

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Емельянов Сергей Геннадьевич

Должность: ректор

Дата подписания: 18.02.2023 15:05:20

Уникальный программный ключ:

9ba7d3e34c012eba476ffd2d064cf2781953be73002174d16836e336A768

МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)**

Кафедра программной инженерии

УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Доктионова

« 14 » 02



**ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ (ПРОЕКТА) ПО
ДИСЦИПЛИНЕ «МЕТОДОЛОГИЯ ПРОГРАММНОЙ
ИНЖЕНЕРИИ»**

Методические указания по выполнению курсовой работы (проекта)
для направления подготовки 09.04.04 «Программная инженерия»

Курск 2018

УДК 004.65

Составители: Т.М. Белова, В.Г. Белов

Рецензент

Кандидат технических наук, доцент кафедры программной инженерии ЮЗГУ И.Н. Ефремова

Выполнение курсовой работы (проекта) по дисциплине «Методология программной инженерии»: методические указания по выполнению курсовой работы (проекта) для направления подготовки 09.04.04 «Программная инженерия»/ Юго-Зап. гос. ун-т; сост.: Т.М. Белова, В.Г. Белов, Курск, 2018. – 34 с.: ил. 16, прилож. 1.

Изложен порядок выполнения, содержание и варианты заданий для курсовой работы (проекта) по дисциплине «Методология программной инженерии».

Материал предназначен для направления подготовки магистров 09.04.04 «Программная инженерия», а также будет полезен студентам всех направлений подготовки, изучающим технологии разработки программных систем.

Текст печатается в авторской редакции

Подписано в печать 14.02.18 Формат 60 x 84 1/16.
Усл. печ. л. 1,7 Уч.- изд. л. 1,6 Тираж 50 экз. Заказ 492
Бесплатно.

Юго-Западный государственный университет.
305040, Курск, ул. 50 лет Октября, 94.

Содержание

Основные положения	4
1 Цели и задачи.....	4
2 Требования к структуре и объему курсовых работ (проектов).....	5
3 Основные требования к содержанию КР (КП)	6
4 Типовые варианты заданий	11
4.1 Система электронного голосования	11
4.2 Обработка заказов на доставку пиццы	12
4.3 Ипподромный тотализатор.....	13
4.4 Система распределения вакансий на рынке труда	15
4.5 Система составления расписания.....	16
4.6 Оболочка для тестирования.....	18
4.7 Система для выбора места проведения саммита	19
4.8 Игра «морской бой»	20
4.9 Система дистанционного обучения	22
4.10 Система контроля за пробками	23
4.11 Конкурс проектов	25
4.12 Бизнес-аукцион	26
4.13 Система для координации участников проекта	27
4.14 Система оптимального распределения производственных мощностей	29
4.15. Концерт по заявкам.....	30
4.16. Игра «шашки»	31
4.17 Комментарии к диаграммам классов.....	32
Приложение.....	34

Основные положения

Курсовая работа – учебная (учебно-исследовательская) работа студента, являющаяся результатом самостоятельного углубленного изучения какого-либо вопроса дисциплины, выполненная по определенным требованиям, предъявляемым к ее структуре, содержанию и оформлению, демонстрирующая способность студента применить для решения учебной (учебно-исследовательской)

Курсовой проект – учебная работа студента, являющаяся результатом самостоятельного решения практической задачи, выполненная по определенным требованиям, предъявляемым к ее структуре, содержанию и оформлению, включающая расчетно-графическую (чертежи, макеты, схемы, таблицы и т.п.) и (или) материальную часть (модели, стенды, фотографии, рисунки, изделия и т.п.), демонстрирующая способность студента применить для решения практической задачи знания, умения, навыки и компетенции, приобретенные в процессе изучения дисциплины.

1 Цели и задачи

Целью КР (КП) как формы промежуточной аттестации является углубление и закрепление знаний, умений, навыков и компетенций, приобретенных обучающимися при изучении конкретной дисциплины, путем применения их к решению поставленной задачи по данной дисциплине, а также оценка уровня их сформированности.

Основными задачами при выполнении КР (КП) являются:

закрепление, углубление и систематизация полученных знаний и выработка умения самостоятельно применять их к решению конкретных задач;

приобретение и подтверждение наличия навыков исследовательской, расчетной и конструкторской работы;

закрепление навыков работы с компьютерной и офисной техникой, использования современных информационных технологий;

формирование компетенций, установленных ФГОС ВО и ОП ВО, закрепленных за дисциплиной, и подтверждение владения ими;

воспитание чувства ответственности за принимаемое решение;

развитие навыков работы с учебной, научной и справочной литературой, нормативно-правовой документацией, периодической печатью, стандартами, типовыми проектами и т.п.;

овладение навыками грамотного, ясного и сжатого изложения результатов работы и аргументированной защиты принятых решений и сделанных выводов;

формирование научного мировоззрения;
приобретение навыков регулярной и ритмичной работы, развитие самостоятельности и инициативы, воспитание сознательного и творческого отношения к труду.

2 Требования к структуре и объему курсовых работ (проектов)

Структура КР (КП) и ВКР включает:

- титульный лист;
- задание;
- реферат;
- содержание;
- обозначения и сокращения (при необходимости);
- введение;
- основная часть, в том числе исследовательский раздел;
- заключение;
- список использованных источников;
- приложения (при необходимости).

После приложений могут быть помещены самостоятельные конструкторские, технологические, программные и другие проектные материалы, выполненные в ходе проектирования согласно заданию.

В структуру основной части работы может входить экспериментальная часть с приложением электронного носителя с результатами работы.

Титульный лист является первой страницей КР (КП) и затем служит источником информации для обработки и поиска документа.

Форма титульного листа для ОП ВО приведена в приложении.

КР (КП) выполняются на основе задания, составленного руководителем КР (КП), содержащего исходные данные, необходимые для решения поставленных в работе задач.

Форма задания на КР (КП) приведена в приложении.

На КР (КП) руководитель пишет отзыв, структура которого приведена в приложении.

3 Основные требования к содержанию КР (КП)

Во введении обосновывается актуальность выбранной темы, формулируются цель, задачи, объект и предмет исследования, указываются избранные методы исследования, анализируется степень разработанности исследуемой проблемы в научной литературе.

Основная часть состоит из 3 разделов и имеет следующее содержание:

- 1 Техническое задание
 - 1.1 Основание для разработки
 - 1.2 Назначение разработки
 - 1.3 Исследование предметной области
 - 1.3.1 Бизнес процессы предметной области до использования разрабатываемой информационно-вычислительной системы
 - 1.3.2 Бизнес процессы предметной области после внедрения разрабатываемой информационно-вычислительной системы
 - 1.3.3 Обоснование целесообразности разработки информационно-вычислительной системы
 - 1.4 Требования пользователя к информационно-вычислительной системе
 - 1.4.1 Прецеденты информационно-вычислительной системы
 - 1.4.2 Требования к данным прецедентов информационно-вычислительной системы
 - 1.4.2.1 Данные прецедента «...»
Входные данные прецедента «...» и их ограничения.
Хранимые данные прецедента «...» и их ограничения.
Выходные данные прецедента «...» и их ограничения.
 - 1.4.2.2 Данные прецедента «...»
Входные данные прецедента «...» и их ограничения.
Хранимые данные прецедента «...» и их ограничения.
Выходные данные прецедента «...» и их ограничения.
 - 1.4.3 Сценарии прецедентов информационно-вычислительной системы
 - 1.4.3.1 Сценарий прецедента «...»
 - 1.4.3.2 Сценарий прецедента «...»
 - 1.4.4 Описание запросов к информационной части системы
 - 1.4.5 Требования к интерфейсу пользователей информационно-вычислительной системы
 - 1.4.5.1 Внешний вид форм для входных данных
 - 1.4.5.2 Внешний вид форм для выходных данных

1.4.5.3 Внешний вид печатных документов для выходных данных

1.4.6 Требования к архитектуре информационно-вычислительной системы

1.4.6.1 Архитектура программного обеспечения

1.4.6.2 Архитектура аппаратной части

1.4.7 Требования к характеристикам информационно-вычислительной системы

1.4.8 Требования к инструментам для разработки информационно-вычислительной системы

1.4.9 Требования к ресурсам, необходимым для разработки информационно-вычислительной системы

2 Технический проект

2.1 Словарь предметной области информационно-вычислительной системы

2.2 Концептуальная модель предметной области информационной части системы, реализованная на основе диаграммы концептуальных классов

2.2.1 Диаграмма концептуальных классов локальных представлений на уровне пакетов

2.2.2 Концептуальная модель локального представления «...» на уровне концептуальных классов

2.2.3 Концептуальная модель локального представления «...» на уровне концептуальных классов

2.3 Концептуальная модель предметной области программной части системы, реализованная на основе диаграммы концептуальных классов

2.3.1 Диаграмма концептуальных классов локальных представлений на уровне пакетов

2.3.2 Концептуальная модель локального представления «...» на уровне концептуальных классов

2.3.3 Концептуальная модель локального представления «...» на уровне концептуальных классов

2.4 Моделирование прецедентов при помощи диаграмм последовательностей системных операций на основе диаграммы концептуальных классов

2.4.1 Прецедент «»

2.4.2 Прецедент «»

2.5 Формализация требований высокого уровня

2.6 Моделирование системной операции «» прецедента «» при помощи диаграммы последовательности используемых ей функций

- 2.6.1 Диаграмма последовательности функций
 - 2.6.1.1 Описание функции «», используемых системной операцией «»
 - 2.6.1.2 Моделирование функции «», используемой системной операцией «», при помощи диаграммы деятельности/активности
 - 2.6.1.3 Тестовые наборы функции «» для модульного тестирования, разработанные методом черного или серого ящика
- 2.7 Определение предлагаемого и требуемого интерфейса для концептуальных классов
 - 2.7.1 Интерфейсы класса «»
 - 2.7.1.1 Интерфейс, предлагаемый классом «»
 - 2.7.1.2 Интерфейс, требуемый классом «»
- 2.8 Моделирование структуры программно-информационной системы с использованием шаблонов объектно-ориентированного проектирования в виде диаграммы проектных классов
 - 2.8.1 Моделирование структуры на уровне пакетов локальных представлений
 - 2.8.1.1 Моделирование структуры локального представления «...» на уровне проектных классов
 - 2.8.1.2 Моделирование структуры локального представления «...» на уровне проектных классов
- 2.9 Моделирование системных операций при помощи диаграмм последовательностей функций на основе диаграммы проектных классов
- 2.10 План тестирования и тестовые наборы для интеграционного тестирования системных операций
- 2.11 Проектирование информационной части системы на основе диаграммы проектных классов
 - 2.11.1 Сущности и их атрибуты
 - 2.11.2 Описание ограничений, накладываемых на данные, объекты и связи между объектами
 - 2.11.3 Моделирование схемы базы данных (при наличии базы данных) на основе диаграммы проектных классов
- 2.12 Моделирование взаимодействия программной и информационной части системы с использованием шаблонов объектно-ориентированного проектирования
- 2.13 Набор тестов для системного и интеграционного тестирования
- 2.14 Набор тестов для нагрузочного тестирования системы

2.15 Формализация требований низкого уровня и трассируемость требований низкого уровня на требования высокого уровня

2.16 Формализация описания тестов и трассируемость тестов на требования низкого уровня

3 Рабочий проект

3.1 Моделирование структуры информационно-вычислительной системы с использованием шаблонов объектно-ориентированного проектирования в виде диаграммы проектных классов системы и классов графического интерфейса пользователя

3.2 Моделирование структуры графического интерфейса пользователя на основании диаграммы объектов классов графического интерфейса

3.3 Описание сценариев физических прецедентов на основании диаграммы объектов классов графического интерфейса пользователя

3.4 Моделирование поведения интерфейса на основании диаграммы состояний объектов графического интерфейса пользователя

3.5 Исходные коды и свойства объектов графического интерфейса пользователя для проверки валидности ограничений, накладываемых на входные данные на основе используемых языков программирования

3.6 Генерация прототипов исходного кода системы

3.6.1 Моделирование и классов, объектов, функций и структур данных с применением инженерных сред разработки для используемых языков программирования и СУБД

3.6.2 Генерация прототипов исходных кодов классов, объектов, функций и структур данных в конкретных средах разработки на основе построенных моделей

3.9 Описание окончательного варианта исходного кода элементов программной модели и формирование программных компонент

3.10 Формализация описания элементов программной модели

3.11 Трассируемость элементов программной модели на требования низкого уровня

3.12 Моделирование разработанной программной структуры (архитектуры) программно-информационной системы на основании диаграммы компонент исходных кодов

3.13 Трассируемость разработанных элементов структуры программно-информационной системы на требования высокого уровня

уровня, предъявленные пользователем к программной архитектуре программно-информационной системы

3.14 Тестовые наборы для модульного, интеграционного и системного тестирования, разработанные на основании критериев покрытия исходного кода

3.15 Трассируемость тестовых наборов на требования низкого уровня

3.16 Драйверы для тестирования

3.16.1 Драйверы для модульного тестирования

3.16.2 Драйверы и «заглушки» для интеграционного тестирования

3.16.3 Драйверы для нагрузочного тестирования

3.17 Моделирование разработанной программной структуры (архитектуры) программно-информационной системы на основании диаграммы компонент исполняемых кодов

3.18 Трассируемость разработанных элементов структуры программно-информационной системы на уровне исполняемых кодов на требования высокого уровня, предъявленные пользователем к аппаратной архитектуре программно-информационной системы

3.19 Сборка программных компонент в единый проект, генерация исполняемого кода и его размещение в аппаратной части системы

3.19.1 Описание плана сборки программных компонент системы на уровне исполняемых кодов

3.19.2 Диаграмма развертывания системы

3.20 Определение характеристик информационно-вычислительной системы

3.21 Временная сложность информационно-вычислительной системы

3.22 Емкостная сложность информационно-вычислительной системы

Заключение содержит конкретные выводы, которые соотносятся с целью и задачами, поставленными во введении, а также включает предложения и рекомендации по использованию полученных результатов в производственной деятельности.

Список использованных источников содержит сведения об источниках, использованных при выполнении КР (КП).

В Приложениях размещаются:

Приложение А Внешний вид графического материала

Приложение Б Текст исходных кодов

4 Типовые варианты заданий

4.1 Система электронного голосования

Спроектировать и разработать информационно-вычислительную систему для проведения выборов. Избиратели должны предварительно зарегистрироваться в системе. Администратор заполняет список кандидатов. Каждый участник (по сети) или с того же самого компьютера входит в систему и голосует. Данные о проголосовавших накапливаются в базе данных. Дважды проголосовать нельзя. По окончании периода голосования администратор запускает процедуру подсчета голосов и система выдает результат.

Основные алгоритмы: подведение итогов выборов.

Рекомендуемая диаграмма классов: представлена на Рисунок 1.

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для регистрации пользователей, интерфейс для создания голосования, интерфейс для голосования, интерфейс для проверки результатов голосования.

Таблицы в базе данных: выборы (наименование, сроки и т.д.), кандидаты, пользователи.

Модель клиент-сервер: предусмотреть голосование по сети.

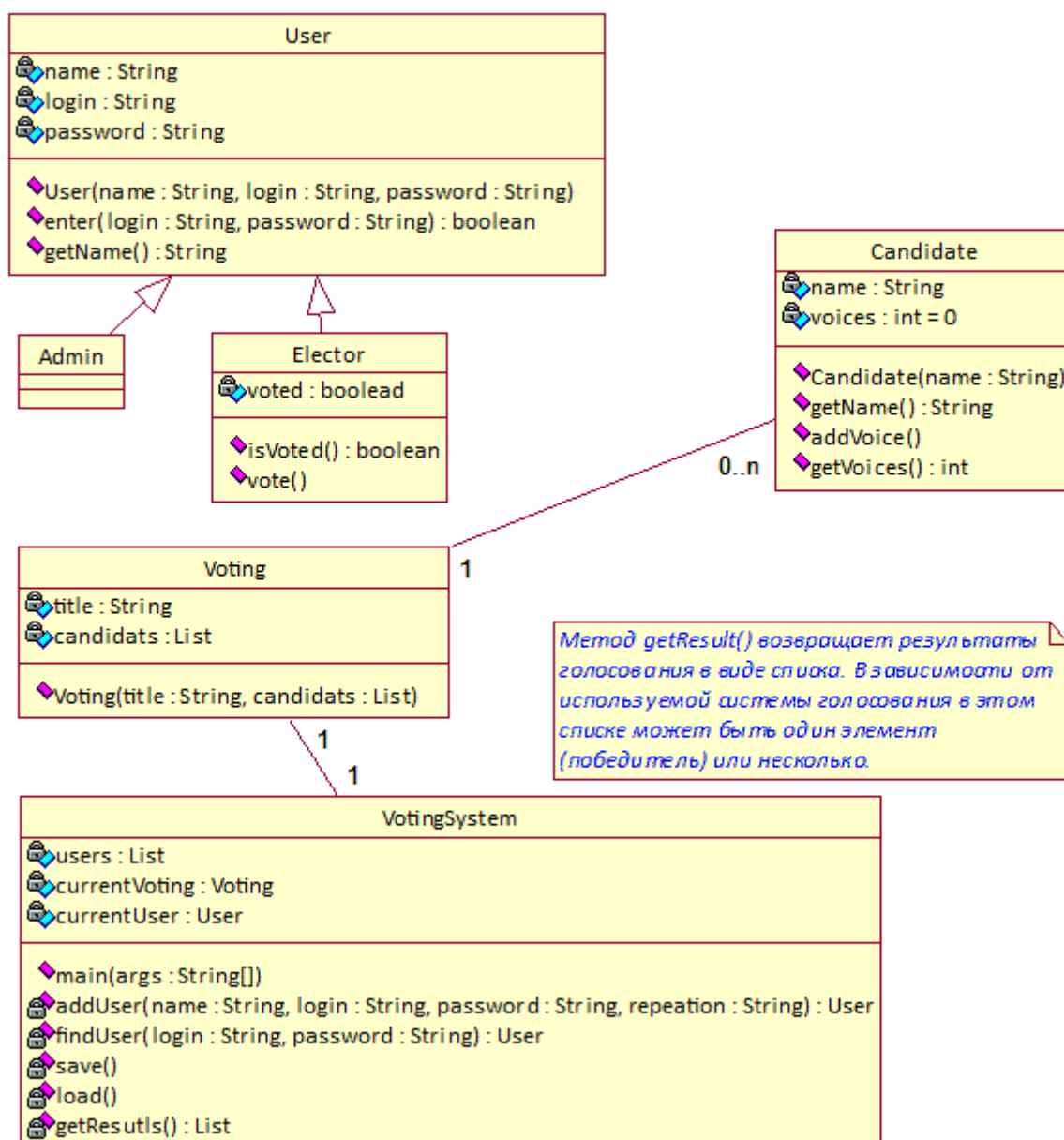


Рисунок 1

4.2 Обработка заказов на доставку пиццы

Спроектировать и разработать информационно-вычислительную систему для составления оптимального маршрута разносчика пиццы. Пользователи, желающие заказать пиццу, оставляют в системе заявку, указывая район своего проживания. Кроме того система постоянно накапливает информацию о способах передвижения между соседними районами города. Когда поступает заказ, система должна найти короткий путь от текущего местонахождения вагончика с пиццей до района проживания заказчика.

Основные алгоритмы: нахождение кратчайшего маршрута.

Рекомендуемая диаграмма классов: представлена на Рисунок 2.

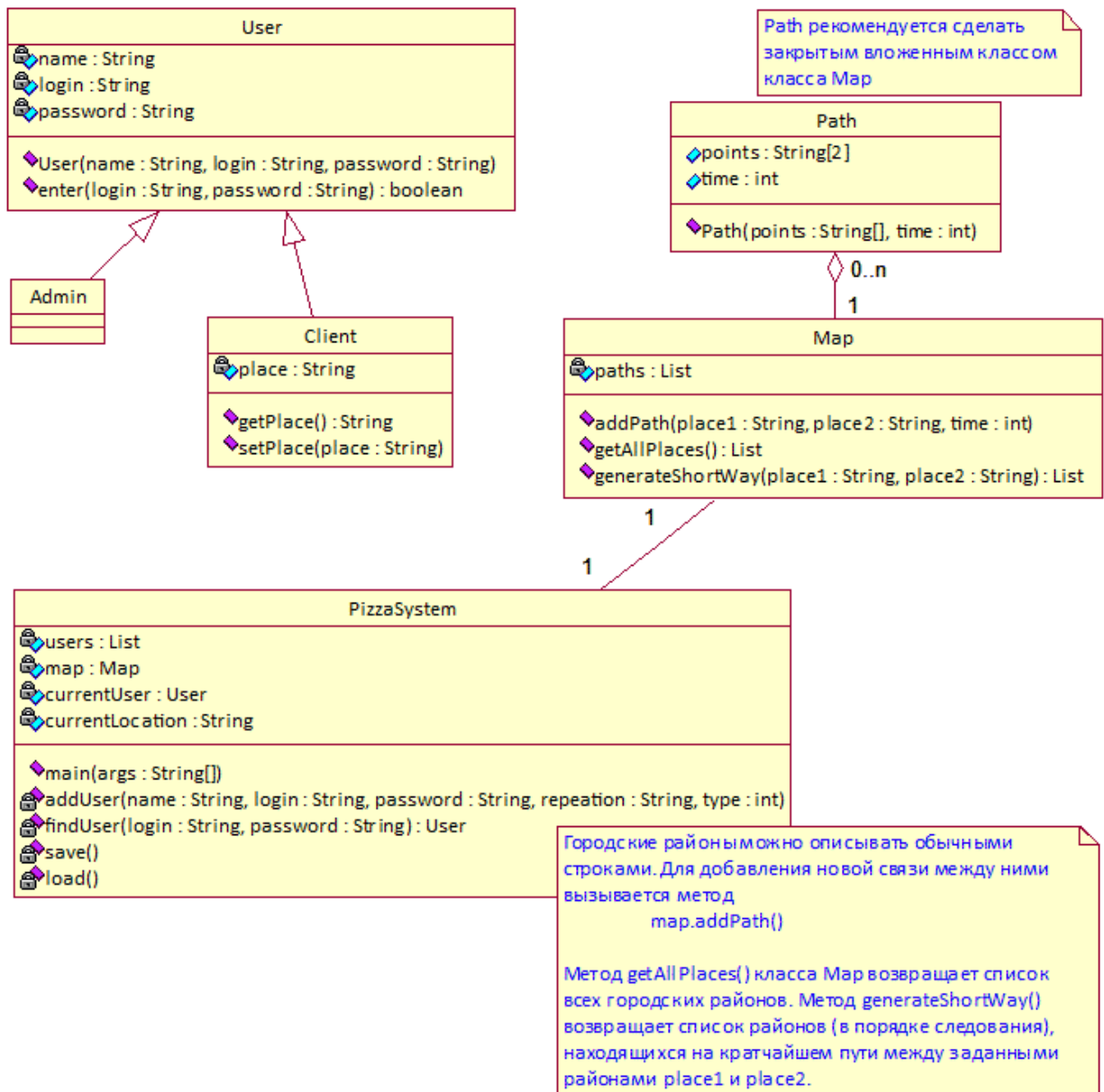


Рисунок 2

Графические интерфейсы пользователей: интерфейс для создания «карты» (на самом деле это просто граф) города, интерфейс для входа в систему. интерфейс для регистрации пользователей, интерфейс для заполнения заявки. интерфейс для обработки результатов.

Таблицы в базе данных: пользователи, карта города (районы и связи между ними).

Модель клиент-сервер: предусмотреть возможность делать заказ и уточнять карту города по сети.

4.3 Ипподромный тотализатор

Спроектировать и разработать информационно-вычислительную систему для приема ставок и расчета выигрышей на скачках.

Пользователи, зарегистрированные в системе, могут поставить любую сумму на одну из семи лошадей. Предварительно администратор системы вносит список лошадей. По результатам заезда (его можно смоделировать на основе случайных чисел) должна рассчитываться сумма выигрыша.

Желательно, чтобы система могла обрабатывать несколько видов ставок (какая лошадь придет первой, какая лошадь придет последней и т.д.).

Основные алгоритмы: моделирование результатов забега на основе случайных чисел (метод `generateResults()`), подсчет и распределение суммы выигрыша (метод `calculateMoney()`).

Рекомендуемая диаграмма классов: представлена на Рисунок 3.

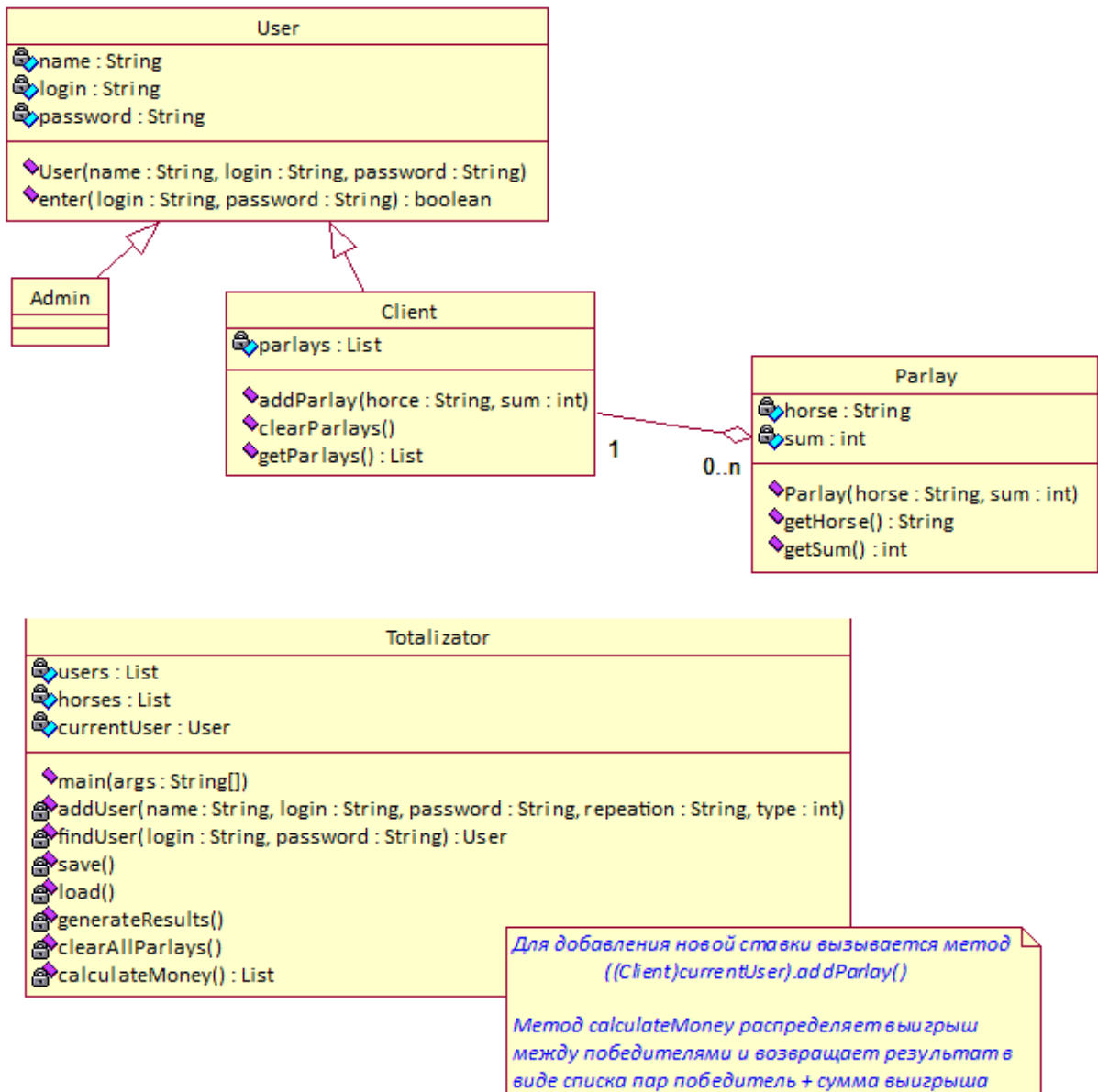


Рисунок 3

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для регистрации пользователей, интерфейс для описания забега, интерфейс для приема ставок, интерфейс для подсчета суммы выигрыша.

Таблицы в базе данных: пользователи, ставки, лошади.

Модель клиент-сервер: предусмотреть возможность делать ставки по сети.

4.4 Система распределения вакансий на рынке труда

Спроектировать и разработать информационно-вычислительную систему для распределения вакансий. Пользователи (работники и работодатели) регистрируются в системе, после чего могут делать заявки. Можно предложить работу (указав требования к работнику, зарплату и другие параметры), либо искать работу (указать желаемую должность, зарплату и т.д.). После накопления достаточного количества заявок система приступает к их распределению, чтобы максимально учесть пожелания пользователей и возвращает результат в виде пар работник-работодатель.

Основные алгоритмы: распределение заявок.

Рекомендуемая диаграмма классов: представлена на Рисунок 4.

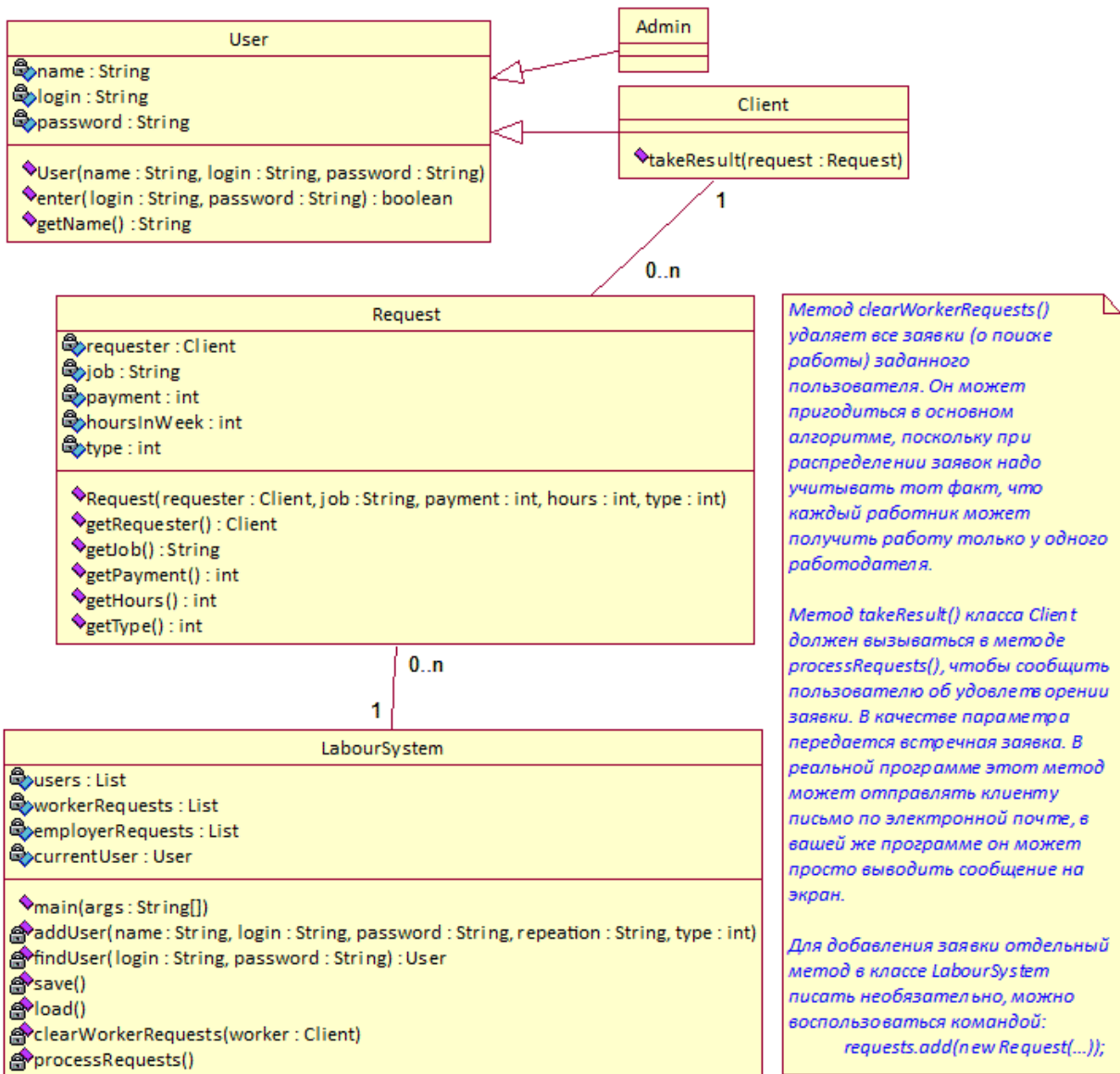


Рисунок 4

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для регистрации пользователей, интерфейс для регистрации заявки, интерфейс для распределения заявок.

Таблицы в базе данных: текущие заявки, пользователи.

Модель клиент-сервер: предусмотреть подачу заявок по сети.

4.5 Система составления расписания

Спроектировать и разработать информационно-вычислительную систему для автоматического составления расписания занятий. Пользователи, зарегистрированные в системе (преподаватели) делают заявки (указывают номер студенческой группы, название предмета и количество занятий в неделю). Администратор указывает количество свободных аудиторий (и другие параметры при необходимости). Накопив все заявки, система должна составить расписание. Желательно, чтобы расписание было как можно более качественным

(отсутствие «дырок» у студентов и преподавателей, равномерная загруженность).

Основные алгоритмы: автоматизированное составление расписания.

Рекомендуемая диаграмма классов: представлена на Рисунок 5.

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для регистрации пользователей (преподавателей), интерфейс администратора (для указания числа аудиторий и др.), интерфейс для приема заявок, интерфейс для составления расписания.

Таблицы в базе данных: пользователи, заявки, информация об университете (число аудиторий и др.).

Модель клиент-сервер: предусмотреть возможность делать заявки по сети.

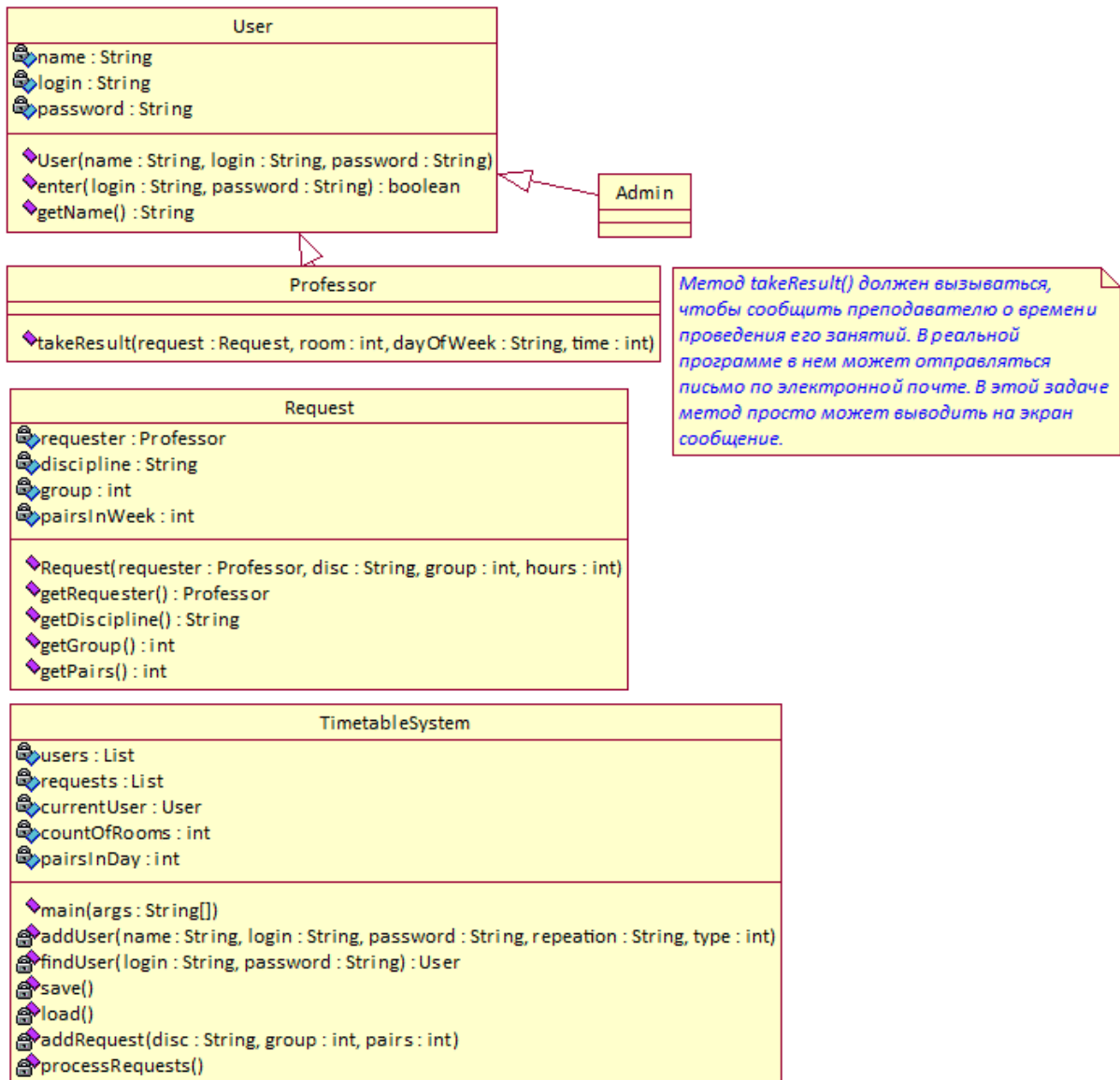


Рисунок 5

4.6 Оболочка для тестирования

Спроектировать и разработать информационно-вычислительную систему для проведения тестирования. Администратор должен иметь возможность добавлять в программу вопросы (указывая при этом варианты ответов и отмечая правильный вариант). Пользователи проходят тестирование, отвечая на эти вопросы и получая оценку в виде процента правильных ответов.

Желательно, чтобы система позволяла переключаться между тестами. Т.е. администратор может иметь возможность создавать новые тесты и при добавлении вопроса указывать, к какому тесту он относится, а пользователь — выбирать нужный тест из списка.

Основные алгоритмы: вычисление результата тестирования.

Рекомендуемая диаграмма классов: представлена на Рисунок 6.

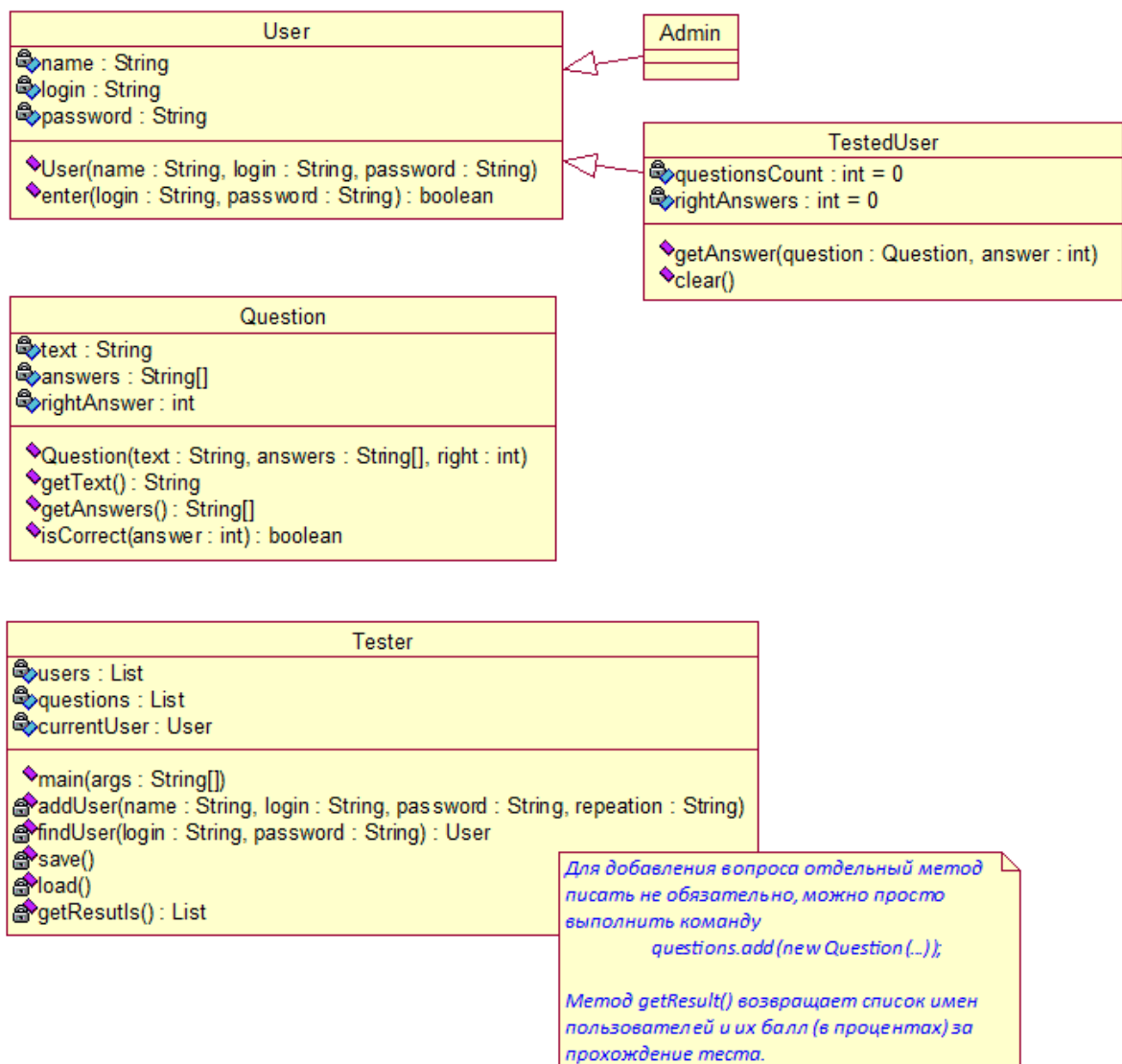


Рисунок 6

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для регистрации пользователей, интерфейс для

добавления вопросов (и тестов), интерфейс для прохождения тестирования.

Таблицы в базе данных: вопросы, пользователи (и результаты их тестирования).

Модель клиент-сервер: предусмотреть возможность тестирования по сети.

4.7 Система для выбора места проведения саммита

Спроектировать и разработать информационно-вычислительную систему для лидеров стран, собирающихся организовать саммит. Каждый лидер может предложить несколько вариантов времени и места встречи. Место встречи — страна (может вводиться вручную или выбираться из списка уже введенных ранее стран), время задается сочетанием месяц + с какого дня + до какого дня. Система подводит итоги, определяя возможные варианты времени и места встречи, подходящие для всех лидеров стран. Если таких вариантов нет, лидеры стран могут продолжать добавлять свои варианты.

Основные алгоритмы: нахождение возможностей для проведения саммита.

Рекомендуемая диаграмма классов: представлена на Рисунок 7.

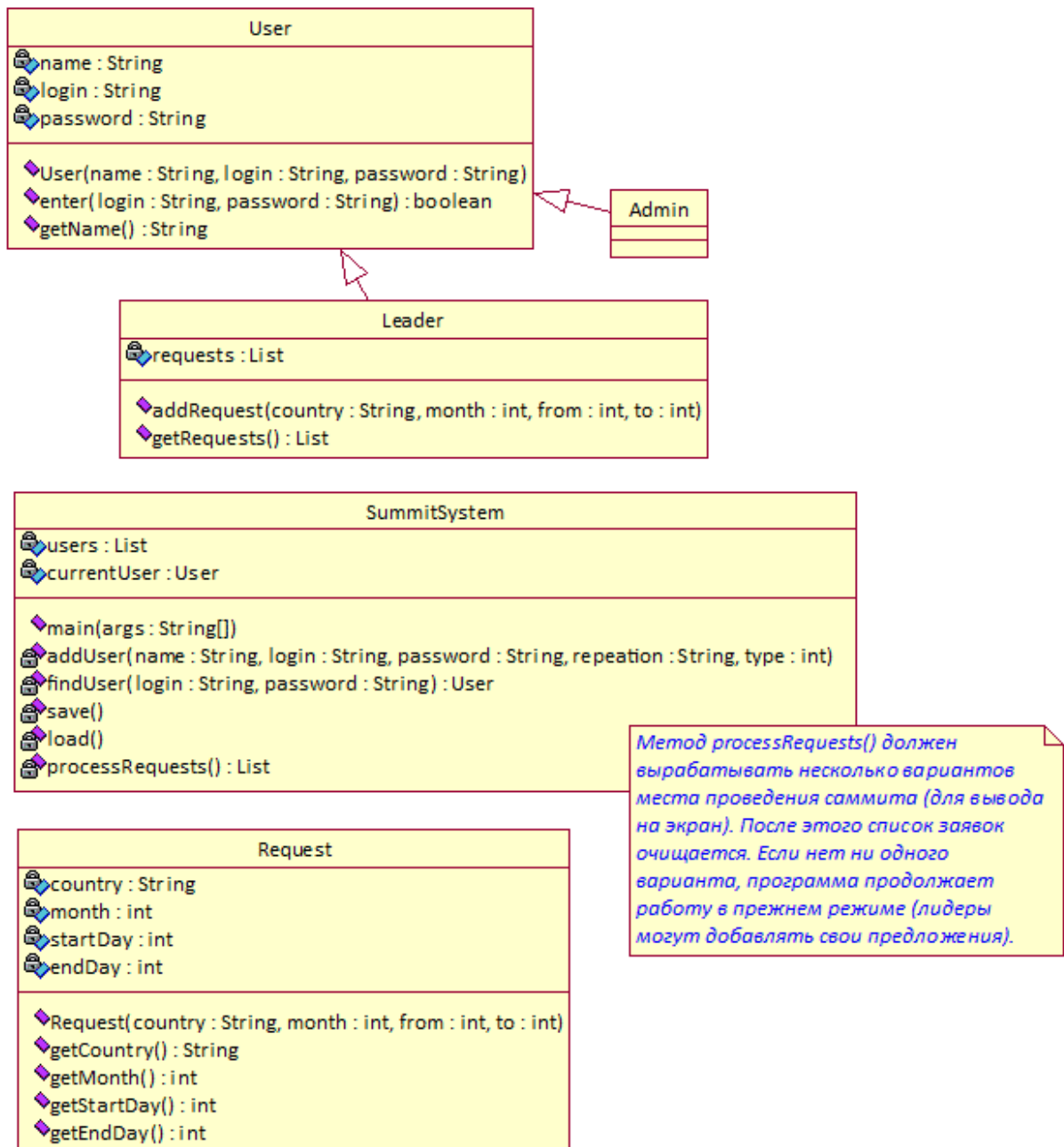


Рисунок 7

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для регистрации пользователей, интерфейс для добавления варианта и проведения саммита, интерфейс для определения результатов.

Таблицы в базе данных: пользователи, варианты мест и дат проведения саммита.

Модель клиент-сервер: варианты можно указывать по сети.

4.8 Игра «морской бой»

Спроектировать и разработать информационно-вычислительную систему для игры в морской бой. Пользователи регистрируется в

системе и указывают расположение своих кораблей. Затем пользователи по очереди пытаются «попасть» в корабли противника (вводя координаты в текстовое поле или указывая мышкой клетку на карте, что предпочтительнее). В упрощенной версии играть могут два пользователя и после каждого хода система завершает «сеанс» связи с текущим пользователем и запрашивает пароль следующего.

В более продвинутой версии система может поддерживать сеанс одновременной игры с несколькими пользователями (в этом случае надо указать, чьи корабли следует бомбардировать).

В сетевой версии игры можно сделать возможность, чтобы игра не прерывалась после каждого хода, а просто блокировалась в ожидании хода противника.

Основные алгоритмы: проверка попадания по кораблю и проверка окончания игры.

Рекомендуемая диаграмма классов: представлена на Рисунок 8.

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для регистрации и расстановки кораблей, интерфейс для выстрела.

Таблицы в базе данных: игроки, позиции кораблей и «прострелянных» точек каждого игрока.

Модель клиент-сервер: игра по сети.

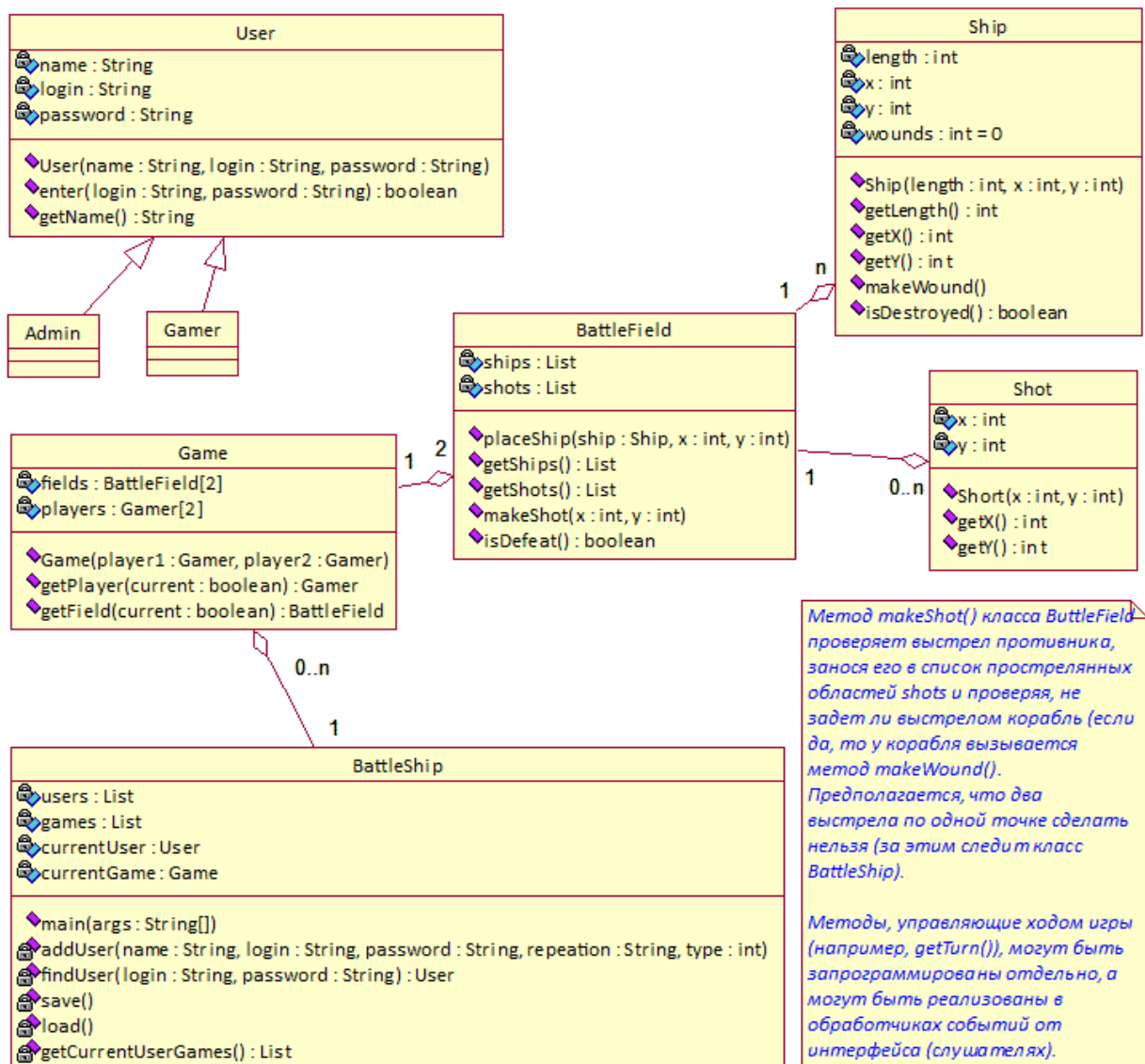


Рисунок 8

4.9 Система дистанционного обучения

Спроектировать и разработать информационно-вычислительную систему для проведения удаленного экзамена. Преподаватель готовит список вопросов к экзамену, указывая для каждого из них правильный ответ. Студенты сдают экзамен, отвечая на пять вопросов, случайным образом выбранных системой и получают оценку.

Желательно сделать так, чтобы системой могли пользоваться несколько преподавателей. Студент после регистрации должен выбрать предмет. Если экзамен уже сдан, его в этом списке не будет.

Основные алгоритмы: проверка ответов и определение оценки.

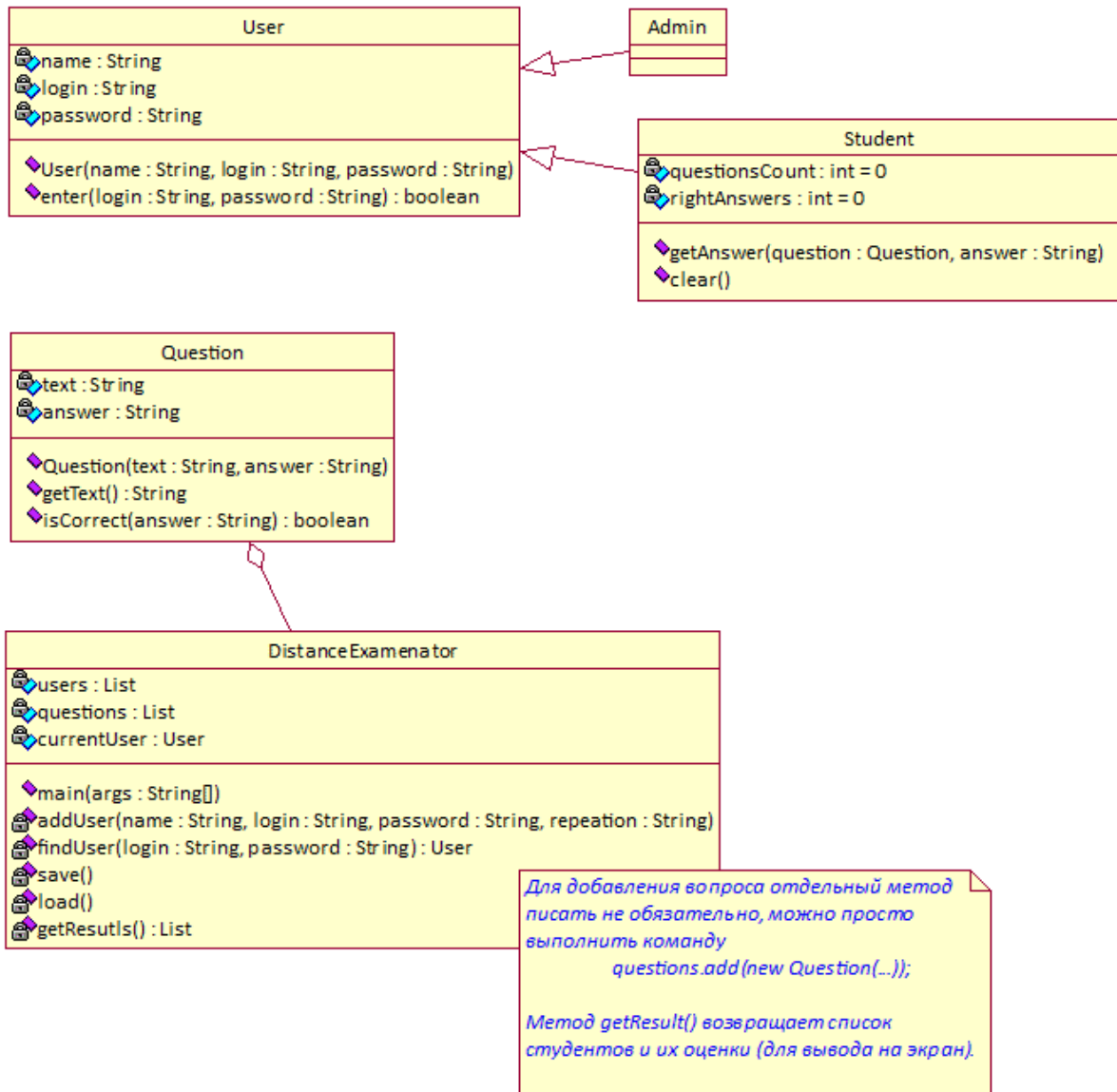


Рисунок 9

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для регистрации студентов, интерфейс для добавления вопросов, интерфейс для сдачи экзамена.

Таблицы в базе данных: вопросы, студенты (и их оценки).

Модель клиент-сервер: студент отвечает на вопросы по сети.

4.10 Система контроля за пробками

Спроектировать и разработать информационно-вычислительную систему для автомобилистов, отслеживающую возникающие на дорогах пробки и предлагающую наименее загруженный путь между двумя точками. Автомобилисты могут использовать эту программу, чтобы сообщить о пробке на пути от одной точки города до соседней (этими точками могут быть смежные районы города). Каждая пробка

получает оценку загруженности по десятибалльной системе. Система также по запросу для двух точек находит маршрут, на котором общая продолжительность стояния в пробках минимальна (разницу в расстоянии между разными смежными районами мы игнорируем).

Основные алгоритмы: поиск пути сквозь пробки.

Рекомендуемая диаграмма классов: представлена на Рисунок 10.

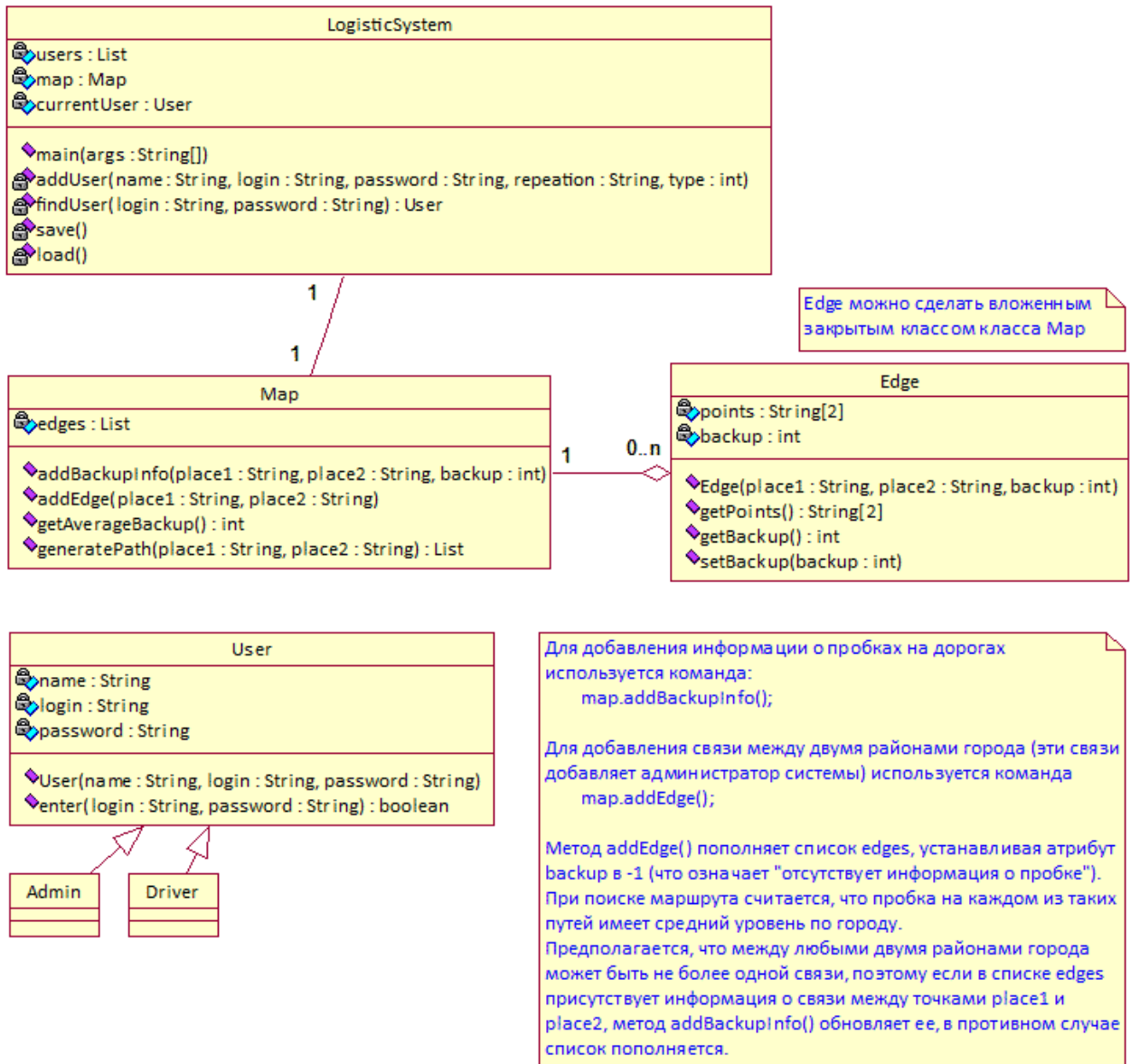


Рисунок 10

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для регистрации пользователей, интерфейс для сообщения о пробке (или ее отсутствии), интерфейс для запроса, интерфейс для отображения составленного маршрута.

Таблицы в базе данных: пользователи, районы (точки), связи между районами (с указанием загруженности).

Модель клиент-сервер: предусмотреть возможность делать запросы и сообщать о пробках по сети.

4.11 Конкурс проектов

Спроектировать и разработать информационно-вычислительную систему для проведения конкурса. Система должна распределять грантовый фонд конкурса между участниками в зависимости от экспертных оценок, выставленных каждому участнику. При этом сумма, выделенная каждому участнику не может быть меньше некоторой минимальной величины (участники, «заработавшие» ниже минимума, из конкурса выбывают). Заявки от участников поступают уже с проставленными экспертными оценками. Процедуру определения результатов конкурса запускает администратор (после того, как заканчивается срок приема заявок).

Основные алгоритмы: определение результатов конкурса.

Рекомендуемая диаграмма классов: представлена на Рисунок 11.

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для регистрации пользователей, интерфейс для заполнения параметров конкурса (общий фонд и минимальная сумма), интерфейс для подачи заявки, интерфейс для расчета результатов.

Таблицы в базе данных: пользователи, параметры конкурса, заявки.

Модель клиент-сервер: предусмотреть возможность делать заявки по сети.

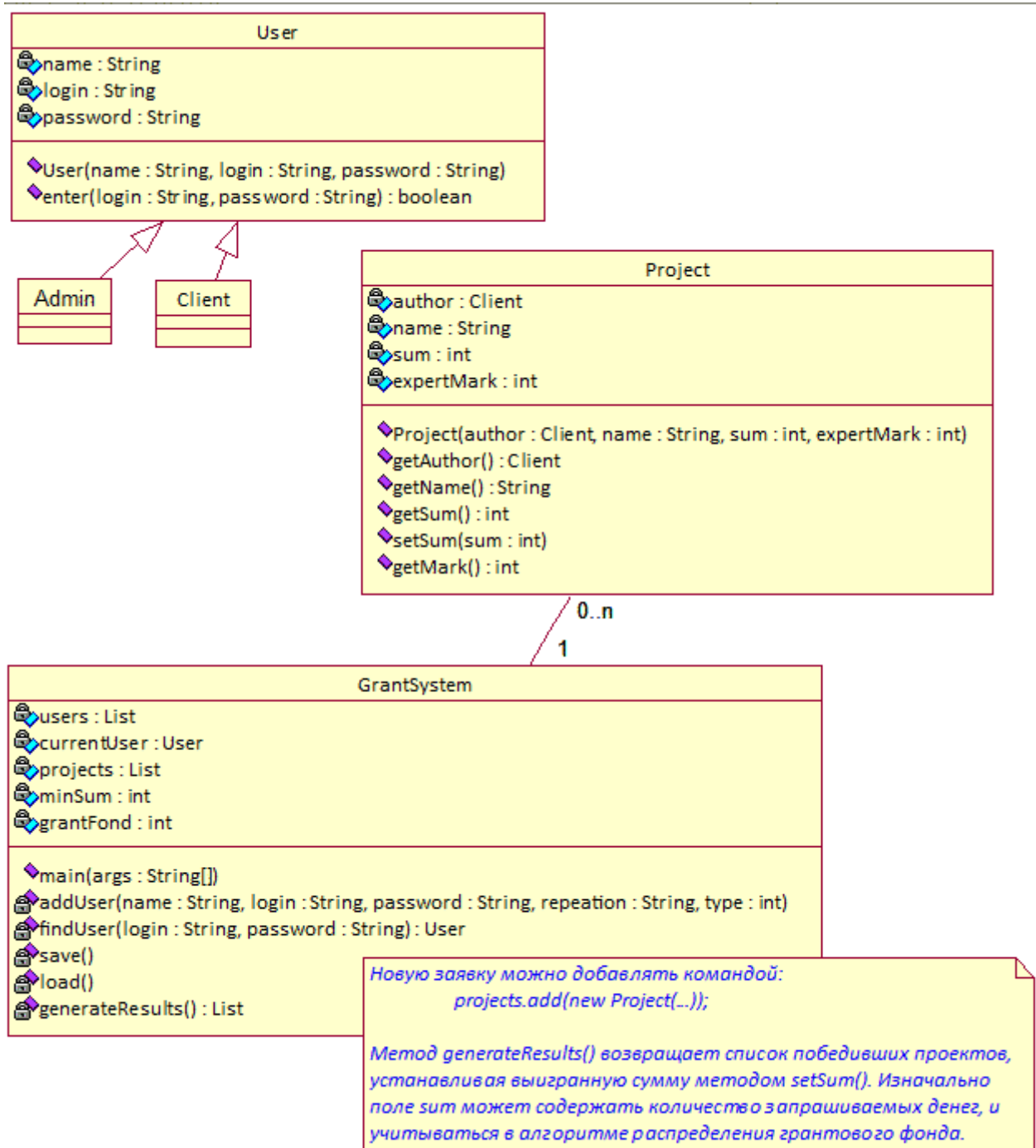


Рисунок 11

4.12 Бизнес-аукцион

Спроектировать и разработать информационно-вычислительную систему для проведения торгов по заявкам. Участники регистрируются в системе и подают заявки на продажу и закупку товара (кроме наименования и цены товара могут указываться дополнительные параметры, такие как качество). В конце дня система (по команде администратора) сопоставляет заявки и выдает результаты торгов, пытаясь максимально удовлетворить потребности участников. Заявки, которые остались не удовлетворены, переносятся на следующий день.

Основные алгоритмы

Расчет результатов торгов (на основе некоторого критерия оптимальности)

Рекомендуемая диаграмма классов: представлена на Рисунок 12.

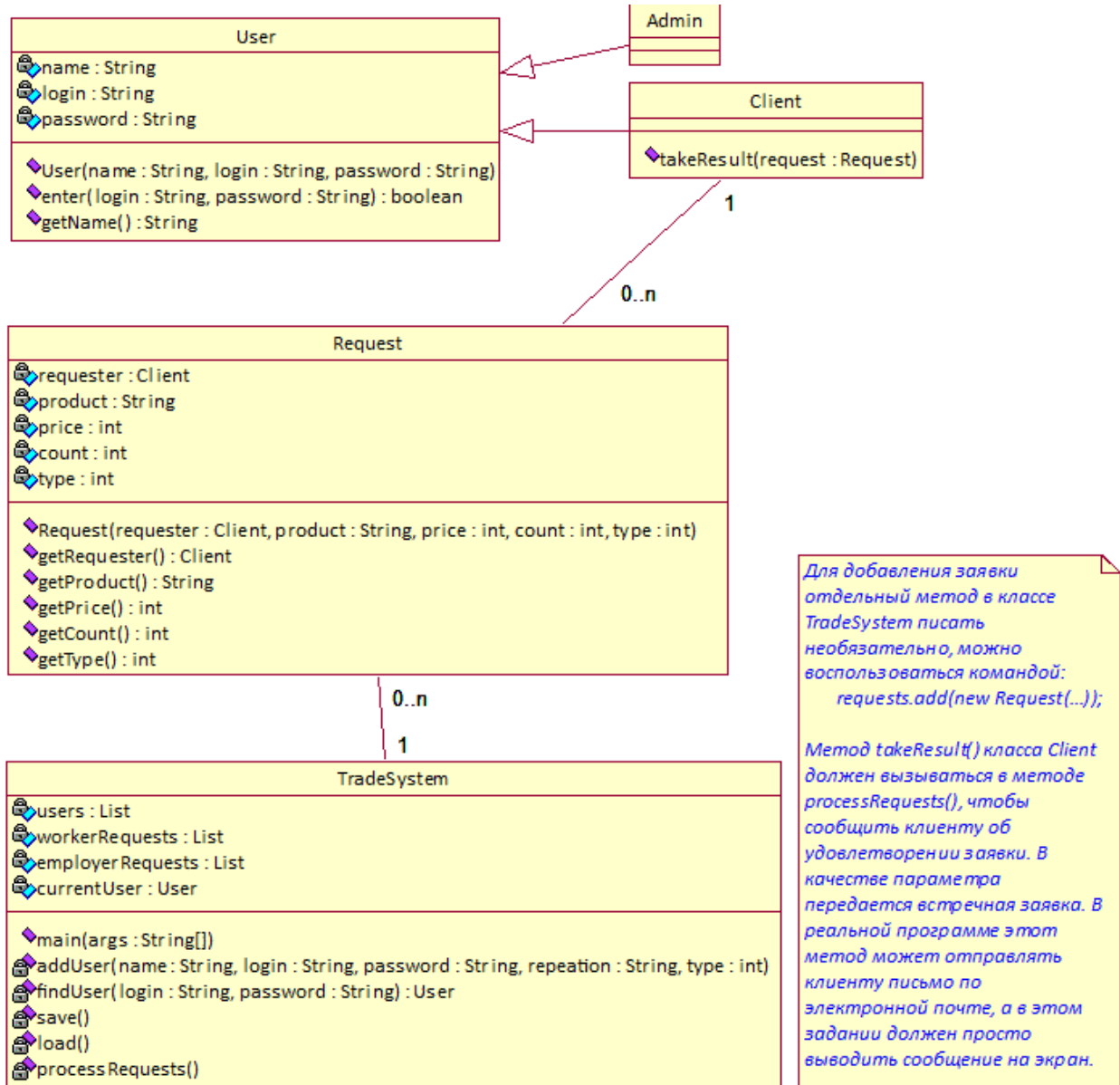


Рисунок 12

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для вывода результатов.

Таблицы в базе данных: участники торгов, заявки.

Модель клиент-сервер: предусмотреть возможность подачи заявок по сети.

4.13 Система для координации участников проекта

Необходимо разработать информационно-вычислительную систему для координации группы участников проекта. Участникам проекта необходимо назначить время и место встречи. Каждый участник регистрируется в системе и

указывает свои возможности (список промежутков времени и мест, где он может встретиться). Система подводит итоги, определяя возможные варианты времени и места встречи, подходящие для всех участников.

Возможное время встречи должно задаваться сочетанием день недели + с какого часа + до какого часа. Место встречи выбирается из списка, который заполняется заранее.

Основные алгоритмы: определение вариантов места встречи.

Рекомендуемая диаграмма классов: представлена на рисунке 13

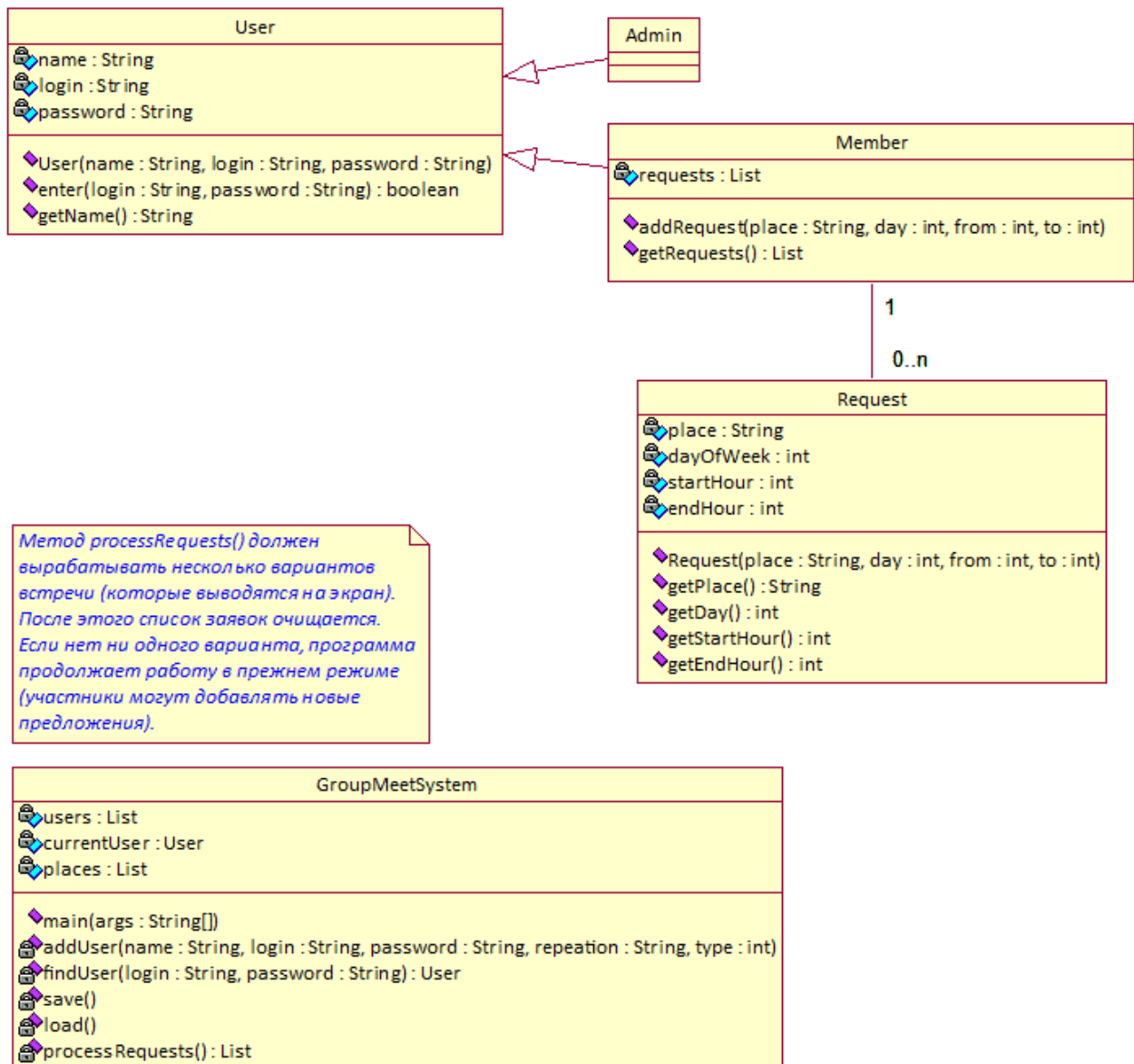


Рисунок 13

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для регистрации пользователей, интерфейс для указания всех возможных мест встречи, интерфейс для добавления возможностей, интерфейс для расчета результатов.

Таблицы в базе данных: пользователи, места встречи, возможности.

Модель клиент-сервер: возможности можно указывать по сети.

4.14 Система оптимального распределения производственных мощностей

Спроектировать и разработать информационно-вычислительную систему для оптимального распределения производственных мощностей. В производственный отдел поступают заказы на изготовление продукции. Для выполнения каждого заказа требуется определенное количество ресурсов. Пусть в качестве ресурсов выступает время использования оборудования. Для простоты предположим, что список необходимого оборудования и времени содержится в самом заказе, причем список упорядочен в соответствии с технологическими операциями заказа (т.е. имеет значение, в каком порядке это оборудование будет предоставлено). Система должна разработать очередность предоставления оборудования, с целью минимизировать общее время выполнения всей партии заказов.

Основные алгоритмы: разработка очередности предоставления оборудования.

Рекомендуемая диаграмма классов: представлена на Рисунок 14.

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для регистрации пользователей, интерфейс для указания перечня оборудования, интерфейс для приема заказов, интерфейс для составления плана загрузки оборудования.

Таблицы в базе данных: пользователи, заказы, оборудование.

Модель клиент-сервер: предусмотреть возможность делать заказы по сети.

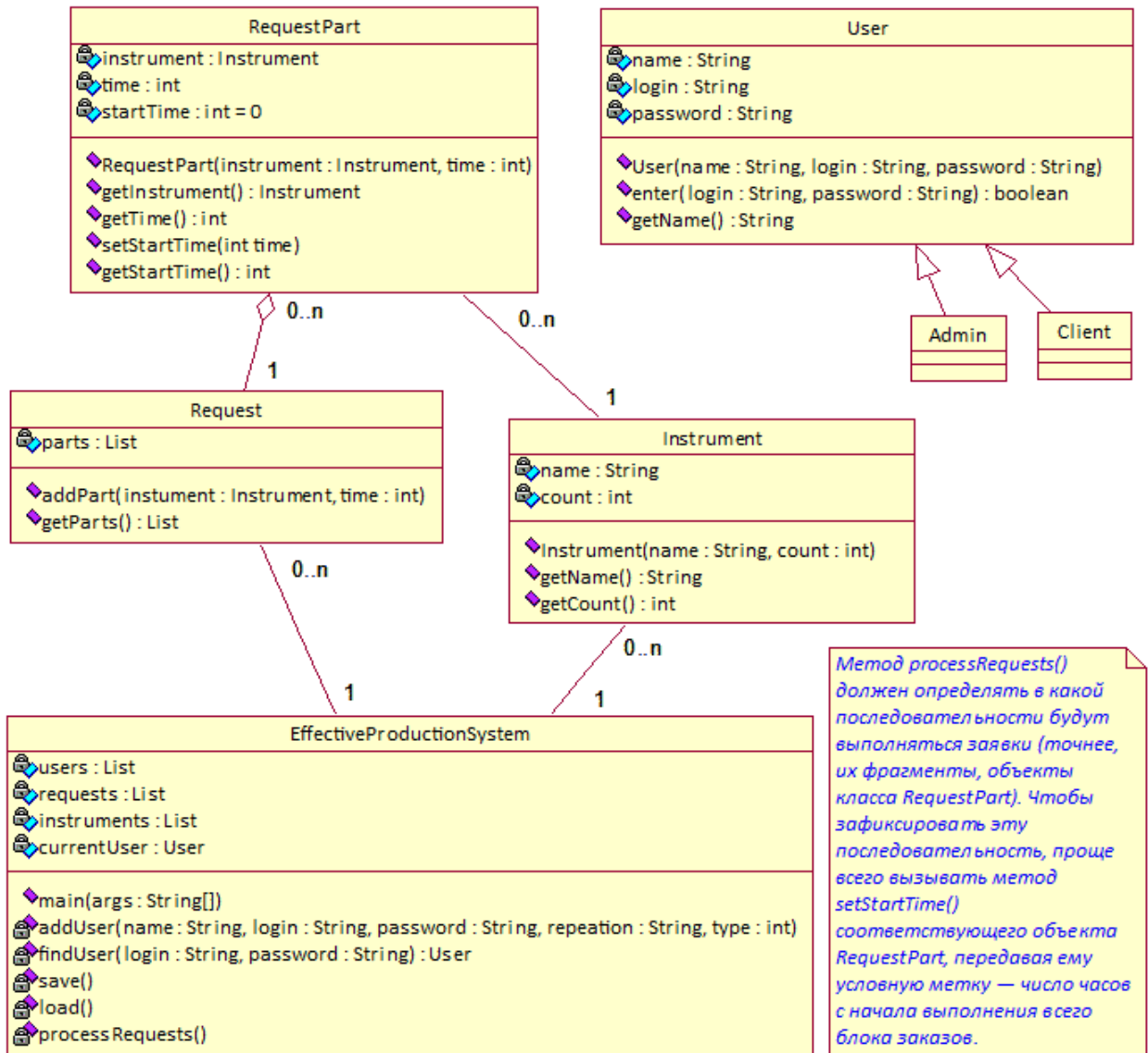


Рисунок 14

4.15. Концерт по заявкам

Разработать информационно-вычислительную систему для формирования программы концерта по заявкам. Пользователи (фанаты) регистрируются в системе и выбирают песни из предложенного списка (или добавляют свои). Каждый пользователь может выбрать любое количество песен. Когда время подачи заявок оканчивается, формируется программа концерта, включающая песни, набравшие наибольшее число заявок (количество песен определяется заранее).

Основные алгоритмы: составление программы концерта по заявкам.

Рекомендуемая диаграмма классов: представлена на Рисунок 15.

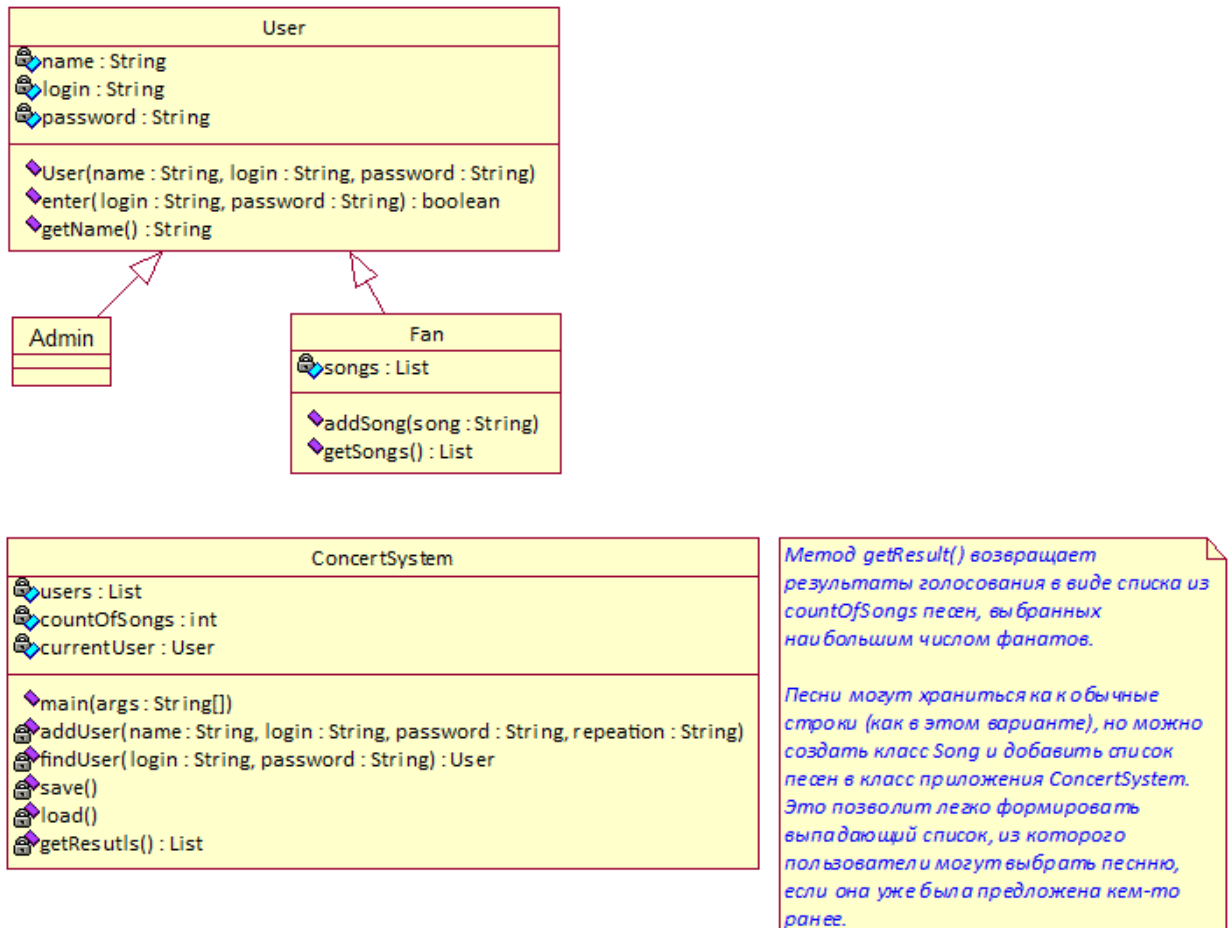


Рисунок 15

Графические интерфейсы пользователей: интерфейс для входа в систему, интерфейс для регистрации пользователей, интерфейс администратора (указание основных параметров концерта: название, цена билетов, место проведения, из которых нас интересует только число песен), интерфейс для добавления песни в список, интерфейс для подачи заявок, интерфейс для вывода результатов (программа концерта).

Таблицы в базе данных: пользователи, голоса.

Модель клиент-сервер: предусмотреть голосование по сети.

4.16. Игра «шашки»

Спроектировать и разработать информационно-вычислительную систему для игры в шашки. Пользователи регистрируются в системе и ходят по очереди. Текущая партия может быть приостановлена, а затем продолжена заново. В продвинутой версии системы может поддерживаться одновременно несколько партий.

Основные алгоритмы: проверка допустимости сделанного хода, проверка окончания игры.

Рекомендуемая диаграмма классов: представлена на Рисунок 16.

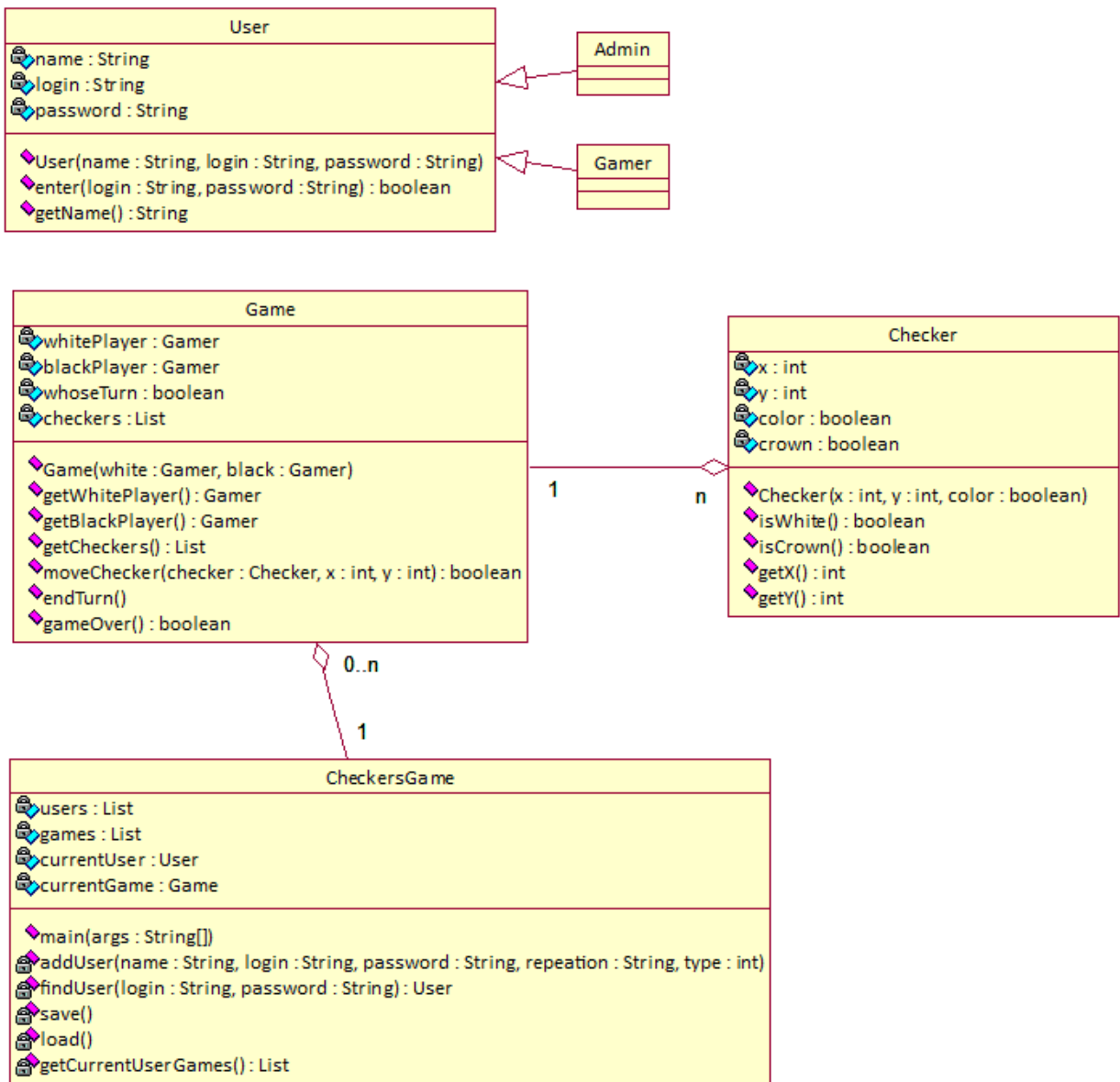


Рисунок 16

Графические интерфейсы пользователей: интерфейс для регистрации игроков, интерфейс входа в систему, интерфейс для начала и загрузки (выбора) партии, интерфейс для очередного хода, интерфейс для объявления результатов партии.

Таблицы в базе данных: игроки, партии, позиции шашек.

Модель клиент-сервер: игра по сети.

4.17 Комментарии к диаграммам классов

1. Классы, необходимые для построения пользовательского интерфейса, на предложенных диаграммах отсутствуют. Так же отсутствуют все детали, касающиеся клиент-серверной организации информационно-вычислительной системы. Эти классы должны быть разработаны самостоятельно на

соответствующем этапе работы. В целом предложенные диаграммы не являются догмой и могут быть усовершенствованы (или вообще изменены).

2. В каждой информационно-вычислительной системе есть пользователи двух типов: администратор и клиенты. Предполагается, что администратор в программно-информационной системе один. Он должен быть добавлен сразу после первого запуска программно-информационной системы.

ПРИЛОЖЕНИЕ**Минобрнауки России
Юго-Западный государственный университет**

Кафедра программной инженерии

КУРСОВАЯ РАБОТА (ПРОЕКТ)по дисциплине « _____ »
(наименование дисциплины)

на тему « _____ »

Направление подготовки (специальность) 09.04.04 «Программная
инженерия»
(код, наименование)Автор работы (проекта) _____
(инициалы, фамилия) (подпись, дата)

Группа _____

Руководитель работы (проекта) _____
(инициалы, фамилия) (подпись, дата)Работа (проект) защищена _____
(дата)

Оценка _____

Члены комиссии _____
(подпись, дата) (инициалы, фамилия)

(подпись, дата) (инициалы, фамилия)

(подпись, дата) (инициалы, фамилия)

Курск 20 __ г.

Минобрнауки России**Юго-Западный государственный университет****Кафедра программной инженерии****ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ (ПРОЕКТ)**

Студент _____ шифр _____ группа _____
(фамилия, инициалы)

Тема « _____ »

Срок представления работы (проекта) к защите « ____ » _____ 20 ____ г.

Исходные данные (для проектирования, для научного исследования):

Содержание пояснительной записки курсовой работы (проекта):

Введение

1 Техническое задание

2 Технический проект

3 Рабочий проект

Заключение

Перечень графического материала:

1. Диаграмма вариантов использования.
2. Концептуальная модель предметной области.

...

Руководитель работы (проекта) _____

(инициалы, фамилия) (подпись, дата)

Задание принял к исполнению _____

(инициалы, фамилия) (подпись, дата)

Минобрнауки России
Юго-Западный государственный университет
Кафедра программной инженерии

ОТЗЫВ

руководителя о выпускной квалификационной работе, курсовой работе
 (проекте) по программе бакалавриата, программе специалитета, программе
 магистратуры

(нужное подчеркнуть)

студента (слушателя) _____

(фамилия, имя, отчество)

группы _____ направления подготовки (специальности) 09.04.04 Программная инженерия
 На тему _____

1. Объем работы: количество страниц _____. Графическая часть _____ листов.

2. Цель и задачи исследования:

3. Актуальность, теоретическая, практическая значимость темы исследования:

4. Соответствие содержания работы по заданию (полное или неполное): полное

5. Основные достоинства и недостатки работы: Недостатков, существенно влияющих на оценку работы, нет.

6. Степень самостоятельности и способности к исследовательской работе (умение и навыки искать, обобщать, анализировать материал и делать выводы):

Данная курсовая работа выполнена студентом самостоятельно с применением современных средств создания программного обеспечения. Считаю, что подготовленность студента по дисциплине «Методология программной инженерии» позволит ему самостоятельно решать практические задачи.

7. Оценка деятельности студента в период выполнения работы (степень добросовестности, работоспособности, ответственности, аккуратности и т.п.):

Во время работы над курсовой работой студент проявил способность к проведению экспериментов, а также оптимизации рабочего времени. Данная курсовая работа выполнена с высокой степенью добросовестности и ответственности.

8. Достоинства и недостатки оформления текстовой части, графического, демонстративного, иллюстративного, компьютерного и информационного материала. Соответствие оформления требованиям стандарта:

Оформление курсовой работы, графического и иллюстративного материала соответствует требованиям стандарта университета СТУ 04.02.030 – 2017 «Курсовые работы (проекты). Выпускные квалификационные работы. Общие требования к структуре и оформлению». Текстовый материал курсовой работы выполнен грамотно на хорошем инженерном уровне.

9. Целесообразность и возможность внедрения результатов исследования:

10. Общее заключение и предлагаемая оценка работы:

Работа выполнена в полном объеме, рекомендуемая оценка «отлично»

Руководитель _____

(фамилия, имя, отчество, должность, ученая степень, ученое звание)

Дата: « ____ » _____ 20 ____ г

Подпись _____