

Для каждой задачи необходимо реализовать объектную модель. В реализации задач должны присутствовать наследование, агрегация и др. взаимодействия классов. Общение с пользователем осуществляется через консоль путем вызова наиболее значимых методов классов, позволяющих увидеть основную функциональность задачи. Для моделирования динамики процессов можно использовать повторяющиеся вызовы пользователем.

*Литература: Страуструп, главы 5-6 (Классы)*

1. Нефтяное месторождение. Месторождению принадлежат скважины, скважины бывают трех видов - нефтяные, газовые, нагнетательные (для закачки воды в пласт). Месторождение оперирует параметрами: объем газа в месторождении, объем нефти, объем воды. У каждой скважины есть параметр – объем выкаченного или закаченного содержимого. Задача заключается в моделировании работы месторождения. Предполагается интерфейс для задания параметров месторождения, добавления/удаления скважин, задания параметром каждой скважины, а также включения/выключения ее работы.
2. Банковская система. Можно создать несколько банков. В каждом банке может быть несколько клиентов. Клиенты банка могут быть физическими и юридическими лицами. У каждого клиента банка существует счет (только один). У банка также существует аналогичный параметр – сумма собственных средств банка. Клиенты банка могут проводить операции:
  - положить деньги на счет;
  - снять деньги со счета;
  - сделать перевод суммы денег на другой счет (возможно в другом банке, только для юридических лиц).При переводе денег со счета на счет с суммы перевода в пользу банка взимается определенный процент. Необходимо предусмотреть интерфейс для создания/удаления банков, создания/удаления клиентов каждого банка, а также выполнения каждым клиентом перечисленных операций. При этом не должны допускаться операции, в результате которых счет клиента становится отрицательным.
3. Маршрутное такси. Маршрутное такси движется по маршруту, который представляет собой остановки, соединенные прямыми линиями. Такси движется от начальной остановки к конечной и обратно и так множество раз. Такси имеет определенную вместимость. На остановках находятся пассажиры (экземпляры классов), которым надо попасть на определенную остановку. Такси останавливается на остановках, если есть свободные места или пассажиры, которым на этой остановке надо выходить и которые, соответственно, выходят. Пассажиры, которые стоят на остановках, садятся в такси, если оно движется в нужном направлении, в порядке очередности их появления на остановке. Задача заключается в моделировании движения такси. Необходимо предусмотреть возможность построения маршрута и добавления/удаления пассажиров на каждую остановку с заданием целевой остановки для каждого пассажира.
4. Файловая система. Необходимо смоделировать файловую систему ОС Unix (без прав доступа). Файловая система состоит из папок, файлов и ссылок. Ссылки бывают 2-х видов жесткие и символьные (при удалении жесткой ссылки удаляется заодно и сам файл/папка, на который эта ссылка ссылается, точнее оба имени являются абсолютно равноправными). Файлы имеют размер, расширение, кроме того файлы и

папки имеют атрибуты: время создания, время изменения и т.п. Файловая система имеет только один корень, представляющий собой папку, которую нельзя удалить. Для каждого элемента файловой системы предусмотреть операции создания, переименования, перемещения, изменения (меняется дата изменения) и удаления.

5. Иерархия классов. Необходимо создать классы для описания элементов языка C#: пространств имен, классов, полей, методов и свойств классов и т.д. Классы принадлежат пространствам имен, в свою очередь каждое пространство имен знает список своих классов. Классы могут наследоваться друг от друга. Для классов необходимо предусмотреть операцию проверки корректности класса (например, если класс наследуется от абстрактного класса, то он должен перегрузить все абстрактные методы и свойства и т.д.). Интерфейс задачи должен представлять собой навигацию по дереву. Любую сущность можно создать/изменить/удалить. Предусмотреть операцию построения кода шаблона классов (с пустыми методами и свойствами).
6. Иерархия графических компонентов. Имеется набор компонент: окно, надпись, список, кнопка, панель, линия. Компоненты окно и панель могут содержать другие компоненты. Компоненты могут быть видимы или скрыты, компоненты можно перемещать в пределах контейнера. Отобразить иерархию компонентов и навигацию по ней в виде дерева.
7. Муравейник. Типы объектов: матка, солдаты, полицейские, рабочие, личинки, вредители. Каждый предстатель является отдельным экземпляром класса. Муравейник имеет склад пищи фиксированного размера. Матка создает личинки, из личинок выводятся солдаты, полицейские или рабочие муравьи и ест очень много, солдаты уменьшают число вредителей и много едят, полицейские помогают увеличить число добываемой пищи и едят меньше солдат, рабочие добывают пищу и едят меньше полицейских, личинки только едят. Матка имеет следующие параметры:
  - кол-во потребляемой пищи;
  - кол-во рождаемых личинок.Солдаты имеют следующие параметры
  - кол-во потребляемой пищи;
  - кол-во уничтожаемых вредителей.Полицейские имеют следующие параметры:
  - кол-во потребляемой пищи;
  - процент, на который увеличивается запас пищи.Рабочие имеют следующие параметры:
  - кол-во потребляемой пищи;
  - кол-во добываемой пищи.Личинки имеют только один параметр: кол-во потребляемой пищи. Через некоторое время личинки должны разродиться в новую форму. При изменении формы личинки должны об этом сигнализировать матку, чтобы она определила тип, в который превратится личинка. Показать работу муравейника для различных значений числа муравьев, вредителей, пищи.
8. Дачная война. В задаче участвуют объекты: дачник, кроты (мужского и женского пола). На игровом поле располагаются грядки, на которых выращивается урожай. Дачник перемещается по полю к ближайшему кроту, который находится на поверхности, и имеет параметры: скорость перемещения и радиус удара. Кроты

хаотично перемещаются по полю под землей и хаотично вылезают на поверхность. Если крот вылез на поверхность, то он находится на ней некоторое время и уничтожает какое-то кол-во урожая. При движении под землей крот не видит движения дачника. Если кроты мужского и женского пола встретятся под землей, то они замирают и через некоторое время у них появляется еще один крот. Перед запуском задаются параметры дачника, размер игрового поля, количество и тип кротов. Игра считается оконченной, если будет уничтожен весь урожай, или погибнут все кроты.

9. Иерархия процессов операционной системы. Объекты: процессы, нити, динамические библиотеки. Реализовать поиск, создание, удаление, вычисление занимаемой памяти для процессов, нитей, библиотек. Процессы, нити и библиотеки могут зависеть друг от друга. Если удаляется объект, то удаляются все зависимые от него объекты.
10. Торговая организация. Организации принадлежат склады с товарами и магазины. Существует два вида клиентов - оптовые и розничные, соответственно, оптовые приобретают товар на складе, розничные в магазине. Количество товара на складах и в магазинах ограничено, товары бывают различных типов, торговая организация получает товары от поставщика, партии товаров разной величины. Учесть распределение товаров по грузовикам для доставки клиентам магазинам.
11. Молекула вещества. Молекула атомы нуклоны (протон, нейтрон), электроны. Между атомами существуют связи различных энергий, связь реализуется посредством взаимодействия электронных оболочек атомов. Реализовать операции: определение суммарной энергии взаимодействия атомов, массы молекулы, суммарной массы нуклонов и электронов, числа атомов одного типа в молекуле.
12. Персональный компьютер. Реализуемые объекты: процессор, память, системная плата, устройства ввода вывода. Реализовать процедуры ввода/вывода с учетом возможностей оборудования.
13. Менеджер контроля версий приложения. Приложение состоит из набора файлов, каждый файл может иметь нуль и более версий, версии одного файла образуют линейную структуру, каждая версия имеет номер и дату создания, версия может находиться в состояниях <зафиксировано> и <редактируется>. Реализовать функции построения конфигурации приложения по дате версий, по номеру версий, по состоянию версий.