Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа № 5. Объектно-ориентированное моделирование. Структурные диаграммы UML

Студент: Пузиков А. А.

ФИТ 4 курс 6 группа

Преподаватель: Якубенко К. Д.

Минск 2024

**Введение**

Наименование: «Программное средство социальная сеть с возможностью организации проведения интеллектуальных игр с элементами социальной сети» (PSSIG).

Краткая характеристика: Информационная система разрабатывается для автоматизации и оптимизации процессов организации и проведения интеллектуальных игр, а также для повышения взаимодействия между пользователями. Область применения системы охватывает онлайн и оффлайн игры, включая командные и индивидуальные форматы, а также предоставляет функции для общения и обмена опытом среди участников.

Объект применения: Информационная система будет внедрена в деятельность организаторов и участников интеллектуальных игр для повышения оперативной эффективности, качества взаимодействия между пользователями и улучшения общего опыта участия в играх.

1. Основания для разработки

Документ, на основании которого ведется разработка: "Приказ о начале разработки программного средства для организации интеллектуальных игр" №222-ОК.

Организация, утвердившая этот документ: ООО "Интеллектуальные Игры".

Дата утверждения: 10.10.2024.

Наименование темы разработки: "Разработка и внедрение программного средства сбора статистики и организации проведения интеллектуальных игр с элементами социальной сети" (PSSIG).

2. Назначение разработки

Функциональное назначение программы "Программное средство сбора статистики и организации проведения интеллектуальных игр" (PSSIG) заключается в автоматизации и оптимизации процессов управления, мониторинга и взаимодействия внутри интеллектуальных игр. PSSIG предоставляет инструменты для:

* Регистрации и аутентификации пользователей (участников и организаторов).
* Создания и настройки игр, включая определение правил и условий участия.
* Автоматического сбора и анализа статистики по результатам игр.
* Обеспечения взаимодействия между участниками, организаторами и администрацией платформы.

1. **Постановка задачи**

Парадигма объектно-ориентированного программирования (ООП) является основой для создания современного программного обеспечения и находит широкое применение в разработке информационных систем. Модель объектов, основанная на данной парадигме, позволяет эффективно описывать свойства и функциональные возможности сущностей, взаимодействующих в рамках разрабатываемой системы. Эти объекты не существуют изолированно; они взаимодействуют друг с другом для достижения общей цели — успешного проведения интеллектуальных игр и обеспечения качественного взаимодействия пользователей.

Основные задачи, которые решает система PSSIG, включают:

* Регистрация и авторизация пользователей (участников, организаторов и администраторов).
* Создание и настройка игр, включая выбор тематики, правил и условий участия.
* Автоматизированный сбор и анализ статистики по результатам игр и активности пользователей.
* Взаимодействие между участниками, включая возможность добавления в друзья и общения в рамках платформы.
* Формирование отчетов по играм, включая количество участников, результаты и оценку взаимодействия.
* Система обратной связи и возможность оценки организаторов и игр участниками.
* Интеграция с внешними сервисами для упрощения оплаты и управления данными.
* Обеспечение поддержки пользователей через систему помощи и FAQ.

Стандартная библиотека языков программирования предоставляет разработчикам набор базовых инструментов, но специфика задач, связанных с интеллектуальными играми, требует создания уникальных классов и функций. Создание собственных классов и объектов в рамках PSSIG добавляет новый уровень абстракции, который позволяет четко определить функционал системы и обеспечивать эффективную работу с данными в дальнейшем.

1. **Описание программных средств**

Для разработки системы PSSIG и обеспечения ее функциональности используются различные программные средства, которые помогают в создании и управлении функционалом приложения.

* Название: Visual Studio.
* Версия: 2024.
* Режим использования: Офлайн. Visual Studio — это интегрированная среда разработки (IDE), которая предоставляет полный набор инструментов для создания, отладки и развертывания приложений.
* Доступность на платформах: Windows, macOS. Существует версия Visual Studio Code для кроссплатформенной разработки.
* С какими моделями работает: Поддержка различных языков программирования, включая C#, JavaScript и TypeScript, что позволяет эффективно разрабатывать как серверную, так и клиентскую части приложения.

Основные особенности:

* Поддержка разработки на нескольких языках программирования с использованием различных фреймворков.
* Встроенные инструменты для отладки и тестирования кода.
* Возможности для работы с системами контроля версий, такими как Git.
* Интуитивно понятный интерфейс с возможностью настройки под нужды разработчика.
* Обширная библиотека расширений и плагинов для повышения функциональности.

Кроме того, для визуализации процессов и взаимодействия в системе PSSIG используются инструменты для создания схем и диаграмм. Одним из таких популярных средств является Lucidchart.

* Название: Lucidchart.
* Версия: Онлайн-сервис.
* Режим использования: Онлайн. Lucidchart предоставляет веб-интерфейс для создания диаграмм и доступ к проектам через браузер.
* Доступность на платформах: Web, доступен на всех основных операционных системах.

Эти программные средства обеспечивают разработчикам необходимые инструменты для создания качественного и эффективного программного обеспечения, отвечающего требованиям пользователей PSSIG.

1. **Описание практического задания**

В диаграмме классов системы PSSIG представлены ключевые объекты, каждый из которых обладает уникальными атрибутами и методами, необходимыми для функционирования платформы.

На рисунке 3.1 можно увидеть диаграмму классов.

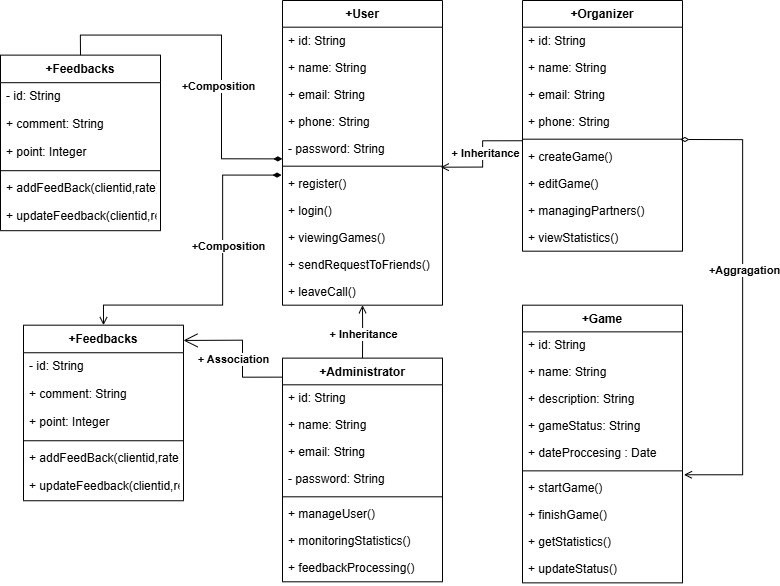


Рисунок 3.1 – Диаграмма классов

Пользователь (участник) имеет следующие атрибуты: Имя, ID, Email, Телефон и Профиль. Его методы включают Регистрация(), Вход(), ПросмотрИгр(), ОтправитьЗапросДрузьям() и ОставитьОтзыв(). Эти методы позволяют пользователю регистрироваться в системе, входить в свой профиль, просматривать доступные игры, добавлять других пользователей в друзья и оставлять отзывы о проведенных играх.

Организатор имеет атрибуты Имя, ID, Email и Телефон. Его методы включают СоздатьИгру(), НастроитьИгру(), УправлятьУчастниками() и ПросмотретьСтатистику(). Эти функции дают организатору возможность создавать и настраивать игры, управлять участниками и анализировать статистику игр.

Администратор системы также имеет свои атрибуты и методы. Его атрибуты включают Имя, ID, Логин и Пароль. Методы администратора включают УправлениеПользователями(), МониторингСтатистики() и ОбработкаОбратнойСвязи(). Эти функции помогают администратору управлять данными пользователей, настраивать тарифы на участие в играх и обрабатывать отзывы от пользователей.

Объект Игра имеет атрибуты ID, Название, Описание, Статус и ДатаПроведения. Методы, доступные для работы с игрой, включают НачатьИгру(), ЗавершитьИгру(), ПолучитьСтатистику() и ОбновитьСтатус(). Эти методы позволяют управлять процессом игры, отслеживать ее статус и собирать статистику по участникам.

Для упрощения сложных диаграмм классов используется диаграмма пакетов. В этой диаграмме классы группируются в логически связанные наборы, называемые пакетами. Пакет представляет собой объединение связанных элементов UML, что помогает лучше структурировать и организовать модель.

На рисунке 3.2 представлена диаграмма пакетов для PSSIG.

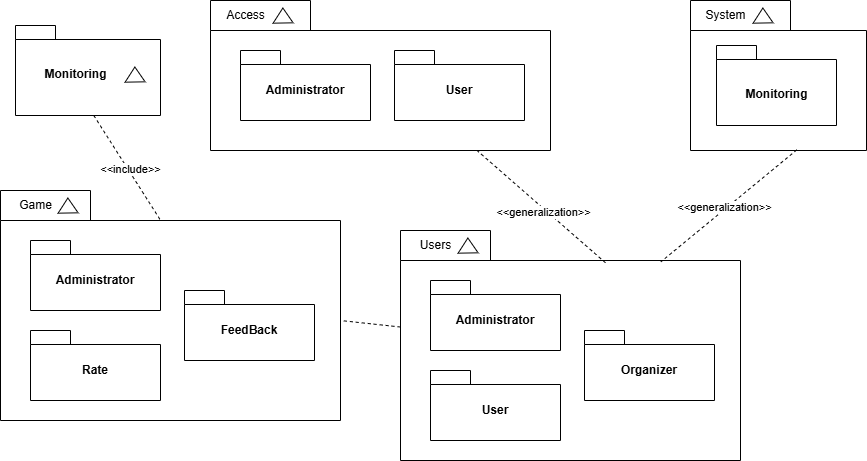


Рисунок 3.2 – Диаграмма пакетов

Эта диаграмма следует иерархической структуре вложенных пакетов, где атомарный модуль для вложенного пакета обычно представляет собой диаграмму классов.

Таким образом, использование диаграмм классов и пакетов позволяет структурировать функциональность системы PSSIG и облегчает разработку и поддержку проекта.

**4. Ответы на вопросы**

1.Перечислите структурные диаграммы, которые входят в UML 2.0.

В UML 2.0 к структурным диаграммам относятся такие диаграммы, как диаграмма классов, диаграмма объектов, диаграмма компонентов, диаграмма развёртывания, диаграмма пакетов, а также диаграмма композитных структур. Эти диаграммы предназначены для моделирования статической структуры системы, отображая её компоненты, их взаимосвязи и иерархические отношения.

2.Укажите назначение структурных диаграммы.

Основное назначение структурных диаграмм — описывать статические аспекты системы, такие как её классы, объекты, компоненты, а также связи между ними. Эти диаграммы позволяют понять, как структурированы элементы системы и как они взаимодействуют между собой на уровне компонентов и классов.

3.Опишите нотации, которые используются для построения Classes диаграмм.

Для построения диаграмм классов используются различные нотации. Классы обозначаются прямоугольниками, содержащими имя класса, атрибуты и методы. Связи между классами показываются линиями, обозначающими ассоциации, агрегации, композиции и зависимости. Могут также использоваться элементы для указания наследования (обобщения) и реализации интерфейсов.

4.Для чего применяются расширения диаграмм UML?

Расширения диаграмм UML применяются для уточнения и адаптации стандартных нотаций UML под специфические потребности проекта. Стереотипы и тегированные значения являются основными механизмами расширения, позволяя добавлять к стандартным элементам диаграммы дополнительные семантические данные.

5.Что означают понятия «стереотип» и «тегированное значение» в контексте расширенных диаграмм?

Понятие «стереотип» в контексте расширенных диаграмм UML обозначает специальные метки, которые расширяют стандартные элементы UML, придавая им дополнительный смысл. Например, стереотип может указывать, что определённый класс является интерфейсом или компонентом. «Тегированное значение» (tagged value) — это дополнительные свойства, прикрепляемые к элементам UML для хранения данных, которые не предусмотрены стандартными атрибутами. Стереотипы и тегированные значения помогают детализировать диаграммы и адаптировать их к конкретным условиям моделирования.

**Заключение**

В системе PSSIG представлено множество типов объектов, каждый из которых обладает уникальными характеристиками и функциями. Эти объекты имеют различные атрибуты, такие как имя, идентификатор, контактные данные и другие параметры, которые определяют их свойства и особенности. Каждый объект также содержит методы, позволяющие выполнять определенные действия или операции. Например, методы могут включать регистрацию пользователей, создание игр, участие в играх, отслеживание статистики и предоставление возможности оставлять отзывы о проведенных играх.

Связи между объектами играют ключевую роль в организации взаимодействий в системе. Эти связи могут быть разнообразными и включать различные типы отношений, такие как ассоциации и зависимости. Один объект может иметь ассоциацию с несколькими другими объектами, что позволяет создавать сложные взаимосвязи между элементами системы. Такая структура обеспечивает гибкость и масштабируемость системы, позволяя пользователям и администраторам управлять и взаимодействовать с множеством объектов. В то же время, каждый из связанных объектов может принадлежать только одному первичному объекту, что помогает сохранить упорядоченность и логику взаимодействий.

Для лучшего понимания и управления сложными структурами данных в PSSIG активно используются диаграммы пакетов. Эти диаграммы служат для группировки связанных элементов в логически обоснованные наборы, называемые пакетами. Пакет представляет собой объединение взаимосвязанных элементов, что позволяет более эффективно организовать и структурировать информацию. Использование диаграмм пакетов упрощает восприятие сложных диаграмм классов, делая их более доступными и понятными. Такой подход способствует лучшему структурированию модели системы и упрощает её управление и анализ, что, в свою очередь, позволяет команде разработчиков и пользователям более эффективно взаимодействовать с функционалом PSSIG.