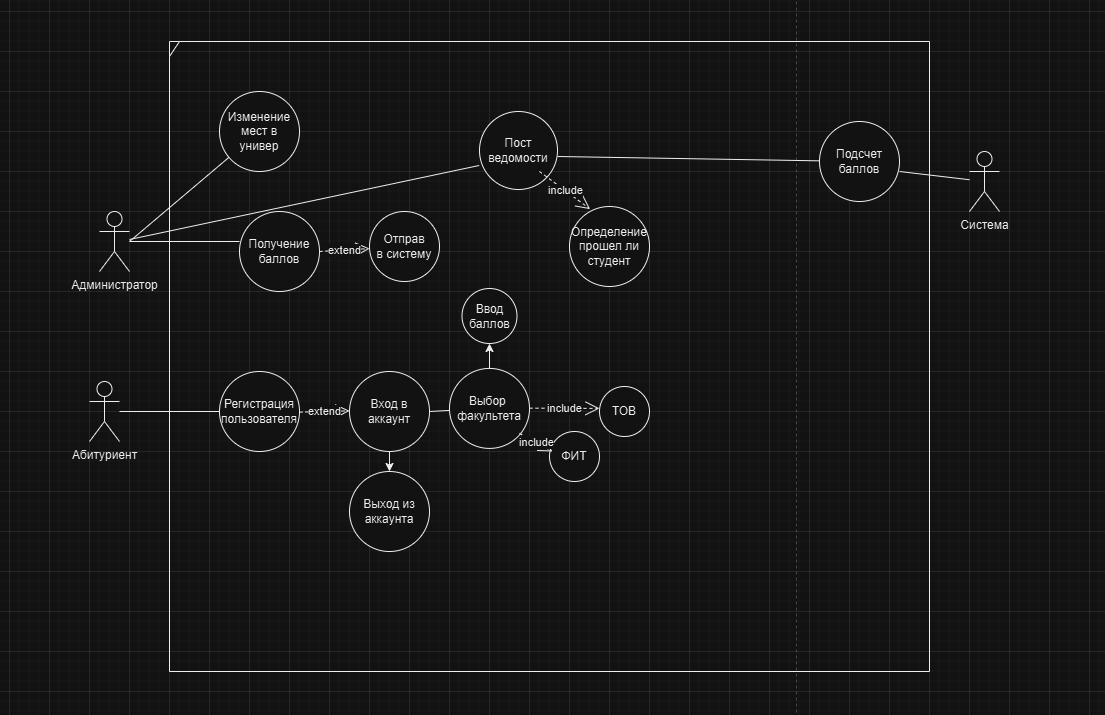
Гулецкий Прохор

Преподаватель: асс. Гончар Егор Андреевич

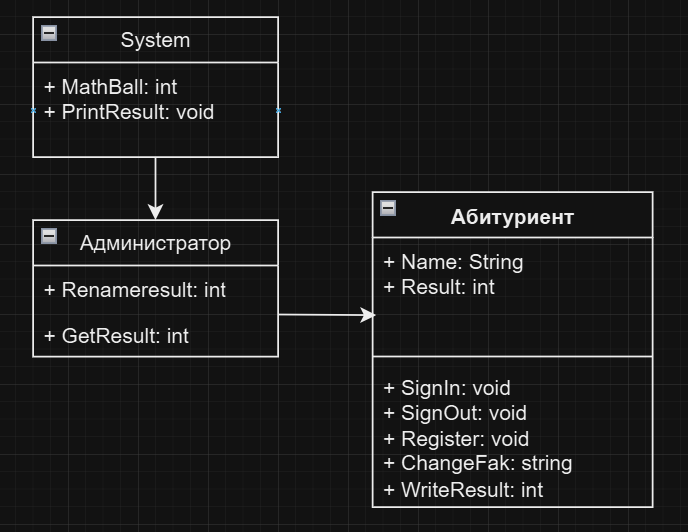
2024, Минск

**Вариант 3**

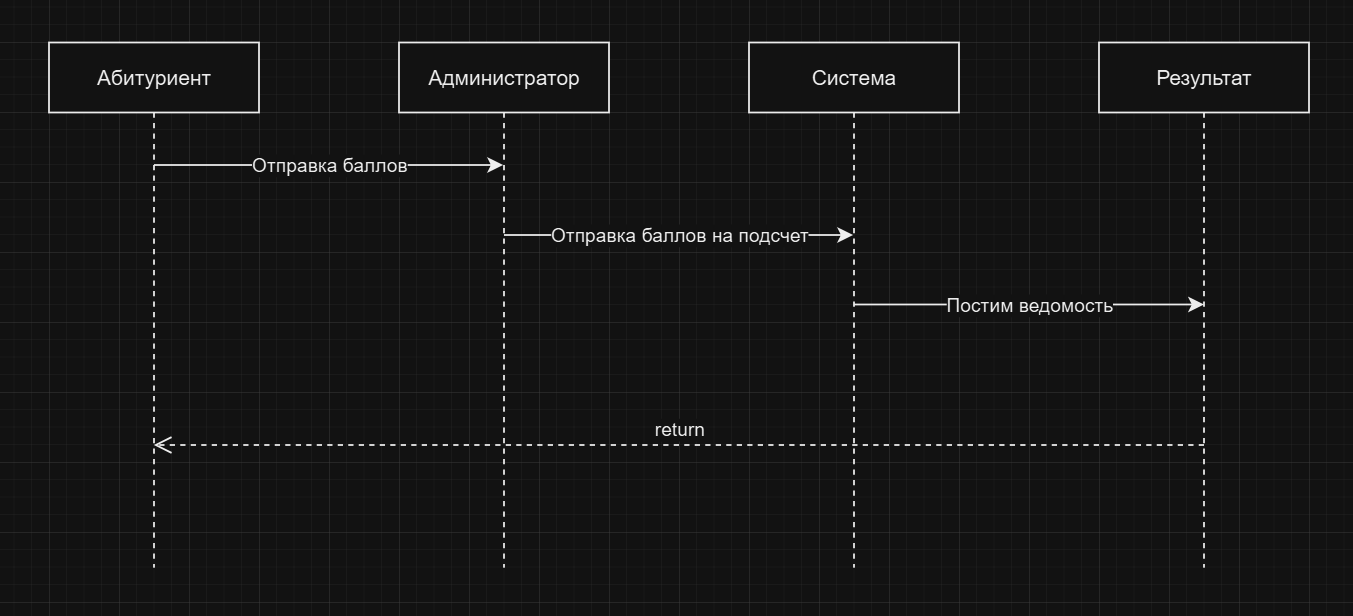
1. Диаграмма вариантов использования с включениями и расширениям



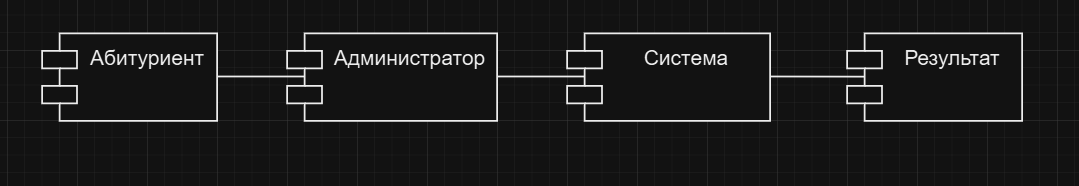
1. Диаграмма классов



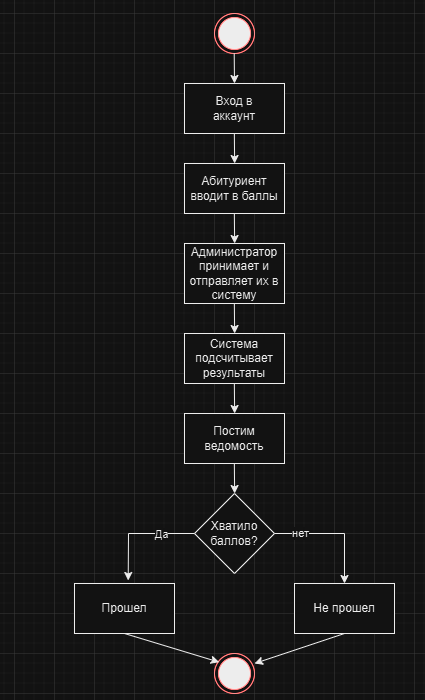
1. Диаграмма последовательности



1. Диаграмма компонентов или/и размещения



1. Диаграмма деятельности



**Ответы на вопросы:**

**1. Что такое UML?**

UML (Unified Modeling Language) — это стандартный язык визуализации, описания, проектирования и документирования программных систем. UML используется для создания диаграмм, которые позволяют отображать структуру и поведение системы. Он особенно полезен на этапе проектирования, так как помогает разработчикам, аналитикам и заказчикам лучше понимать архитектуру системы.

**2. Перечислите типы диаграмм UML**

UML включает несколько типов диаграмм, которые можно разделить на диаграммы структуры и диаграммы поведения:

*1. Диаграммы структуры:*

Диаграмма классов (Class Diagram)

Диаграмма объектов (Object Diagram)

Диаграмма компонентов (Component Diagram)

Диаграмма развертывания (Deployment Diagram)

Диаграмма пакетов (Package Diagram)

*2. Диаграммы поведения:*

Диаграмма прецедентов (Use Case Diagram)

Диаграмма активности (Activity Diagram)

Диаграмма последовательности (Sequence Diagram)

Диаграмма состояний (State Machine Diagram)

Диаграмма взаимодействия (Communication Diagram)

**3. Какие отношения между классами могут быть на диаграмме классов?**

На диаграмме классов можно обозначить несколько типов отношений:

**Ассоциация** — показывает связь между двумя классами.

**Агрегация** — слабая форма ассоциации, где один класс может существовать отдельно от другого.

**Композиция** — сильная форма агрегации, где один класс не может существовать без другого.

**Наследование** (Обобщение) — связь, при которой один класс наследует свойства и методы другого.

**Реализация** — связь, при которой класс реализует интерфейс.

**4. Как обозначаются абстрактные классы на диаграмме классов?**

Абстрактные классы в UML обозначаются с помощью курсивного написания их названия или добавлением слова "abstract". Также часто используется слово «<abstract>» в имени класса, что подчеркивает его абстрактность.

**5. Как обозначаются интерфейсы на диаграмме классов?**

Интерфейсы обозначаются как классы с курсивным названием, и перед ним добавляется <<interface>>. Интерфейсы связываются с реализующими классами стрелкой с пунктирной линией и пустым треугольником на конце.

**6. Как отображается доступность членов класса на диаграмме классов?**

Уровни доступа членов класса обозначаются следующими символами:

`+` — публичный (public)

`-` — закрытый (private)

`#` — защищенный (protected)

`~` — внутренний (package)

**7. Что такое агрегация? Как обозначается?**

**Агрегация** — это отношение "частьцелое" между объектами, где один объект (целое) включает другой (часть), но оба могут существовать независимо. Например, университет включает факультеты, но факультеты могут существовать вне контекста университета.

На диаграмме классов агрегация обозначается стрелкой с пустым ромбом у конца, направленного к целому объекту.

**8. Что такое ассоциация?**

**Ассоциация** — это связь между двумя классами, которая показывает, что объекты одного класса взаимодействуют с объектами другого. Ассоциация может быть однонаправленной или двунаправленной, и часто сопровождается указанием множества (количества объектов) с каждой стороны.

Ассоциация обозначается простой линией между двумя классами на диаграмме.

**9. Какие обозначения используют на диаграмме последовательности?**

На диаграмме последовательности используются следующие обозначения:

**Участники** (Actors) — горизонтальные объекты, представляющие роли или объекты.

**Линии жизни** (Lifelines) — вертикальные пунктирные линии, которые обозначают время жизни объекта.

**Сообщения** (Messages) — стрелки между линиями жизни, показывающие взаимодействие (вызов метода, возврат значения и т. д.).

**Активности** (Activations) — прямоугольные блоки на линии жизни, обозначающие выполнение действия.

**10. Для чего нужна диаграмма последовательности?**

Диаграмма последовательности помогает отобразить порядок взаимодействий между объектами в рамках определенного сценария. Она полезна для анализа и проектирования систем, так как показывает, какие сообщения отправляются между объектами и в каком порядке, что помогает понять бизнеспроцессы или поведение системы.

**11. Каково назначение диаграммы использования, пакетов и активности?**

Диаграмма прецедентов (диаграмма использования) показывает, как пользователи (актеры) взаимодействуют с системой через определенные функции (прецеденты). Она полезна для понимания требований к системе.

Диаграмма пакетов отображает, как классы и другие элементы логически сгруппированы в пакеты. Это помогает организовать крупные системы, упрощая структурирование и модульное проектирование.

Диаграмма активности моделирует рабочий процесс или бизнеспроцесс, отражая последовательность действий, состояния и переходы**.**