**Министерство образования и науки Российской Федерации**

**федеральное государственное бюджетное образовательное учреждение высшего профессионального образования**

**«Российский экономический университет имени Г.В. Плеханова»**

Институт цифровой экономики и информационных технологий

Кафедра информатики

**Практическая работа**

по дисциплине

«Объектно-ориентированные технологии в программной инженерии»

на тему «Лабораторная работа №4»

Выполнили

студенты 3 курса

группы 15.27Д-БИ19/22Б

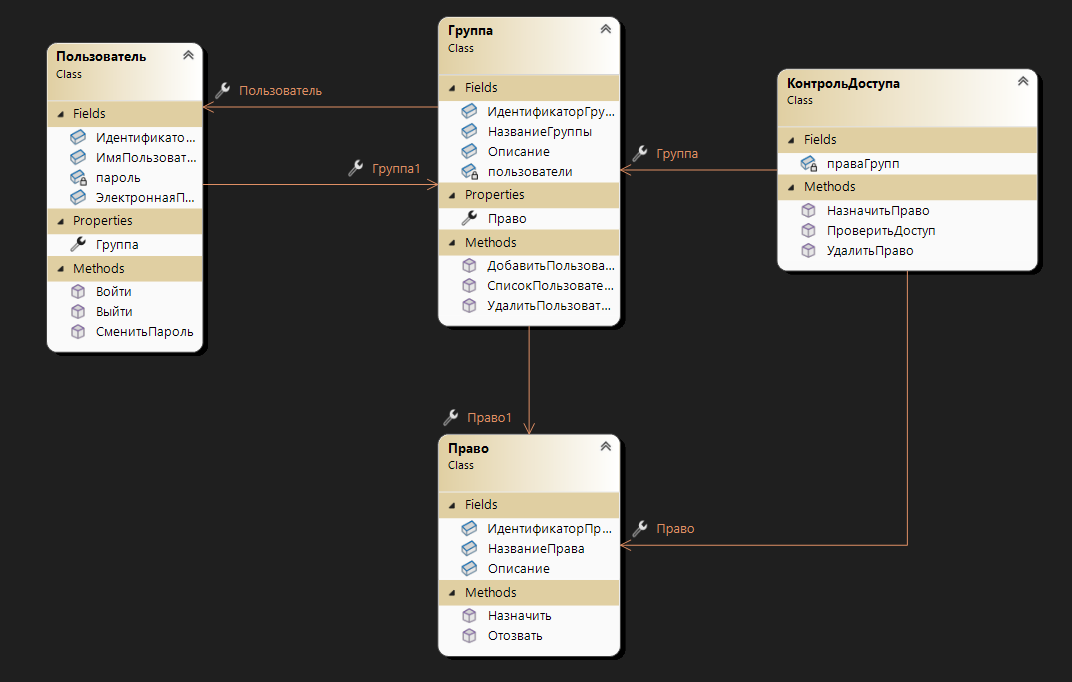
очной формы обучения

высшая школа ВШКМиС

Фамилии студентов Нгуен Као Бач,

Москва, 2025

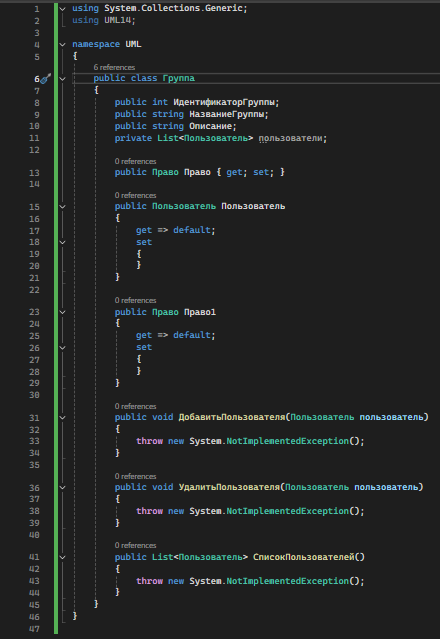
**Тема 14: Работа с группами пользователей, назначение прав доступа**



Система содержит четыре основных класса:

* Пользователь: представляет пользователя системы, содержит такие поля, как идентификатор, имя, электронная почта и пароль. Каждый пользователь связан с объектом класса Группа.
* Группа: содержит информацию о группе — идентификатор, название, описание и список пользователей. Она связана с классом Пользователь и с классом Право, что означает возможность назначения прав группе. Так как связь не содержит спецификации кратности, нельзя однозначно сказать, сколько прав может быть у одной группы или сколько пользователей она включает — это остаётся на усмотрение проектировщика.
* Право: описывает объект доступа, включающий идентификатор, название и описание. Этот класс не владеет другими объектами, но используется в Группа и КонтрольДоступа как сущность для управления правами.
* КонтрольДоступа: реализует функции назначения, проверки и удаления прав. Он связан с Группа и Право, так как оперирует ими при реализации логики доступа.

Все связи между классами отображены как **обычные ассоциации**, без указания кратности.

s

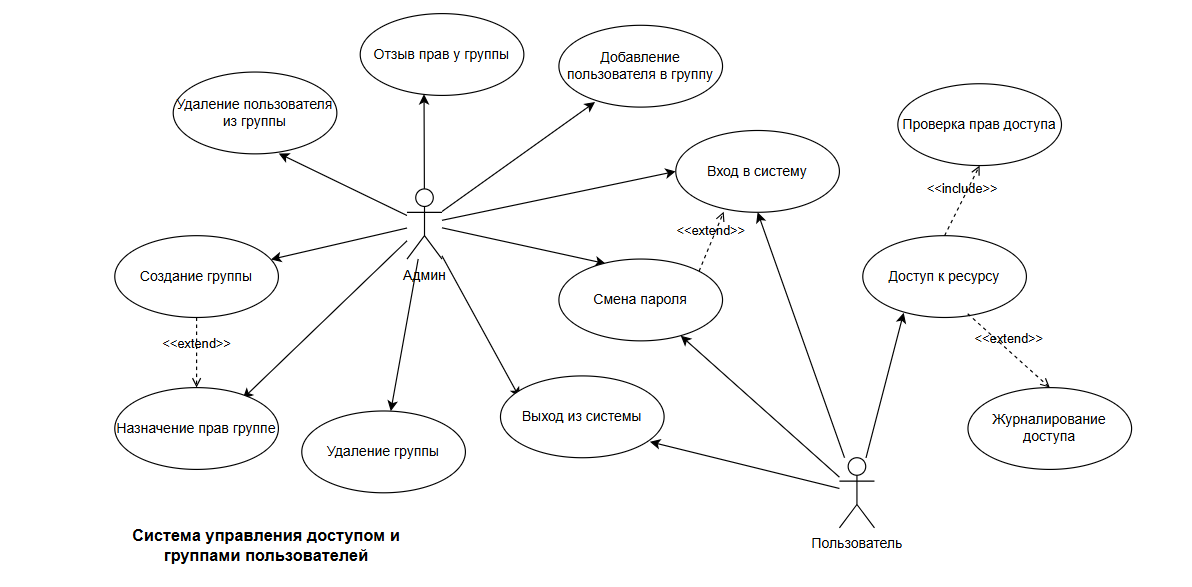
A screen shot of a computer program

AI-generated content may be incorrect. A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.s



Пользователь может выполнить стандартные действия: **вход в систему**, **выход из системы**, а также **доступ к ресурсу**. При этом при доступе к ресурсу обязательно выполняется **проверка прав доступа**, реализованная как связь <<include>>, а **журналирование доступа** происходит по расширению <<extend>>, то есть по мере необходимости. После успешного входа пользователь может **сменить пароль**, что также реализовано как <<extend>>.

Администратор обладает расширенными правами. Он может **создавать** и **удалять группы**, **добавлять** и **удалять пользователей из групп**, **назначать права группе**, а также **отзывать ранее назначенные права**. Важно отметить, что после создания группы возможна немедленная настройка прав, что отражено с помощью связи <<extend>> от варианта использования «Создание группы» к «Назначение прав группе». Все действия администратора направлены на управление структурой пользователей и настройку уровня их доступа к системным ресурсам.

A diagram of a company

AI-generated content may be incorrect.

На представленной диаграмме объектов показано конкретное состояние системы управления доступом на заданный момент времени. Диаграмма включает двух пользователей — **Иван** и **Ольга**, представленных объектами :Пользователь1 и :Пользователь2. Оба пользователя принадлежат одной группе — :Группа1, с названием **Администраторы**.

Группе :Группа1 назначено право :Право1, имеющее название **Удаление данных**. Связь между группой и правом реализуется через объект :КонтрольДоступа1, который отображает логическое управление назначением прав.

Связи между объектами отражают следующие отношения:

* :Пользователь1 и :Пользователь2 → :Группа1: принадлежность пользователей к группе.
* :Группа1 → :Право1: наличие у группы определённого права.
* :КонтрольДоступа1 → :Группа1, :Право1: механизм логического контроля доступа.

**Контрольные вопросы**

1. **Назначение диаграммы классов?**

Диаграмма классов предназначена для отображения **статической структуры системы** в терминах классов, их атрибутов, операций и связей между ними. Она помогает разработчику понять логическую модель системы и служит основой для реализации кода.

1. **Как обозначается класс и что в себя включает?**

Класс изображается в виде **прямоугольника**, разделённого на секции:

* имя класса (вверху)
* атрибуты (в середине)
* операции (внизу)

1. **Что такое атрибут в диаграмме классов?**

Атрибут — это **свойство класса**, отражающее состояние объекта. Атрибут указывается в формате:  
модификатор\_доступа имя : тип  
Например: +имя : String

1. **Как обозначаются области видимости у атрибута?**

Обозначения следующие:

* + — public (общедоступный)
* # — protected (защищённый)
* - — private (закрытый)

1. **Что такое операция в диаграмме классов?**

Операция — это **метод класса**, который может выполняться объектом. Указывается в нижней секции прямоугольника и включает имя метода, параметры и возвращаемый тип.  
Пример: +сохранить(данные: String): bool

1. **Перечислить виды отношений между классами, охарактеризовать каждое из них?**

* **Ассоциация**: связь между объектами (например, "Пользователь принадлежит Группе").
* **Агрегация**: отношение "часть-целое", где части могут существовать отдельно.
* **Композиция**: более жёсткая агрегация — при удалении целого удаляются и части.
* **Обобщение**: наследование (дочерний класс наследует свойства родительского).
* **Зависимость**: изменение одного элемента может потребовать изменения другого.

1. **Как обозначается объект (экземпляр) класса?**

Объект обозначается так же, как класс, но с **именем экземпляра**, следующим за двоеточием, и **всё подчеркивается**.

Пример: :Пользователь1 — экземпляр класса Пользователь

Внутри блока указываются конкретные значения атрибутов, например:  
Имя = "Иван"