**Курсовая работа**

**Задание**

1. Выполнить построение объектной модели предметной области.
2. Разработать диаграмму классов предметной области предложенного варианта.
3. Выполнить реализацию описанного задания с использованием языка программирования С++.
4. Оформить отчет, который должен содержать:
   1. титульная страница;
   2. название, цель, постановка задачи;
   3. описание функциональных требований к разрабатываемой системе;
   4. объектную декомпозицию (диаграмму объектов, классов);
   5. листинг программы;
   6. результат работы, на предложенных тестовых данных;
   7. вывод о проделанной работе;
   8. список литературы.

**Критерии оценки:**

Оценку “отлично” заслуживает студент, который выполнил и сдал работу в срок предусмотренным планом проведения дисциплины, а также применивший при проектировании системы своего задания два и более паттерна проектирования, один из которых должен быть порождающий. Количество объектов не менее 10.

Оценку “хорошо” заслуживает студент, который не успел выполнить задание в срок, но выполнил остальные условия, или применил меньше 2-ух паттернов проектирования. Количество объектов не менее 8.

Оценку “удовлетворительно” заслуживает студент, который не успел выполнить задание в срок, но выполнил остальные условия. Количество объектов не менее 7.

Приветствуется использование графических компонент для реализации предложенного задания.

**Варианты задания**

1. Система учета успеваемости студентов в ВУЗе. Программа должна учитывать результаты выполнения студентами различных видов учебной нагрузки (лабораторные работы, РГЗ, курсовые проекты, зачеты, экзамены). Студенты принадлежат различным группам, факультетам, институтам.
2. Редактор UML-диаграмм. Программа должна обеспечить возможность построения UML-диаграмм различных типов. Пользователю должна быть предоставлена возможность сохранения/открытия диаграммы из файла.
3. Программа построения графиков функций. Программа должна строить графики функций различного вида: многочлены, тригонометрические функции и т.п. Графики функций выводятся в специальном поле вывода. Пользователь должен иметь возможность изменять видимый диапазон по осям абсцисс и ординат, изменять коэффициенты отображаемых функций, очищать поле вывода.
4. Программа моделирования движения транспорта. В специальном поле вывода отображается структура улиц населенного пункта. Транспортные средства (автомобили, автобусы и т.п.) представляются условными изображениями. Каждое транспортное средство имеет свой маршрут и скорость. Транспортные средства перемещаются по улицам населенного пункта с учетом своей скорости и модельного времени. Пользователь должен иметь возможность изменять скорость модельного времени, добавлять и удалять транспортные средства, изменять маршрут транспортных средств.
5. Программа для игры в морской бой. Программа должна обеспечивать возможность игры человека с компьютером. На экране отображаются два игровых поля: поле для расстановки кораблей человеком и поле для отметки наносимых ударов по кораблям противника. Должна быть предусмотрена возможность использования кораблей различного типа: одно-, двух-, трех- и четырехпалубных. Пользователь в режиме расстановки кораблей должен иметь возможность размещения кораблей на игровом поле, перемещения, поворота и удаления кораблей. В режиме игры пользователь в наглядном виде должен информироваться о том, достиг ли цели его выстрел и выстрел противника.
6. Редактор векторных изображений. Программа должна обеспечивать возможность создания векторных изображений с использованием простейших примитивов (линия, прямоугольник, эллипс), заливки произвольных замкнутых областей, масштабирования, изменения цвета, перемещения, сохранения во внешний файл в некотором формате и загрузки изображения из внешнего файла.
7. Программа для построения генеалогического дерева. Программа должна обеспечивать возможность ввода информации о людях, состоящих в родстве друг с другом (пол, дата рождения, место рождения, основной вид деятельности и т.п.). Для каждого человека должна быть возможность указания его родителей, детей. Для регистрации изменений в программе должны быть предусмотрены события различного типа: рождение, смерть, женитьба и т.п. По данным о родственных связях программа должна обеспечить вывода на экран генеалогического дерева.
8. Программа моделирования спортивной игры (футбол). Программа должна обеспечить возможность учета игроков различных команд, моделирования матчей команд, учета результатов матчей в различных турнирах. Каждая команда имеет отличительные свойства (например, настрой, умение нападать, умение защищаться), которые влияют на результаты матчей с участием этой команды. Каждый игрок характеризуются некоторыми атрибутами, которые влияют на его поведение во время матча (например, скорость, точность удара, выносливость и т.п.).
9. Программа моделирования спортивной игры (волейбол). Программа должна обеспечить возможность учета игроков различных команд, моделирования матчей команд, учета результатов матчей в различных турнирах. Каждая команда имеет отличительные свойства (например, настрой, умение нападать, умение защищаться), которые влияют на результаты матчей с участием этой команды. Каждый игрок характеризуются некоторыми атрибутами, которые влияют на его поведение во время матча (например, скорость, точность удара, выносливость и т.п.).
10. Программа моделирования спортивной игры (баскетбол). Программа должна обеспечить возможность учета игроков различных команд, моделирования матчей команд, учета результатов матчей в различных турнирах. Каждая команда имеет отличительные свойства (например, настрой, умение нападать, умение защищаться), которые влияют на результаты матчей с участием этой команды. Каждый игрок характеризуются некоторыми атрибутами, которые влияют на его поведение во время матча (например, скорость, точность удара, выносливость и т.п.).
11. Программа для игры в шахматы с ограниченным набором фигур. В специальном поле на экране отображается шахматная доска. На доске выводятся шахматные фигуры. Программа должна обеспечивать три режима игры: компьютер с компьютером, человек с компьютером, человек с человеком.
12. Пошаговая стратегия поле 10 на 10. Игра с компьютером, или игра один на один. Для реализации использовать не менее трех разных типов юнитов, отличающихся между собой характеристиками и поведением. Для отрисовки можно использовать openGL.
13. Каждый блог принадлежит одному пользователю и состоит из нескольких записей. В момент создания блога в нем записей нет. Пользователь может создавать записи лишь в собственном блоге. Читать записи любого пользователя может каждый пользователь. В каждой записи есть заголовок, дата, текст. На каждой странице блога находятся ссылка на профиль. На начальной странице блога отображаются 10 последних записей (или менее, если в блоге их недостаточно). Если записей в блоге больше 10, то с начальной страницы можно перейти на вторую, где отображается второй десяток записей и т. д. Записи в блоге упорядочены по убыванию даты. Любая запись может быть отредактирована, но дата записи не может быть изменена. Запись может быть удалена автором. Перейдя на страницу отдельной записи, пользователь может ее комментировать, независимо от того, чужая это запись или его собственная. Комментарий состоит из текста и даты, указывающей, когда он написан. Комментарий может быть дан к записи блога или к ранее написанному комментарию. Пользователь может запретить комментирование любой записи в своем блоге. Комментарии, полученные раньше запрета, сохраняются. Запрет на комментирование может быть снят. Все комментарии к записи отображаются под ее текстом в виде дерева. Каждый комментарий помечен именем его автора. Имя может быть использовано для перехода к начальной странице блога с этим именем. Комментарии не могут быть отредактированы, но могут быть удалены автором записи. При удалении комментария также стираются все ответы на него, если таковые были. Любой желающий вести блог должен зарегистрироваться. При регистрации пользователь заполняет форму, указывая желаемое имя, пароль, список своих интересов, дату рождения, контакты, краткие сведения о себе. Если имя не занято, и форма заполнена полностью, система заводит блог нового пользователя. Сведения из регистрационной формы (кроме пароля) отображаются на странице профиля пользователя. Пользователь может отредактировать собственный профиль, изменив в нем все, кроме имени. Пользователь может изменить свой пароль. Пароль не может быть короче 8 символов. Зарегистрированный пользователь, осуществляет вход в собственный блог, указывая имя и пароль. Все описанные выше действия с записями, комментариями и собственным профилем, действия с друзьями доступны пользователю только после входа. Пользователь может полностью удалить свой блог со всеми записями, комментариями и профилем. После удаления блога регистрационные данные стираются из системы.
14. Система обеспечивает составление расписания некоторого учебного заведения, внесение в него изменений, выдачу полного расписания и дополнительной информации (например, по итоговому расписанию составляется расписание указанной группы или указанного преподавателя на заданный день или неделю). В расписании фиксируются время и место проведения занятия, предмет и преподаватель, проводящий занятие, а также номер группы, для которой это занятие проводится. Некоторые занятия проводятся для нескольких групп: потока или всего курса. Расписание не должно содержать коллизий (например, разные занятия не должны пересекаться друг с другом по месту и времени их проведения, один преподаватель не может вести одновременно два разных занятия, в одно и то же время, у одной и той же группы не допускается одновременно два различных занятия и т. д.). Система по требованию пользователя должна осуществлять поиск свободных аудиторий на заданную дату и время.
15. Домофон регулирует доступ в подъезд многоквартирного дома. В подъезде имеется дверь с замком. С наружной стороны двери установлена панель с кнопками на каждую квартиру, микрофон и динамик. В каждой квартире имеется кнопка «СВЯЗЬ», «БЛОКИРОВКА» и «ОТКРЫТЬ». Кроме того, в квартире имеется микрофон и динамик. Жильцы могут открывать дверь ключом. Посетитель может нажать кнопку квартиры. В квартире раздается звонок. При нажатии кнопки «СВЯЗЬ» внутри квартиры устанавливается звуковое сообщение между квартирой и посетителем. Звуки, произносимые посетителем в микрофон, установленный в подъезде, должны быть слышны в динамике, установленном в квартире. При нажатии на кнопку «ОТКРЫТЬ» дверь в подъезд открывается. Нажатие на кнопку «БЛОКИРОВКА» отключает подачу звонка в квартиру при нажатии на кнопку квартиры в подъезде. Повторное нажатие на кнопку «БЛОКИРОВКА» вновь включает подачу звонка. Сведения о посещении дома (когда и из какой квартиры открывали замок домофона) сохраняются в журнале домофона. По прошествии суток журнал домофона пересылается на пульт начальника охраны. После пересылки данных журнал обнуляется.
16. Пошаговая игра в стиле roguelike– игрок перемещается по случайно cгенерированному уровню, содержащему случайно сгенерированных противников, пытаясь найти выход на следующий уровень. У игрока есть базовые атрибуты (здоровье, выносливость, и пр.) и инвентарь, в который он может поместить для дальнейшего использования найденные предметы. Игра заканчивается, если здоровье игрока достигнет 0. Противники не двигаются по уровню, а генерируются группой в случайном месте (от 1 до 3 противников в группе). Противники, как и игрок, обладают некоторыми атрибутами. Они могут атаковать игрока, некоторые также могут применять атакующие или защитные способности, например, лечить союзников. При столкновении с игроком начинается пошаговый бой – сперва ход игрока, затем ходы противников. Игрок может зарабатывать опыт (путем убийства противников), и повышать уровень персонажа, увеличивая свои атрибуты и получая новые способности, которые могут быть применены в бою. Предметы расположены по уровню в виде случайно сгенерированных сундуков. Сундук может содержать какую-либо вещь (которую игрок может взять или выбросить) или не содержать ее. На каждом новом уровне генерируются все более сильные противники, а персонажа постоянно терзает голод, поэтому недостаточно просто найти выход на следующий уровень – нужно исследовать уровень в поисках еды и экипировки, а также заработать достаточное количество опыта, так как у низкоуровневого персонажа нет шансов в бою с высокоуровневым противником.
17. Система – «Управления задачами». В системе требуется реализовать персонализированный доступ для пользователя. Каждому пользователю доступны N досок с тематическими M колонками и T задачами на них. Каждой задачи можно добавлять атрибут. Атрибуты составляются вручную пользователем. Рассмотреть возможность добавления логики уведомлений, таймеров, дедлайнов и других стандартных атрибут для задачи. К каждой задачи можно добавлять исполнителей.
18. Карточные игры
19. Клеточные игры