Московский государственный университет
имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Е. И. Большакова

Задания практикума по объектно-ориентированному программированию

Учебно-методическое пособие

Москва
2011

УДК
ББК

Рецензенты: доцент, к.ф.-м.н. Е.А. Кузьменкова
доцент, к.ф.-м.н. В.В. Малышко

Большакова Елена Игоревна

Задания практикума по объектно-ориентированному программированию: Учебно-методическое пособие. – М.: Издательский отдел факультета ВМК МГУ (лицензия ИД № 05899 от 24.09.2001), 2010 – 48 с.

В пособии описываются постановка задачи и варианты задания практикума по имитационному моделированию процессов и явлений на основе объектно-ориентированного программирования. Приводятся методические указания и пояснения. Пособие предназначено для поддержки практикума по программированию для студентов 3 и 4 курсов факультета ВМК МГУ.

Печатается по решению Редакционно-издательского совета факультета вычислительной математики и кибернетики МГУ им. М. В. Ломоносова

ISBN 978-5-89407-440-5

СОДЕРЖАНИЕ

Предисловие

Парадигма объектно-ориентированного программирования является одной из ведущих в современной программной индустрии и одновременно одной из самых сложных в практическом освоении. Описываемые в данном учебном пособии задания практикума относятся в основном к области *имитационного моделирования*, которая охватывает широкий круг процессов и явлений (физических, производственных, управленческих и т.п.) и для которой очень естественно применение методологии объектно-ориентированного программирования. В ходе объектного анализа таких задач обычно несложно выявить ключевые понятия проблемной области, образующие в ходе дальнейшего проектирования программные объекты и классы имитационной модели. Именно поэтому для семестрового студенческого практикума по объектно-ориентированному программированию в течение нескольких лет специально подбирались разнообразные задачи имитационного моделирования.

Предлагаемые в пособии задания допускают различные уточнения (как усложняющие, так и упрощающие их выполнение), и в результате для одного задания возможно несколько разных, но удачных способов проектирования и построения программных систем. Все задания не зависят от применяемого языка объектно-ориентированного программирования.

Автор признательна Н.В. Баевой за ценные предложения по улучшению вариантов заданий, а также благодарит всех своих коллег, помогавших на разных стадиях подготовки пособия.

1. Общее описание заданий

Постановка задачи

Разработать *программную систему*, осуществляющую имитационное моделирование процесса или явления (определяемого вариантом задания) и визуализирующую этот процесс или явление.

Использовать для создания системы один из объектно-ориентированных языков программирования: C#, Object Pascal, C++, Java, Python, PHP, Ruby, Visual Basic и др., а также поддерживающие его инструментальные средства.

Провести с помощью разработанной системы исследование поведения моделируемого процесса, задавая для этого различные значения параметров, от которых зависит этот процесс.

Основные требования

- Система должна быть спроектирована на основе методологии объектно-ориентированного программирования, т.е. должна быть представлена в виде совокупности взаимодействующих друг с другом *объектов*, причем каждый объект является экземпляром определенного *класса*, а классы образуют иерархию. В ходе объектно-ориентированного проектирования необходимо определить и зафиксировать *логическую* структуру (классы и объекты) и *файловую* (модульную) структуру системы.

- Система должна предоставлять удобный и понятный *пользовательский интерфейс*, предусматривающий проведение *экспериментов по моделированию* и выдачу в ходе экспериментов необходимой информации (определяемой вариантом задания).
- Для проведения экспериментов по моделированию перед началом каждого эксперимента пользователь должен иметь возможность устанавливать нужные значения параметров, от которых зависит этот процесс или явление. Такие параметры называются *параметрами моделирования*, обычно в их числе – *шаг моделирования*, т.е. отрезок времени, измеряемый в тех или иных единицах времени (секундах, минутах, часах, днях, неделях и пр.) и/или число шагов моделирования.
- Поскольку в большинстве вариантов задания моделируемый процесс или явление зависит от нескольких *неопределенных факторов*, следует моделировать такие факторы статистически – на основе одного из законов вероятностного распределения (равномерного, нормального и др.).

Содержание работы

1) Выбор и изучение инструментальных средств: языка программирования, соответствующей *интегрированной среды разработки* приложений (Visual Studio, Eclips, Delphi, C++ Builder и т.п.), графических библиотек.
2) Общее проектирование системы: уточнение постановки задачи выбранного варианта задания, определение изменяемых *параметров* моделируемого процесса/явления, *метода моделирования*, средств и *объектов визуализации*; составление *эскиза пользовательского интерфейса.*
3) Объектно-ориентированное проектирование: объектный анализ решаемой задачи и разработка *диаграмм*, характеризующих соответственно классы и объекты системы, выделенные в ходе анализа; составление *текстовых спецификаций интерфейса классов.*
4) Программирование системы на основе всех проектных решений, определение *файловой (модульной)* структуры программы.
5) Проведение исследования (экспериментов) по моделированию на базе реализованной программной системы.
6) Составление отчета, в который включаются:
   - Уточненная постановка задачи для выбранного варианта задания.
   - *Диаграмма классов* программной системы.
   - *Текстовые спецификации* основных классов системы.
   - *Диаграмма объектов* программной системы.
   - Указание использованных при выполнении задания *инструментальных средств* (языка программирования, интегрированной среды, библиотек).
   - Описание *файловой структуры* программной системы.
   - Краткая характеристика *пользовательского интерфейса*.
   - Краткое описание проведенных экспериментов.

2. Варианты

Демонстрационные модели физических явлений

Компьютерная модель Солнечной системы

Необходимо изобразить на экране компьютера Солнце и восемь *планет* Солнечной системы (от Меркурия до Нептуна), а также основные их *спутники* (например, для Земли – Луну, для Марса – Фобос и Деймос), в их *движении по орбитам*. Можно считать, что вращение планет вокруг Солнца происходит в одной плоскости (поскольку плоскости орбит планет близки к плоскости земной орбиты), к этой плоскости можно отнести и орбиты спутников планет. Вращение планет вокруг своей оси можно не учитывать. Требуется также смоделировать и показать одно из астрономических событий в Солнечной системе: пролет *кометы* через Солнечную систему, запуск с какой-либо планеты искусственного спутника, перелет *ракеты* с планеты на планету или другое.

Цель моделирования – наблюдение различных конфигураций планет и их спутников, а также выбранного астрономического события. В изменяемые параметры моделирования следует включить массу и начальную скорость кометы, космического корабля (ракеты) или искусственного спутника.

Моделирование движения планет может базироваться на трех законах Кеплера, определяющих эллиптические орбиты планет и спутников, количественные изменения скорости движения планеты по орбите и связь средних расстояний планет от Солнца с периодами их звездных обращений. Моделирование может осуществляться и на основе закона всемирного тяготения Ньютона – это более точный способ, поскольку позволяет учесть притяжения планет друг к другу, а не только к Солнцу. При моделировании на основе закона Ньютона для расчета местоположения планет потребуется численно решать систему дифференциальных уравнений второго порядка – для этого могут быть применен метод Рунге-Кутта или другой численный метод.

При моделировании следует учесть, что траектория движения спутника складывается из двух вращательных движений: вращения спутника вокруг своей планеты и вращения планеты вокруг Солнца. Необходимо учесть также зависимость вида траектории кометы или ракеты (эллипс, парабола или гипербола) от величины их начальной скорости.

При визуализации Солнечной системы на экране компьютера должно быть соблюдено правильное соотношение размеров орбит планет и скоростей их движения (т.е. они должны быть пропорциональны их реальным значениям). Соотношение размеров изображений самих планет также должно соответствовать действительности, но при этом для наглядности масштаб их изображения должен быть больше масштаба показа их орбит, иначе некоторые планеты отображаются на экране точками. Кроме того, поскольку при показе на экране сразу всех восьми планет Солнечной системы изображения ближайших к Солнцу планет (и их спутников) получаются слишком мелкими и сливаются друг с другом, следует либо использовать крупный масштаб и предоставить скроллинг изображения, либо предусмотреть визуализацию Солнечной системы

в двух масштабах (для всех планет и для ближайших к Солнцу). Пользователь должен иметь возможность включать и отключать показ названий планет и спутников, их орбит и траекторий движения других тел, а также ускорять или замедлять движение тел в Солнечной системе.

Система разработки оптических конструкций из линз

Разрабатываемая конструкция состоит из N $(1 \le N \le 5)$ различных тонких линз, расположенных вдоль одной горизонтальной *оптической оси* (на этой оси расположены центры линз). Линзы могут быть двух основных типов – *рассеивающие* и *собирающие* линзы, для каждого типа возможны 3 вида линз, а именно: для первого типа – плосковыпуклая линза, двояковыпуклая линза, собирающий мениск; для второго типа – плосковогнутая линза, двояковогнутая линза, рассеивающий мениск [1].

Для задания *оптической конструкции* (конфигурации линз) необходимо определить количество линз, их тип, вид и фокусное расстояние, а также указать их местоположение на оптической оси. Основная функция программной системы – проведение *оптического эксперимента* для заданной конструкции, при котором определяются и показываются пути лучей света от некоторого исходного объекта через эту конфигурацию линз, а также показывается сформированное каждой линзой *реальное* или *мнимое изображение* объекта.

Исходный объект имеет линейный *размер* и *ориентацию* (по вертикали) и изображается обычно стрелкой, направленной вверх или вниз. Для формируемого изображения этого объекта должны быть определены и показаны линейный размер, место расположения на оптической оси и ориентация. Система должна различать реальные и мнимые изображения, используя при их визуализации, например, разные цвета.

Если конфигурация состоит только из одной тонкой линзы, и исходный объект не находится ни в бесконечности, ни в *точке фокуса* этой линзы, то линза сформирует изображение, расстояние L_i которого от центра линзы определяется из равенства $1 / L_0 + 1 / L_i = 1 / F$
где L_0 – расстояние от объекта до центра линзы, а F – фокусное расстояние линзы (фокусное расстояние собирающей линзы положительное, фокусное расстояние рассеивающей линзы – отрицательное). Получаемое изображение будет либо реальным (в случае собирающей линзы), либо мнимым (в случае рассеивающей), оно может располагаться в зависимости от величины L_0 и знака F или с той же стороны линзы, что и сам объект, или на противоположной стороне. Линейное увеличение D изображения, осуществляемое одиночной линзой, определяется по формуле $D = -L_i / L_0$.

Система разработки оптических конструкций должна допускать также случай, когда исходный объект находится в бесконечности и от него на линзу идут параллельные лучи света (при этом в зависимости от типа линзы лучи либо рассеиваются, либо собираются в точке фокуса), и случай, когда объект расположен в точке фокуса (тогда из линзы выходят параллельные лучи света).

Точка, в которой формируется реальное или мнимое изображение, определяется пересечением трех *главных лучей*, которые получаются так:

1. Луч, параллельный оптической оси, после преломления в линзе, проходит через точку фокуса собирающей линзы или является продолжением луча, выходящего из точки фокуса рассеивающей линзы;
2. Луч, проходящий через центр тонкой линзы, не отклоняется;
3. Луч, проходящий через точку фокуса или идущий по направлению к ней, выходит параллельным оптической оси.

В общем случае в конструкцию входит более одной линзы, и реальное изображение, сформированное одной линзой, служит исходным объектом для следующей линзы в ряду линз вдоль оптической оси.

Пользователь системы должен иметь возможность:

- определять и изменять все параметры оптического эксперимента, включая виды и фокусные расстояния линз, их позицию на оптической оси, размеры и ориентацию исходного объекта;
- запоминать копию оптического эксперимента в файле, сохраняя все его параметры, и считывать эту копию из файла в рабочее окно;
- включать и выключать изображения *масштабной шкалы* (линейки) вдоль оптической оси, путей лучей через линзы и сформированных изображений объекта.

Требуется, чтобы указанные действия пользователь мог производить в произвольном, удобном для него порядке, и изменение одного параметра эксперимента не должно затрагивать все остальные установленные параметры.

Модель оптических экспериментов в зеркальной комнате

Зеркальная комната представляет в плане произвольный замкнутый M-угольник ($4 \le M \le 9$), каждая стена – *плоское* или *сферическое* зеркало. Для проведения экспериментов необходимо определить для каждой стены комнаты вид зеркала (плоское или сферическое), а для каждого сферического зеркала – его тип (*вогнутое* или *выпуклое*) и *радиус кривизны*.

Основная функция программной системы – проведение *оптического эксперимента*, при котором из некоторой точки на одной из стен комнаты, под определенным углом к этой стене (угол может варьироваться от 0 до 180 градусов) выпускается *луч света*, и затем показывается его путь внутри комнаты с учетом отражений от зеркал. Траектория луча определяется физическими законами отражения от зеркальных поверхностей.

Цель моделирования – подбор пользователем системы параметров зеркал и исходного угла выпущенного луча, при которых луч, отражаясь от зеркальных стен, попадает в нужную точку (зону) комнаты.

При визуализации оптического эксперимента должен быть показан план комнаты и изображен путь луча в комнате.

Пользователь системы должен иметь возможность:

- определять число M стен комнаты и рисовать ее план (например, указывая мышью на экране компьютера угловые точки комнаты);
- задавать и изменять параметры зеркал (вид, тип, радиус кривизны), точку выпускания луча и его исходный угол;

- запоминать в файле копию оптического эксперимента, сохраняя все его параметры, и считывать сохраненную копию из файла в рабочее окно.

Требуется, чтобы указанные действия пользователь мог производить в произвольном, удобном для него порядке, и изменение одного параметра эксперимента не должно затрагивать другие установленные параметры.

Возможно усложнение рассматриваемой задачи, когда при отражении от зеркальной поверхности учитывается эффект *рассеивания света* – в этом случае после нескольких отражений луч становится невидимым. При этом в число параметров эксперимента входят коэффициенты рассеивания каждого зеркала.

Система конструирования и расчета электрических схем

Рассматривается *замкнутая электрическая цепь*, по которой течет постоянный ток. В цепь включено *N (1≥N≥12) элементов* нескольких видов: электрическая лампочка, резистор, электронагревательный прибор, источник постоянного тока (э.д.с.), выключатель (замыкающий ключ). В цепь могут входить две-три подцепи, а значит, в ней есть точки разветвления тока.

Основная функция программной системы – расчет для заданной электрической схемы основных ее характеристик, т.е. величины *тока* в каждой ее точке и величины *напряжения* между двумя произвольными точками. Вычисление характеристик цепи производится на основе законов Ома (для замкнутой цепи и для участка цепи).

Цель моделирования – конструирование и исследование пользователем системы нужной электрической схемы, например, схемы *электрического реле,* состоящего из основной цепи и замыкающей (размыкающей, переключающей) подцепи. Пользователь системы должен иметь возможность:

- Задать *контур электрической цепи*, либо выбрав его из нескольких возможных (зафиксированных в системе), либо определив мышью основные ее точки (включая точки разветвления тока);
- Расположить в нужных точках заданного контура необходимые элементы и задать их внутренние характеристики (включая величину внутреннего сопротивления);
- Изменять расположение отдельных элементов заданной цепи и их характеристики, а также замыкать и размыкать входящие в цепь выключатели (замыкающие ключи);
- Узнавать величины тока, напряжения и сопротивления в определенных точках/участках построенной схемы;
- Запоминать построенную схему в файле и считывать ее из файла в рабочее окно.

Необходимо, чтобы указанные действия пользователь мог производить в произвольном, удобном для него порядке, и изменение характеристик и расположения отдельных элементов построенной схемы не должно затрагивать уже установленные характеристики других элементов.

Визуализация электрической схемы должна включать изображение цепи и всех входящих в нее элементов, а также показ текущего состояния

выключателей (замыкающих ключей) и горящих лампочек, сигнализирующих о наличии тока на соответствующем участке цепи.

Моделирование движения транспорта

Моделирование движения на автостраде

Рассматривается движение *автомобилей* на прямом однорядном (однополосном) *участке автострады* без перекрестков, в течение некоторого времени. Автомобили появляются на одном конце дороги и проезжают по ней до другого конца, стараясь по возможности сохранить начальную (заданную при их появлении) скорость. Автомобили могут иметь разную *начальную скорость*: начальная скорость – случайная величина, изменяющаяся в заданном диапазоне (например, от 50 до 100 км/час). Интервалы между появлениями автомобилей на дороге также являются случайными величинами из определенного интервала (например, от 1 до 5 секунд).

Считается, что минимальное допустимое сближение двух автомобилей составляет одну длину (корпус) автомобиля, в ином случае происходит авария. Когда автомобиль приближается к идущей впереди машине на утроенное допустимое расстояние, он начинает притормаживать по определенному закону, пока его скорость не сравняется со скоростью передней машины.

Пусть в таком потоке машин организована *искусственная кратковременная задержка* одного автомобиля: автомобиль сначала резко замедляется, сбрасывая за некоторое время скорость, а затем после некоторой паузы вновь набирает первоначальную скорость. В результате, если следующий автомобиль не успел притормозить, возникает авария. Может возникнуть и так называемая *пробка* – область с высокой плотностью автомобилей, включающая чередование притормаживаний и ускорений до прежней скорости. Действительно, если какой-то автомобиль начинает резко замедляться, идущий за ним автомобиль тоже через некоторое время тормозит. После торможения следует замедленное движение автомобиля, но как только дорога перед ним освобождается, автомобиль ускоряется до первоначальной скорости.

Пробка обычно возникает, если плотность потока автомобилей достаточно велика, и существует некоторое время, медленно двигаясь навстречу потоку автомобилей и постепенно рассеиваясь.

Заметим, что в аварию может попасть не тот автомобиль, что был искусственно задержан, а идущие за ним машины. В случае аварии должно пройти некоторое фиксированное время, прежде чем движение на этом участке вновь станет возможным, и после вынужденной остановки машины вновь набирают первоначальную скорость.

Необходимо разработать систему моделирования движения машин на автостраде, позволяющую наблюдать за возникновением и скоростью рассасывания возникающей пробки в зависимости от нескольких факторов-параметров. В число параметров моделирования следует включить: диапазон скоростей автомобилей, возможный интервал между их появлениями на дороге (от этого параметра зависит плотность потока), величину уменьшения скорости

искусственно притормаживаемой машины и время ее движения с меньшей скоростью.

При визуализации движения автомобилей по дороге следует учесть, что изображения самих автомобилей необходимо сделать крупнее, чем это определяется масштабом, иначе эти изображения будут слишком мелкими. Полезно использовать разные цвета для изображения различных состояний автомобиля (ускорение, торможение, движение с постоянной скоростью, авария). Интерфейс с программой моделирования движения должен обеспечивать один из способов указания автомобиля, который необходимо притормозить: например, нужный автомобиль отмечается мышью, или же задается отметка на дороге – тогда тормозится машина, пересекающая первой эту отметку.

Возможны усложнения задания, включающие:

- движение автомобилей в несколько *рядов* (полос);
- наличие на дороге одного или нескольких *светофоров*.

В первом случае потребуется определить правила перехода автомобилей с одного ряда на другой, во втором – задать законы работы светофоров (от них зависит скорость рассасывания возникающих заторов). Допускается вместо прямого участка дороги организовать автомобильный круг, поскольку при этом можно дольше наблюдать за рассасыванием пробок.

Моделирование движения на перекрестке дорог

На перекрестке двух автомобильных дорог расположены регулирующие движение *светофоры*. Каждая из дорог содержит несколько *полос* (рядов), автомобили двигаются в обоих направлениях. Светофоры обеспечивают проезд автомобилей по обеим дорогам, включая левый и правый повороты автомобилей, а также переход через эти дороги пешеходов.

Программа моделирования и визуализации движения на таком перекрестке служит для исследования характера возникающих на перекрестке автомобильных дорог *заторов* и их рассасывания в зависимости от плотностей потоков автомобилей и *режимов работы светофоров*.

Автомобили должны появляться на концах каждой из дорог случайным образом, проезжать по ним со скоростью, заданной при их появлении, притормаживая и останавливаясь при необходимости на перекрестке, и исчезая после проезда всей дороги на ее противоположном конце. У каждого автомобиля может быть своя *начальная скорость*, она определятся как случайная величина из некоторого диапазона (например, от 30 до 120 км/час). Случайной величиной является также интервал между появлениями автомобилей на каждой дороге – от диапазона изменения этой величины (и закона ее распределения) зависит плотность потока автомобилей. Как случайную величину, определяемую в момент появления автомобиля на дороге, следует моделировать и направление его проезда через перекресток (прямо / налево / направо).

Автомобили должны *перестраиваться* из одного ряда в другой и пересекать перекресток в соответствии с правилами дорожного движения. В частности, в левый ряд перед светофором становятся автомобили, которым необходим поворот налево. Кроме правил смены полосы, в программе должны

быть зафиксированы законы торможения и ускорения автомобилей на перекрестке, которые в общем случае зависят от допустимого сближения между автомобилями, величин их скорости и др. Возможность аварий (например, из-за нарушений правил дорожного движения) в модели можно не учитывать.

Цель проводимого моделирования – изучение различных режимов работы светофоров для поиска режима их оптимальной работы. Следует рассмотреть два типа режимов работы: статический, когда интервалы свечения каждого цвета (желтый, зеленый, красный) зафиксированы заранее, и динамический, при котором интервалы свечения изменяются в соответствии с количеством автомобилей (и пешеходов), ожидающих проезда (прохода) через дорогу.

В изменяемые параметры моделирования движения следует включить: тип режима работы светофора, интервалы свечения каждого цвета (для статического режима), дистанцию видимости светофора, диапазон возможных скоростей автомобилей, интервалы случайного появления автомобилей на каждой из дорог.

Визуальная картина движения на перекрестке дорог должна содержать изображения дорог, светофоров, движущихся машин. Полезно отобразить тем или иным образом (например, разными цветами) возможные направления движения автомобиля через перекресток (прямо/налево/направо). Желательно также предусмотреть вывод некоторых подсчитанных в ходе моделирования величин, к примеру, среднее время остановки автомобилей на перекрестке.

Модель движения на круговой автомобильной развязке

Рассматривается *круговая дорога*, соединяющая несколько (от трех до пяти) прямых автомобильных дорог и служащая для поворота автомобилей с одной прямой дороги на другую. Круговая дорога содержит *две полосы* (ряда), по которым автомобили двигаются в одном направлении (против часовой стрелки); а по входящим в круг дорогам допускается обычное двунаправленное движение, причем каждому направлению соответствует своя полоса дороги.

Требуется разработать компьютерную модель автомобильного движения на круговой развязке. Цель моделирования – исследование характера возникающих на ней заторов в зависимости от плотностей потоков автомобилей и их скоростей на каждой из входящих в развязку дорог.

Автомобили должны появляться на концах каждой из прямых дорог случайным образом, для этого интервал между последовательным появлением на дороге двух машин определяется как случайная величина. Для каждого нового автомобиля задается его *начальная скорость*, которая также является случайной величиной из некоторого диапазона (например, от 30 до 120 км/час). Случайным образом определяется и дорога, на которую нужно свернуть автомобилю при проезде через круговую развязку.

Автомобили стараются по возможности сохранить свою начальную скорость, но при необходимости они притормаживают и/или останавливаются перед другими автомобилями. Например, если автомобиль приближается к идущей впереди него машине на некоторое фиксированное расстояние (5-10 м), он начинает притормаживать, пока его скорость не сравняется со скоростью передней машины.

Когда автомобиль достигает развязки, он въезжает на круг и проезжает по нему до той дороги, на которую ему необходимо свернуть. Въезжать на круговую дорогу, двигаться по ней (перестраиваясь из одного ряда в другой), и выезжать с нее на нужную дорогу автомобили должны в соответствии с правилами дорожного движения. После выезда на нужную дорогу автомобиль проезжает по ней до конца и исчезает.

Кроме правил въезда/выезда и смены полос, в программе должны быть зафиксированы законы торможения и ускорения автомобилей, которые в общем случае зависят от допустимого сближения между автомобилями, дистанции видимости, видов автомобилей (например, легковых и грузовых).

В параметры моделирования необходимо включить диапазон возможных скоростей автомобилей и интервал между их появлениями на дороге (отдельно для каждой дороги, входящей в развязку, от этого зависит плотность потока автомашин на ней). Как параметр можно задавать и *приоритет дорог* развязки

Визуальная картина движения на круговой развязке должна включать изображения всех дорог и движущихся машин. Следует учесть, что изображения самих автомобилей необходимо сделать крупнее, чем это определяется масштабом, иначе они будут плохо видны. Полезно использовать разные цвета для изображения различных состояний автомобиля (ускорение, торможение, остановка, поворот).

Система контроля движения электропоездов

Рассматривается *линейный участок железной дороги*, соединяющий N станций $(7 \le N \le 20)$. Известно *суточное расписание* движения электропоездов между этими станциями (в одном направлении), которое включает M *маршрутов* $(5 \le M \le 20)$. Каждый маршрут фиксирует:

- станцию-пункт отправления и станцию-пункт назначения;
- промежуточные станции маршрута, в которых электропоезд делает остановку;
- время прибытия и отправки электропоезда в каждой станции маршрута.

Фактическое движение электропоездов зависит не только от расписания, но и от некоторых непредвиденных событий, к числу которых относятся задержки поездов на станциях маршрута, а также *аварии поездов* и *повреждения железнодорожных путей* (которые на некоторое время нарушают движение).

Требуется разработать систему контроля движения электропоездов, которая отслеживает их движение по маршрутам, регистрирует возникающие события и отклонения движения от расписания, а также корректирует при необходимости расписание, определяя время предполагаемого прибытия поездов на каждую станцию маршрута. Можно считать, что электропоезда двигаются по маршрутам с определенной скоростью (например, 70 км/час).

Цель моделирования – изучение стабильности движения поездов при заданном расписании движения в условиях возникающих непредвиденных событий. Период моделирования – один день.

Случайные факторы движения (задержки и аварии) следует моделировать статистически, определив для каждого фактора свой вероятностный закон распределения. Интервал времени между возникновением двух непредвиденных событий – случайная величина, изменяющаяся в некотором диапазоне

(например, для аварий – от 2 до 15 часов). Случайными величинами являются также длительность ремонта после аварии и время задержки при остановках электропоезда на станции.

Кроме величин N, M и расписания движения, в изменяемые параметры моделирования целесообразно включить: время суток начала моделирования движения электропоездов, шаг моделирования – 15 или 30 минут, диапазон изменения указанных случайных величин (от диапазона зависит частота возникновения непредвиденных событий).

В ходе моделирования на экране компьютера должна быть изображена схема рассматриваемого участка железной дороги, на которой показано движение поездов в соответствии с расписанием, возникающие аварии и неисправности путей. По запросу необходимо также показать расписание движения, скорректированное в соответствии с уже случившимися событиями. По окончании моделирования должен быть выведен *график* смоделированного движения поездов, подсчитанная средняя задержка на станциях, общее время нарушения движения вследствие аварии.

Полезной функцией создаваемой системы контроля является введение и расчет *дополнительного маршрута*. При этом система по заданному времени отправления поезда, заданным станциям отправления и назначения должна определить расписание движения по новому маршруту (по возможности, без изменений старых маршрутов). Время остановки электропоезда на промежуточных станциях должно быть в интервале от 2 до 5 минут.

Возможное усложнение задачи – рассмотреть движение электропоездов на рассматриваемом участке дороги в обоих направлениях (по каждому направлению действует свое расписание). При этом авария поезда нарушает движение только в одну сторону, а неисправность железнодорожного пути – движение в обоих направлениях.

Система управления движением на линии метро

Необходимо разработать систему, контролирующую движение на *линии метрополитена*, соединяющей N станций $(7 \le N \le 20)$. На двух конечных станциях линии расположены депо, в которых стоят свободные поезда. Движение по линии осуществляется в обе стороны, известно стандартное время перегона между каждыми двумя соседними станциями. Определен также *график движения* поездов метро, зависящий от времени суток и дня недели. График задает временной интервал между прибытием поездов на станцию и время остановки поезда на станции (например, в час пик интервал равен 1 минуте, а время стоянки – 2 минуты, в вечерние часы и в воскресные дни этот интервал движения увеличивается, время стоянки уменьшается). Время работы метро – с 6.00 утра до 12.00 ночи.

Фактическое движение поездов метро зависит не только от графика, но и от *непредвиденных задержек* поездов на станциях. В этом случае на следующем перегоне между станциями поезд двигается с большей скоростью (при этом стандартное время перегона можно сократить лишь в полтора раза) и сокращает время стоянки на следующей станции (но время стоянки не может быть менее 1 минуты), пытаясь тем самым «нагнать» время задержки и восстановить график

движения на линии. В общем случае для восстановления графика движения потребуется такое ускоренное движение поезда на нескольких перегонах между станциями (и сокращение времени его остановки на нескольких станциях), при этом следующие за ним поезда вынуждены удлинять свою остановку на станциях – чтобы сохранить минимальный интервал между прибытием поездов на станцию (равный 1 минуте), требуемый для безопасности движения.

Создаваемая система должна отслеживать движение поездов на линии, регистрировать (по обоим направлениям) возникающие на станциях задержки и соответствующим образом корректировать движение поездов. Для тестирования системы требуется смоделировать процесс движения поездов на рассматриваемой линии. Период моделирования – некоторый отрезок суток (например, с 6 до 12 часов). Цель моделирования – изучение стабильности движения поездов на линии.

При моделировании движения непредвиденные задержки поездов на станциях следует моделировать статистически: отклонение от времени отправления – случайная величина, изменяющаяся в некотором диапазоне (например, от 20 до 60 секунд), причем вероятность задержки зависит от времени суток и дня недели (в час пик она выше).

В число параметров моделирования следует включить величину N , шаг моделирования – отрезок времени от 30 до 120 секунд, период моделирования.

В ходе моделирования на экране компьютера должна быть изображена линия метро с названиями станций, показано движение поездов по обоим направлениям линии, указано – в виде табло на каждой станции – время, прошедшее после прибытия последнего поезда на станцию, и длительность его стоянки, а также время сдвига движения относительно исходного графика. По окончании моделирования следует предусмотреть вывод итоговой информации, в том числе количество и среднюю величину возникших задержек поездов, максимальный и средний интервал между прибытием поездов на станцию.

Система управления воздушным движением

Рассматривается работа *аэропорта* с N *взлетно-посадочными полосами* $(2 \leq N \leq 10)$. Необходимо создать систему-диспетчер, обрабатывающую *заявки на взлет и посадку* самолетов нескольких авиакомпаний, и провести эксперименты с ней, программно смоделировав поток заявок.

Заявки на взлет и посадку формируются на основе действующего *суточного расписания полетов* в данном аэропорту (оно включает расписание вылетов и расписание прилетов самолетов) с учетом возможных *отклонений от расписания* (из-за задержек загрузки топлива и по другим причинам). Отклонение от расписания моделируется как случайная величина, имеющая нормальное распределение в некотором интервале, например, от 0 до 120 минут. Для вылетов возможны только задержки, для посадок – как задержки, так и досрочные прилеты. *Фактическое время начала взлета или посадки* самолета определяется как время по расписанию, измененное на случайную величину отклонения, а также на время ожидания свободной полосы для взлета/посадки.

Длительность взлета/посадки зависит от типа самолета. Между последовательными взлетами/посадками самолетов на одной полосе обычно предусмотрены небольшие временные промежутки.

Цель моделирования функций диспетчера взлетно-посадочных полос – определение оптимального количества полос для безопасного обслуживания взлетов/посадок. Одним из безопасных режимов считается разделение всех полос на непересекающиеся множества – полосы для взлета и полосы для посадки. Период моделирования – сутки.

В параметры моделирования следует включить: число N взлетно-посадочных полос, исходное расписание полетов в данном аэропорту, диапазон разброса случайной величины отклонения от расписания, шаг моделирования (интервал времени от 5 до 30 минут), а также время суток, с которого начинается моделирование работы диспетчера (например, 7 часов утра).

Визуализация моделируемого процесса должна предусматривать показ состояния (занята/свободна) каждой взлетно-посадочной полосы и очереди самолетов на взлет и посадку, а также *график уже произведенных взлетов и посадок* с указанием времени и взлетной полосы. В ходе и по окончании моделирования следует вывести статистическую информацию: общее количество обслуженных заявок, максимальную и среднюю задержку вылета, максимальную и среднюю длину очереди на взлет или посадку, среднюю занятость взлетно-посадочных полос.

Как усложнение модели можно рассмотреть возможность задержки взлетов и посадок из-за погодных условий, аварий и терактов, длительность задержки также моделируется как случайная величина из некоторого диапазона.

Автоматизация производственных процессов

Система управления оптовым складом

Оптовый склад, на котором хранятся K видов $(12 \le K \le 20)$ *продуктовых товаров*, обслуживает M $(3 \le M \le 9)$ близлежащих *торговых точек* (мелких магазинов и палаток). Вместимость склада ограничена: каждого вида товара может храниться не более определенного количества *оптовых упаковок*. Все продукты имеют *срок годности*, и если этот срок истекает через несколько дней, то товар *уценивается*, чтобы продать его быстрее. После истечения срока годности продуктовый товар списывается и вывозится со склада.

Система управления складом хранит данные о наличии и количестве каждого продукта в текущий момент (например, 12 оптовых упаковок риса, в каждой 20 пачек по 1 кг), а также о сроке годности продукта и его стоимости. Система фиксирует поступающие в течение рабочего дня *заказы на доставку* товаров со склада в торговые точки, последовательно их обрабатывает и формирует список соответствующих перевозок на следующий рабочий день.

Каждый день от любой торговой точки на склад поступает не более одного заказа, который включает перечень заказываемых продуктов и их количество. Поскольку любой товар отпускается в оптовых упаковках, при выполнении заказа на определенный продукт система выделяет такое

количество оптовых упаковок, которое дает чуть большее или чуть меньшее количество товара (килограмм, пачек или др.) по сравнению с заказанным. Если заказанного товара нет в достаточном количестве, то он отпускается частично. Система управления складом отслеживает убыль товаров, и если какого-то вида товара становится меньше определенного количества, то составляется *заявка в фирму-поставщик* на доставку на склад нужного количества этого продукта.

Сформированный системой список перевозок заказанных товаров, список заявок в фирмы-поставщики и список продуктов, подлежащих уценке, может просматривать и утверждать пользователь системы (заведующим складом), при этом он может принимать решения о проценте уценки продукта.

Для тестирования заложенных в систему процедур автоматизации обработки заказов и составления заявок требуется смоделировать поток заказов, поступающих от торговых точек, а также поставку продуктов на склад фирмами-поставщиками. Период моделирования – N дней $(10 \le N \le 30)$, шаг моделирования – один день.

Поток поступающих заказов на продукты следует моделировать статистически: случайными величинами, изменяющимися в некоторых диапазонах, являются все составляющие каждого заказа, причем вероятность заказа уцененных продуктов выше, чем неуцененных, и зависит от процента уценки. Случайной величиной (от 1 до 5 дней) является также время выполняемой по заявке поставки продуктов на склад фирмой-поставщиком.

В параметры моделирования следует включить величины N, M, K, начальный набор продуктов на складе, а также диапазоны изменения вышеописанных случайных величин. В ходе моделирования должна быть доступна основная информация о работе склада: о наличии товаров, о заказах за текущий день и перевозках на следующий, о вывозе просроченных продуктов и денежных потерях склада за счет уценки продуктов и их списания др. По окончании моделирования целесообразно вывести некоторые статистические данные о работе склада за весь период моделирования, например, общий объем и стоимость проданных продуктов.

Моделирование службы доставки лекарств

Необходимо создать компьютерную модель работы *службы доставки*, действующей при складе *аптечных товаров*. На складе хранятся K различных *лекарств* $(15 \le K \le 35)$, которые можно заказать по телефону, факсу или электронной почте. Заказанные лекарства доставляются покупателям M *курьерами* службы $(3 \le M \le 9)$ в течение следующего дня. Все лекарства имеют *срок годности*, и если срок годности лекарства истекает через месяц, то лекарство *уценивается* в два раза. После истечения срока годности соответствующее лекарство списывается и подлежит вывозу со склада.

Компьютерная система хранит данные о наличии и количестве каждого лекарства в текущий момент (например, валидол – 137 штук), его дозировке (обычно в мг), виде (таблетки, суспензия, спрей, мазь и т.п.), сроке годности, оптовой цене, а также его группе (например, сердечно-сосудистое средство, антибиотик и т.п.). В системе хранится также информация о *постоянных покупателях* – их фамилия, телефон, адрес, номер дисконтной карты, список

регулярно закупаемых лекарств (например, инсулин) и периодичность такой закупки (например, еженедельно).

В течение каждого рабочего дня система фиксирует поступающие *заказы на лекарства* и автоматически формирует на этой основе *список доставки*. Заказ включает номер телефона и адрес покупателя, номер его дисконтной карты (если таковая имеется), а также перечень заказываемых лекарств (возможно, с уточнением дозировки и вида лекарства) с указанием нужного количества упаковок. Если требуемое в заказе лекарство имеется в нужном количестве на складе, то оно включается в соответствующую покупку, в ином случае выделяется только доступное на складе количество.

Общая стоимость покупки (выполненного заказа) подсчитывается как сумма стоимостей всех включенных в нее лекарств, с учетом действующей *розничной наценки* на лекарства (например, 25%) и с учетом возможной *скидки по дисконтной карте* (например, 5% от стоимости покупки). При отсутствии дисконтной карты в случае стоимости покупки выше 1 тыс. рублей дается *скидка* в 3%.

В определенные дни в список доставки включаются также плановые закупки постоянных покупателей – согласно хранящейся о них информации. Постоянным покупателям дополнительно дается скидка в 5%, но при этом общий процент скидки для одного покупателя не может превышать 9%. Сформированный список доставки распределяется между курьерами службы таким образом, чтобы каждый курьер выполнял доставку не менее 7 и не более 15 заказов (покупок) в день.

Компьютерная система отслеживает текущее количество каждого лекарства на складе, и если какого-то лекарства (его вида и/или дозировки) становится меньше определенного количества, то составляет *заявку в фармацевтическую фирму* на завоз в аптечный склад нужного количества этого лекарства, заявки выполняются несколько дней.

Цель моделирования работы службы доставки лекарств – определение оптимального соотношения процентов розничной наценки и скидок, а также минимально необходимого числа курьеров. Период моделирования – N дней $(10 \le N \le 25)$, шаг моделирования – один день.

Поток поступающих заказов на лекарства следует моделировать статистически: все составляющие каждого заказа определяются случайным образом. Плотность потока заказов зависит от величины розничной наценки, а вероятность заказа уцененных лекарств выше, чем неуцененных. Завоз лекарств на аптечный склад по заявке на фармацевтическую фирму моделируется с помощью случайной величины из диапазона от 1 до 3 – числа дней выполнения заявки (от даты заявки).

В параметры моделирования работы службы доставки следует включить числа N, M, K, начальный набор лекарств на складе, проценты скидок и наценки. В ходе моделирования должны быть доступны сведения о наличии лекарств на аптечном складе и о выполненных заказах, а по окончании моделирования следует дополнительно показать общую прибыль склада, его потери за счет списания лекарств и данные о загрузке курьеров.

Система контроля ассортимента книжного магазина

Книжный магазин осуществляет продажу широкого *ассортимента* книг нескольких издательств. Книги различаются по тематике и категории читателей (детская литература, учебники, научная литература по отдельным областям знаний, литература на иностранных языках, научная фантастика, фэнтези и т.п.).

Компьютерная система контроля ассортимента хранит данные о наличии и количестве экземпляров книг в магазине, при этом для каждой книги хранятся сведения о ее авторах, названии, издательстве, годе издания, количестве страниц, тематике и категории, цене и розничной наценке, рейтинге спроса. Для *новых книг* розничная наценка обычно на некоторый период устанавливается больше обычной.

В течение каждого рабочего дня система фиксирует *заказы на книги* (заказы записываются в магазине, а также поступают по телефону и электронной почте). Заказ включает фамилию покупателя, его номер телефона и/или электронный адрес, а также перечень заказываемых книг с указанием их количества. В заказе может быть указан только автор книги, без ее названия, но с пометой «новая», в этом случае имеется в виду последняя изданная книга данного автора. Если требуемая в заказе книга имеется в магазине, то она откладывается для покупателя, и делается соответствующая запись о продаже. В ином случае система сохраняет информацию о запрошенной книге в специальном списке заказов (с сохранением данных о заказчике). *Рейтинг спроса* каждой книги рассчитывается по числу ее заказов.

Система отслеживает фактическое количество экземпляров каждой книги в магазине; и если оно становится меньше определенного порога (3-5 экземпляров), то составляется *заявка в издательство* на доставку в магазин дополнительных экземпляров этой книги. Заявка в издательство составляется и в случае, когда поступило несколько заказов на (возможно, разные) книги этого издательства. Заявки выполняются в течение нескольких дней.

Основная функция системы управления ассортиментом – автоматизация обработки заказов на книги и составления заявок в издательства. Для тестирования работы системы требуется смоделировать поток поступающих заказов. Период моделирования – N дней $(10 \le N \le 30)$, шаг – 1-3 дня.

Поток заказов на книги следует моделировать статистически: все составляющие заказа подбираются случайным образом, но при этом новые книги заказываются чаще. Плотность потока заказов зависит от разнообразия ассортимента книг в магазине. Фактический срок доставки книг в магазин (т.е. срок выполнения заявки в издательство) также моделируется с помощью случайной величины из диапазона от 1 до 5 дней.

В параметры моделирования работы книжного магазина следует включить число N, начальный ассортимент книг в магазине, диапазоны разброса указанных случайных величин, процент обычной розничной наценки и наценки на новые книги. В ходе моделирования должна быть доступна информация об ассортименте магазина, о поступивших и обработанных заказах, а также о выполненных заявках в издательство. По окончании моделирования целесообразно вывести статистическую информацию о работе магазина, в том

числе – количество проданных книг по каждой тематике, топ самых продаваемых книг, их рейтинг и т.п.

Менеджмент курсов иностранного языка

Курсы иностранного языка предлагают обучение нескольким иностранным языкам (английскому, французскому, немецкому, японскому и др.), причем на нескольких *уровнях* (начальном, среднем, продвинутом) и с разной степенью *интенсивности* (интенсив, обычное и поддерживающее обучение). На курсах учатся *K слушателей (15≤ K ≤30)*, каждый слушатель может быть записан на обучение нескольким языкам, причем для каждого языка у него может быть свой уровень и интенсивность.

Занятия могут быть *индивидуальные* и *групповые*, причем в группу записываются слушатели одного языка, уровня и интенсивности. В группе не может быть менее 5 и более 10 человек (оптимально – 7 человек). *Длительность курса* для групповых занятий может варьироваться от двух недель до трех месяцев (для индивидуальных занятий подобных ограничений нет). Количество занятий в неделю (от 1 до 5 раз) и длительность курса зависит от его степени интенсивности.

Оплата курсов осуществляется авансом за две очередные недели обучения, стоимость курсов различается для разных языков, а стоимость индивидуального обучения выше группового.

Требуется создать компьютерную систему, автоматизирующую управление деятельностью курсов. Создаваемая система хранит информацию о преподаваемых языках, группах, слушателях, их посещаемости и оплате. В случае, когда некоторый слушатель прекращает посещать и оплачивать курсы, он исключается из соответствующей группы.

В начале очередного двухнедельного периода обучения система рассматривает поступившие *заявки на обучение* от новых и старых слушателей. В заявке указывается фамилия слушателя, язык, уровень и интенсивность обучения. В зависимости от поступивших заявок и численности уже существующих групп, система организует новые группы, дополняет старые группы новыми слушателями и/или переформирует старые группы, объединяя в одну несколько групп – так что численность групп остается близкой к оптимальной. В случае невозможности подобрать группу для нового слушателя система предлагает ему индивидуальные занятия, сохраняя тем не менее его заявку – с тем, чтобы через две недели вновь попробовать подобрать ему подходящую группу. По окончании установленного курса его слушатели автоматически переводятся на следующий уровень изучения языка (если они оплачивают очередной двухнедельный период и не подают другой заявки).

Необходимо испытать построенную систему менеджмента языковых курсов, задав некоторое их начальное состояние (языки изучения, слушатели, группы) и смоделировав поток заявок на обучение от новых слушателей. Цель моделирования – сбор статистики для анализа работы курсов, период моделирования – *M* месяцев *(3≤ M ≤12)*, шаг моделирования – две недели.

Поток поступающих заявок на обучение следует моделировать статистически: случайными величинами являются количество заявок на

очередном шаге моделирования и все составляющие каждой заявки: нужный иностранный язык, его уровень, интенсивность обучения. Прекращение обучения слушателей на курсах (в том числе досрочное) также следует моделировать как случайное событие, происходящее с определенной вероятностью.

В ходе моделирования должна быть возможность просмотреть информацию о текущей работе языковых курсов: о слушателях, группах, их численности и расписании, стоимости курсов и произведенной оплате и т.д. По окончании моделирования следует вывести статистику их работы в течение всего периода моделирования, в том числе – число слушателей каждого языка и уровня, число групп, средняя их численность и т.п.

Модель составления программ радиостанции

Некоторая радиостанция осуществляет круглосуточную трансляцию музыкальных произведений. В течение суток радиостанция предлагает несколько *радиопрограмм* $(7 \le K \le 12)$, посвященных разным *жанрам музыки*. Существует два вида программ: в одних подбор произведений выполняется *по заявкам пользователей*, другие же программы составляются как *хит-парады*. Длительность каждой программы – M часов $(1 \le M \le 3)$. Необходимо создать компьютерную систему, составляющую программы радиостанции в течение дня на основе поступающих заявок слушателей.

В фонотеке радиостанции хранятся музыкальные записи разных жанров и исполнителей. *Каталог фонотеки* учитывает для каждой музыкальной записи: жанр музыки (классика, джаз, рок, поп, рэп и др.), название произведения, авторы, исполнители, название и год выпуска альбома, количество минут звучания, рейтинг.

Для составления программы по заявкам слушателей система фиксирует поступающие по телефону заявки, в которых заказывается либо конкретное музыкальное произведение, либо любое произведение определенного автора, либо любое произведение из некоторого альбома, либо любая запись определенного исполнителя. Заявки выполняются по возможности последовательно, но так, чтобы не допускать однообразия исполняемых подряд произведений (например, не допускается подряд один и тот же исполнитель). При большом количестве поступивших заявок делается попытка выбрать очередную музыкальную запись так, чтобы удовлетворить несколько заявок. При невозможности выполнить все заявки удовлетворяются те, которые позволяют составить более разнообразную программу.

В хит-парадах проигрываются произведения определенного жанра, получившие наибольший *рейтинг* за последние дни. Рейтинг рассчитывается по поступившим заявкам слушателей, отдельно по каждому жанру.

Для тестирования построенной модели составления радиопрограмм необходимо статистически смоделировать поток заявок от слушателей. Каждая составляющая заявки (автор, произведение, альбом, исполнитель) определяется случайным образом. Период моделирования – N дней $(1 \le N \le 7)$, шаг моделирования – 10-30 минут.

В параметры модели составления музыкальных программ следует включить числа *K, M, N*, а также некоторые величины, от которых зависит поток поступающих заявок. На каждом шаге моделирования необходимо предусмотреть вывод и просмотр информации о прозвучавших и текущих программах радиостанции, о выполненных и невыполненных заявках. По окончании моделирования следует вывести список прозвучавших произведений, количество заявок слушателей, список самых рейтинговых произведений и т.п.

Система автоматизации функций секретаря

Некоторая фирма включает *K* различных *отделов* $(5 \le K \le 9)$, для согласования работы которых используется еженедельный *общий календарь*. В этом календаре представлены разнообразные *мероприятия* и *события*: ежедневные планерки сотрудников каждого отдела, еженедельные совещания руководителей отделов, командировки сотрудников отделов и руководства, периодические тематические семинары, в которых могут участвовать сотрудники разных разделов и т.п. У каждого календарного события есть дата, время начала и конца (или его длительность), место проведения, участники, степень важности и другие характеристики. Внутренние мероприятия фирмы могут проходить либо в помещениях отделов, либо в конференц-зале фирмы.

Требуется создать компьютерную систему, автоматизирующую отдельные функции секретаря фирмы – формирование и поддержку календаря фирмы и контроль обозначенных в нем событий. Основные функции системы:

- ✓ Уточнение (актуализация) календаря: из календаря автоматически удаляются уже произошедшие события дня, а также (по указанию пользователя) добавляются новые запланированные мероприятия или же уточняются характеристики событий, уже представленных в календаре;
- ✓ Выдача по запросу пользователя всей информации о календарных событиях, включая удобный просмотр событий, запланированных на определенный день или период дня – либо для всей фирмы, либо для определенного отдела, либо для определенного сотрудника;
- ✓ Посылка по электронной почте *напоминаний* о запланированных событиях всем участвующим в них сотрудникам фирмы; частота напоминаний и их начало зависит от степени важности события (например, ежедневно за 3-4 дня до события).

При добавлении в календарь новых событий проверяется их осуществимость, при этом могут быть выявлены *конфликты*: например, участие некоторого сотрудника в одно и то же время в двух разных мероприятиях, или же наложение по времени двух разных мероприятий, проводимых в конференц-зале. Информация о выявленных конфликтах показывается пользователю системы, и он должен разрешить каждый конфликт, изменяя некоторые характеристики либо добавляемого в календарь события либо же уже назначенного и внесенного в календарь события. Возможность последнего зависит, вообще говоря, от *категории пользователя*: например, руководители могут менять уже назначенные время и место событий, но рядовым сотрудникам это не разрешено. Желательно реализовать в компьютерной системе автоматический подбор нескольких способов разрешения возникающих

конфликтов – с тем, чтобы пользователь выбирал нужный способ из числа предложенных системой.

Необходимо испытать построенную систему автоматизации, установив для этого некоторое первоначальное состояние календаря событий и смоделировав пошаговое изменение времени. Цель моделирования – уточнение заложенных в систему процедур актуализации календаря. Период моделирования – N дней $(7 \le N \le 30)$, шаг – полчаса или час. В параметры моделирования можно включить условия выдачи напоминаний о событиях календаря и/или условия разрешения выявленных конфликтов. По окончании моделирования система выводит итоговый отчет по всем событиям, произошедшим на фирме в течение периода моделирования.

Моделирование работы курьерской службы

Курьерская служба создана для оперативной пересылки корреспонденции (служебных писем) между N $(3 \le N \le 7)$ *филиалами* крупной фирмы. Пересылка осуществляется несколькими $(1 \le M \le 5)$ *курьерами* службы, работой которых управляет диспетчер. Курьеры могут перевозить более одного письма, в начале рабочего дня они рассредоточены по филиалам фирмы. Известно среднее время, необходимое для переезда курьера из одного конкретного филиала в другой.

Требуется разработать систему, моделирующую функции диспетчера, который фиксирует поступающие в течение рабочего дня (с 9.00 до 18.00 вечера) *заявки на пересылку корреспонденции* и организует работу курьеров. Заявки на пересылку поступают случайным образом, но их распределение неравномерно по филиалам и времени дня, максимальное количество заявок возникает в середине рабочего дня. Заявка включает указание филиала, являющегося пунктом отправления письма; указание пункта назначения отправляемого письма, а также допустимый срок доставки (срочность доставки).

Цель моделирования – сбор информации о работе курьеров для последующей оптимизации курьерской службы: например, необходимо найти число курьеров, при котором сокращается общее время «холостых» переездов курьеров между филиалами фирмы (т.е. переездов без перевозки писем). Период моделирования – одна неделя, шаг моделирования – полчаса или час.

Интервал между появлением двух заявок следует моделировать как случайную величину из определенного диапазона (например, от 2 до 20 минут) – от этого диапазона зависит плотность потока заявок. Случайным образом определяются и составляющие заявки: пункты отправления и назначения и срочность заявки. *Фактическое время доставки* письма курьером отличается от среднего (обычного) времени на величину случайного отклонения (изменяется в диапазоне от –5 до 30 минут).

В параметры моделирования следует включить числа M и N, шаг моделирования, диапазоны изменения вышеуказанных случайных величин.

Визуализация работы курьерской службы может включать показ схемы расположения филиалов фирмы, на которой указывается текущее местоположение курьеров и номера выполняемых ими заявок. Отображаются также списки всех поступивших и выполненных заявок, для последних должно быть также указано время начала и окончания их выполнения. По окончании

моделирования должна выводиться статистическая информация, в том числе – показатели занятости курьеров фирмы, средняя длительность их поездок, общее время «холостых» переездов.

Система поддержки бронирования и заселения гостиницы

Небольшая гостиница содержит *K номеров (20≤ K ≤30)*, различающихся по степени комфорта и стоимости: «люкс», «полулюкс», одноместные, простые двухместные, двухместные с раскладным диваном (например, 70 у.е. за день проживания в одноместном номере, 120 у.е. – за номер «люкс»).

Требуется создать компьютерную систему, автоматизирующую управление занятостью номеров гостиницы. Система обрабатывает входной поток заявок двух видов:

- ✓ заявки, *бронирующие* определенные типы номеров на определенный срок;
- ✓ заявки на *заселение в текущий момент*.

Система хранит информацию о фактической занятости всех номеров и о их занятости в ближайшие дни (учитываются уже оплаченные вперед дни), а также сведения о произведенной брони номеров, и использует все эти данные при обработке заявок. При бронировании номеров система автоматически формирует сообщение-подтверждение брони, а при выезде постояльцев она оформляет им *счета*.

Стратегия обработки заявок строится так, чтобы добиться максимальной занятости гостиницы с целью увеличения ее прибыли. Для этого система гибко распоряжается номерным фондом: в частности, при нехватке нужных номеров можно использовать пустующие номера большей комфортности (по меньшей цене), например, при нехватке одноместных номеров можно поселить одного человека в двухместный номер (за 70% его стоимости).

Для тестирования построенной системы необходимо смоделировать входной поток заявок на бронирование и поселение. Вид и параметры каждой заявки определяются случайным образом. Интервал между появлением двух заявок следует моделировать как случайную величину из определенного диапазона (например, от 1 до 5 часов).

Период моделирования – *M* дней *(12≤ M ≤30),* шаг – несколько часов. Цель моделирования – изучение стратегий обработки заявок на заселение. В параметры моделирования следует включить: числа *K* и *M*, количество номеров каждой категории, характеристики используемых случайных величин.

В ходе моделирования система должна предоставлять всю необходимую информацию о занятости номеров гостиницы. По окончании моделирования выводится статистика заселения номеров, выполненных заявок, процент загруженности отдельных категорий номеров и гостиницы в целом.

Моделирование работы морского порта

Требуется создать компьютерную модель обслуживания потока *заявок на разгрузку*, поступающих от *грузовых судов* (сухогрузов и танкеров), прибывающих в морской порт. Грузовые суда прибывают в порт согласно *расписанию*, но возможны опоздания и досрочные прибытия. Расписание

включает день и время прибытия, название судна, вид груза и его вес, а также планируемый срок стоянки в порту для разгрузки.

Для разгрузки судов в порту используются три вида *разгрузочных кранов*, соответствующих трем *видам грузов*: сыпучим и жидким грузам, контейнерам. Число разгрузочных кранов каждого вида ограничено, так что поступающие заявки на разгрузку одного вида груза образуют *очередь*. Длительность разгрузки судна зависит от вида и веса его груза, а также некоторых других факторов, например, погодных условий. Любой дополнительный (сверх запланированного срока) день стояния судна в порту (из-за ожидания разгрузки в очереди или из-за задержки самой разгрузки) влечет за собой выплату *штрафа* (например, 2 тыс. у.е. за каждый дополнительный день простоя судна).

При моделировании прибытия судов отклонение их от расписания рассматривается как случайная величина с равномерным распределением в некотором интервале (например, от -2 до 9 дней). Еще одной случайной величиной, изменяющейся в фиксированном диапазоне (например, от 0 до 12 дней), является время задержки окончания разгрузки судна по сравнению с обычным (зависящим только от вида груза и его веса).

Цель моделирования работы морского порта – определение для заданного расписания прибытия судов минимально достаточного числа кранов в порту, позволяющего уменьшить штрафные суммы. Период моделирования – месяц, шаг моделирования – 1-3 дня. В параметры моделирования следует включить расписание прибытия судов, количество кранов каждого вида, диапазоны разброса случайных величин (отклонения от расписания прибытия и отклонения от обычного времени разгрузки), а также шаг моделирования.

Визуализация моделируемого процесса должна предусматривать показ очередей у разгрузочных кранов, приход судов в порт и их отход после разгрузки. Должен быть показан также список произведенных разгрузок, в котором указывается название разгруженного судна, время его прихода в порт и время ожидания в очереди на разгрузку, время начала разгрузки и ее продолжительность. По окончании моделирования должна быть выведена итоговая статистика: число разгруженных судов, средняя длина очереди на разгрузку, среднее время ожидания в очереди, максимальная и средняя задержка разгрузки, общая сумма выплаченного штрафа.

Моделирование в сфере обслуживания

Моделирование обслуживания в филиале банка

Необходимо создать компьютерную модель обслуживания *потока заявок*, поступающих от *клиентов банка*, несколькими *клерками* ($2 \leq N \leq 7$) в одном из филиалов банка. Известно *недельное расписание* работы филиала банка: 5 дней по 8 часов и один день – 6 часов, возможны перерывы на обед.

При моделировании работы заявки на обслуживание (т.е. приход клиентов) поступают случайным образом. Случайной величиной является отрезок времени между последовательным появлением двух заявок, она имеет нормальное или равномерное распределение в некотором интервале (например, от 0 до 10 минут), причем плотность потока заявок зависит от дня недели и

времени дня – в конце недели и в конце дня клиенты приходят чаще, т.е. плотность потока выше. Длительность обслуживания каждой заявки – также случайное число в некотором диапазоне (например, от 2 до 30 минут), но длительность не зависит от входного потока заявок. Еще одна случайная величина – прибыль, получаемая банком от обслуживания клиента, она варьируется в пределах от 3 тыс. до 50 рублей.

Поступившие заявки (клиенты) образуют *общую очередь*, максимальная длина которой – K человек $(10 \le K \le 25)$. Если очередь достигла такой длины, то вновь прибывающие клиенты уходят, и вероятность прихода следующих уменьшается – тем самым банк теряет своих потенциальных клиентов.

Клиенты банка ожидают своей очереди на обслуживание в общем зале с *информационным табло*, на котором высвечиваются номер клиента, взятого только что на обслуживание, и номер места клерка, обслуживающего этого клиента. В каждый день работы филиала заявки на обслуживание нумеруются последовательно, начиная с 1, по мере их прихода в банк.

Цель моделирования работы банка – определение прибыли банка и ее зависимости от числа работающих клерков; выявление "узких" мест в работе банка: нехватки клерков (возможное следствие этого – потеря клиентов), простой клерков (следствие – лишние траты на их зарплату). Прибыль высчитывается с учетом дневной зарплаты каждого клерка (2 тыс. руб.).

Период моделирования – месяц, шаг моделирования – интервал времени от 10 минут до 1 часа. Следует включить в параметры моделирования: числа N и K, шаг моделирования, диапазоны разброса случайных величин – промежутка между приходом клиентов и время их обслуживания, законы распределения этих случайных величин. Необходимо также учесть в модели уменьшение потока клиентов при наличии достаточно большой (более 7 человек) очереди.

Визуализация моделируемого процесса должна предусматривать показ текущей ситуации в банке, том числе – скопившуюся очередь, занятость клерков, появление новых и уход обслуженных клиентов, информационное табло. Следует предусмотреть вывод в ходе моделирования и по его окончании подсчитанной статистики: количества обслуженных и потерянных клиентов, максимальную, минимальную и среднюю длину очереди, среднее время ожидания клиентов в очереди, среднюю занятость клерков, а также полученную банком прибыль.

Модель обслуживания на бензозаправочной станции

Требуется создать компьютерную модель обслуживания потока *заявок*, поступающих от владельцев автомашин, несколькими $(3 \le K \le 7)$ *разливочными автоматами* на бензозаправочной станции.

Бензозаправочная станция работает круглосуточно. При моделировании ее работы заявки на обслуживание (т.е. приезд автомашин на заправку) поступают случайным образом: случайной величиной является отрезок времени между появлением двух заявок, оно имеет нормальное или равномерное распределение в некотором интервале (например, от 0 до 20 минут), причем плотность потока заявок зависит от дня недели, времени дня и цены на бензин. Случайными величинами (не зависящими от плотности потока заявок) является

объем закупаемого каждым владельцем бензина (от 10 до 50 литров) и его *марка*. Длительность обслуживания каждой заявки (заполнение бака автомашины) зависит от объема закупаемого бензина (1-3 минуты).

Максимально возможная длина очереди около разливочного автомата – N машин $(5 \leq N \leq 9)$. Если у автомата скопилась максимальная очередь, то вновь прибывающие автомашины уезжают без обслуживания – тем самым станция теряет своих потенциальных клиентов. Очереди формируются по определенному закону – так, что разница между максимальной и минимальной очередью у автоматов с одним типом бензина не превышает 2 человека.

Цель моделирования работы бензозаправочной станции – определение *торговой наценки* на бензин, при которой увеличивается прибыль от его продажи, а также вычисление необходимых запасов бензина каждой марки (на день и на неделю). Считается, что торговая наценка может устанавливаться владельцем станции в пределах от 5 до 15% от базовой стоимости литра бензина (отдельно для каждой его марки), и каждый процент наценки уменьшает поток покупателей на 3 %. Прибыль станции зависит от количества проданного бензина и торговой наценки.

Период моделирования – неделя, шаг – интервал времени от 10 минут до 1 часа. В параметры моделирования следует включить величины K и N, процент торговой наценки на бензин; диапазоны изменения случайных величин – временного промежутка между последовательным поступлением заявок на обслуживание, марки и объема требуемого в каждой заявке бензина.

Визуализация моделируемого процесса должна предусматривать показ очередей у каждого автомата, приезд и отъезд автомашин, а также вывод статистической информации о работе станции: количество обслуженных автомашин и средний объем проданного бензина за день и неделю (по отдельности для разных марок), количество автомашин, уехавших без заправки, полученная станцией прибыль.

Моделирование работы автосервиса

Автосервис предоставляет разные виды *услуг* по ремонту и обслуживанию автомобилей и включает нескольких цехов: техосмотра, кузовного ремонта, шиномонтажа, ремонта двигателя. В каждом цеху работает несколько *мастеров* $(2 \leq K \leq 7)$. Известно *недельное расписание* работы автосервиса: 5 дней по 12 часов и два дня по 8 часов, без перерывов на обед.

Необходимо разработать имитационную модель работы автосервиса, при которой *заявки на обслуживание* автомобилей поступают случайным образом, каждая заявка включает одну или несколько услуг. Каждая услуга выполняется в определенном цеху, известна средняя длительность ее выполнения и получаемая при этом прибыль. *Фактический срок выполнения заявки* может отличаться от среднего на некоторую случайную величину, изменяющуюся в некотором диапазоне (например, от часа до нескольких дней, в зависимости от вида услуги). Случайной величиной является также отрезок времени между последовательным появлением двух заявок, она имеет нормальное или равномерное распределение в некотором интервале (например, от 15 минут до

часа), причем плотность потока заявок зависит от времени дня – в середине рабочего дня заявки поступают чаще, т.е. плотность потока выше.

Поступившие заявки образуют несколько *очередей* – по числу цехов автосервиса, причем в общем случае заявка может сохраняться в очереди несколько дней. Один и тот же автомобиль может находиться одновременно в нескольких очередях (с заявками на разные услуги), но его обслуживание в нужных цехах производится последовательно. Максимальный общий срок обслуживания каждого автомобиля – неделя, и если по окончании этого срока ремонт (обслуживание) автомобиля еще не закончен, то владелец забирает его из автосервиса – тем самым автосервис теряет своих клиентов.

Цель моделирования работы автосервиса – определение оптимального соотношения числа рабочих в его цехах, выявление "узких" мест в его работе (таких как нехватка мастеров или их простой). Недельная зарплата каждого мастера определяется как 35% от приносимой им прибыли автосервису, но не менее 1 тыс. рублей в день. Период моделирования – неделя, шаг – M часов.

Следует включить в параметры моделирования величины K и M, а также диапазоны разброса вышеуказанных случайных величин. Визуализация моделируемого процесса должна предусматривать показ текущей ситуации в автосервисе, том числе – получившиеся очереди в каждом цеху, занятость рабочих, появление новых заявок. Также должен быть предусмотрен вывод подсчитанной статистики: общее число обслуженных автомобилей и предоставленных услуг разного вида, среднее время обслуживания одного автомобиля; средняя длина очередей в каждом цеху; средняя занятость рабочих и средняя их зарплата, общая прибыль автосервиса.

Модель работы магазина или супермаркета

Необходимо разработать имитационную модель обслуживания покупателей *супермаркета* или обычного *магазина* несколькими $(1 \le K \le 7)$ *кассами* супермаркета или *продавцами* магазина. Известно *недельное расписание* работы магазина: 11 часов по рабочим дням, 9 часов в субботу, и 7 часов в воскресенье. Супермаркет работает круглосуточно.

При моделировании работы супермаркета/магазина его *покупатели* приходят случайным образом: случайной величиной является отрезок времени между последовательным появлением двух покупателей. Эта случайная величина имеет нормальное или равномерное распределение в некотором интервале (например, от 0 до 7 минут), причем плотность потока заявок зависит от дня недели, времени дня и величины очередей у касс/продавцов (в конце недели и в конце дня клиенты приходят чаще, плотность потока выше). Длительность обслуживания каждого покупателя также случайное число в некотором диапазоне (например, от 1 до 7 минут), но оно не зависит от входного потока заявок. Еще одной случайной величиной является сумма покупки (от 30 до 9 тыс. руб.), причем сумма не зависит от других случайных величин.

Максимально возможная длина *очереди* у каждой кассы – N человек $(5 \le N \le 8)$, не считая обслуживаемого покупателя. Очереди формируются по определенному закону – так, что разница между максимальной и минимальной очередью у касс не превышает три человека. Если у каждой кассы скопилась

очередь из N человек, то вновь прибывающие покупатели уходят, и вероятность прихода следующих уменьшается – тем самым супермаркет теряет своих потенциальных покупателей.

Максимальная длина очереди около каждого продавца-консультанта – 3 человека (не считая обслуживаемого покупателя). Аналогично, разница между максимальной и минимальной очередью у продавцов не превышает 2 человека. Если у каждого продавца скопилась очередь из трех человек, то вновь прибывающие покупатели уходят, и потенциальные покупатели теряются.

Цель исследования работы супермаркета или магазина – определение оптимальных *режимов* его работы, т.е. режимов, при которых работающие кассы или продавцы всегда заняты, и увеличиваются прибыли от продаж. Режим работы включает число касс или продавцов, *рекламу* и *скидки* на товары.

Считается, что затрата 7 тыс. руб. в день на рекламу увеличивает поток покупателей на 10%, а при объявлении скидок на товары каждый процент скидки увеличивает плотность потока на 0.5 %. Известна также средняя прибыль, получаемая при обслуживании каждого покупателя – 9 % от стоимости покупки, и дневная зарплата каждого кассира или продавца (1.5 тыс. руб.), причем продавцов можно нанимать поденно.

Период моделирования – неделя, шаг – интервал времени от 10 до 60 минут. Кроме шага, в параметры моделирования следует включить числа K и N, диапазоны разброса случайных величин – промежутка между последовательным приходом покупателей, времени их обслуживания, стоимости покупки, а также затраты на рекламу, величину скидки, прибыль от суммы покупки в 1 тыс. руб., зарплату кассира или продавца и степень уменьшения потока покупателей при возникновении максимальной очереди.

Визуализация моделируемого процесса должна предусматривать показ очередей у каждой кассы или продавца, приход и уход покупателей, а также вывод статистической информации, собираемой в ходе моделирования: количество обслуженных и потерянных (потенциальных) покупателей, средняя длина очереди у касс и среднее время ожидания в ней, средняя занятость продавцов или касс, общая прибыль, полученная супермаркетом или магазином.

Моделирование работы парикмахерского салона

Необходимо создать компьютерную модель работы парикмахерского салона, состоящего из 2-3 *залов*. В каждом зале работает несколько *($2 \le K \le 5$) мастеров* и предоставляются услуги определенного вида. Известно *недельное расписание* работы парикмахерской: 5 дней по 12 часов и один день (суббота) – 8 часов, без перерывов на обед.

При моделировании работы салона *заявки на обслуживание* (т.е. приход клиентов) поступают случайным образом. Случайной величиной является отрезок времени между последовательным появлением двух заявок, она имеет нормальное или равномерное распределение в некотором интервале (например, от 0 до 10 минут), причем плотность потока заявок зависит от дня недели и времени дня: в начале и конце недели, как и в начале и конце дня клиенты приходят чаще, т.е. плотность потока выше.

Каждая заявка на обслуживание обозначает нужную услугу или даже несколько услуг (в соответствующих залах салона). Разные услуги появляются в заявке с разной вероятностью. Известна средняя длительность обслуживания клиента по каждой услуге (от 20 минут до 2 часов) и ее цена (от 200 до 3 тыс. рублей). *Фактическая длительность обслуживания* клиента может отличаться от средней на некоторую случайную величину, изменяющуюся в заданном диапазоне (от 5 до 30 минут). Недельная зарплата каждого мастера определяется как 40% от цены предоставленных им услуг, но не менее 7 тыс. рублей.

Поступившие заявки (клиенты) образуют несколько *очередей* – по числу залов салона. Максимальная длина каждой очереди – 5 человек, если очередь достигла такой длины, то вновь прибывающие клиенты уходят, и вероятность прихода следующих уменьшается – тем самым парикмахерская теряет своих потенциальных клиентов.

Цель моделирования парикмахерского салона – исследование его работы в зависимости от потока заявок и числа мастеров в каждом зале, выявление "узких" мест в работе: нехватки мастеров (возможное следствие этого – потеря клиентов), простой мастеров (следствие – падение их зарплаты). Период моделирования – неделя. Следует включить в параметры моделирования: числа *K* мастеров в каждом зале салона, шаг моделирования – интервал времени от 15 до 30 минут, а также диапазоны разброса случайных величин – промежутка времени между возникновением заявок и отклонения от средней длительности обслуживания.

Желательно включить в компьютерную модель возможность обслуживания клиентов по *предварительной записи*, которая фиксирует желаемое время услуги, саму услугу и контактный телефон клиента. Обслуживание таких клиентов осуществляется без очереди, с опозданием не более чем в 15 минут от предварительно зафиксированного времени.

Визуализация моделируемого процесса должна предусматривать показ текущей ситуации в салоне, том числе – скопившиеся очереди, занятость мастеров, появление новых и уход обслуженных клиентов. Следует предусмотреть подсчет и вывод статистической информации (как во время работы салона, так и по окончании моделирования): число обслуженных клиентов, средняя длительность обслуживания, длина очереди и занятость мастеров в каждом зале, средняя зарплата мастеров и общее время их простоя.

Экономические игры

Модель управления страховой компанией

Рассматривается работа страховой компании, выполняющей *страхование* населения по трем направлениям (видам страховок): страхование жилища, страхование автомобиля и страхование здоровья. *Договор страхования* учитывает *условия страховки* каждого вида: размер месячного, квартального или годового взноса (например, 90 у.е.), срок действия договора (например, год), максимальную сумму страхового возмещения (например, 20 тыс. у.е.), франшизу – размер ущерба, с которого начинаются страховые выплаты.

Создаваемая система реализует экономическую игру, пользователь которой – менеджер, управляющий работой страховой компании. В начале игры страховая компания обладает некоторым капиталом (например, 30 тыс. у.е.), и известен *базовый спрос* (т.е. количество потенциальных покупателей) на страховки каждого вида, при заданных начальных условиях страховок. Игрок-менеджер компании может изменить эти условия на новые, а также установить срок их действия (например, 6 месяцев).

Игра моделирует работу компании в течение *M* месяцев *(6≤ M ≤24)*. Шаг моделирования – один месяц, он включает следующие *операции*:

1. выплата государству постоянного налога на общий капитал компании (например, 9% от суммы);
2. *продажа* населению всех видов страховок согласно действующим условиям страховок и текущему спросу на них (для разных видов страховок спрос может быть различным);
3. выплата населению страховых сумм согласно *страховым случаям*, произошедшим в текущем месяце;
4. определение на последующие месяцы условий (сохранение их или изменение) для тех видов страховок, срок действия которых истек.

Операции 1-3 выполняются автоматически, а 4 – менеджером компании.

Общий капитал компании складывается из исходного капитала и стоимости уже проданных страховок, за вычетом выплаченного налога и суммы выплаченных страховок. В ходе игры компания может обанкротиться – это происходит, когда она не выполняет своих финансовых обязательств по выплатам страховых вознаграждений (операция 3). Игра заканчивается либо после установленного числа шагов, либо досрочно, после банкротства компании.

Текущий спрос на каждый вид страховки определяется соотношением общей стоимости страховки (размер страхового взноса, умноженный на длительность договора) и максимальной суммы страхового возмещения: чем меньше отношение стоимости страховки к сумме страхового возмещения, тем больше это повышает базовый спрос. Количество проданных страховок каждого вида следует рассчитывать с учетом вероятностного фактора, т.е. как текущий спрос, скорректированный случайной величиной, изменяющейся в некотором диапазоне (например, от 0 до 10 покупателей).

Число страховых случаев каждого вида, произошедших за один месяц, необходимо моделировать как случайную величину, имеющую равномерное распределение в некотором диапазоне (например, от 1 до 25 случаев). Для каждого страхового случая сумма возмещения подсчитывается как величина, пропорциональная размеру ущерба, например, как максимальная сумма страхового возмещения, умноженная на коэффициент ущерба – случайное число в диапазоне (0,1].

Цель моделирования работы страховой компании – выявление условий страхования, позволяющих наращивать ее общий капитал. В изменяемые параметры целесообразно включить число *M*, размер исходного капитала страховой компании, процент налога на общий капитал, базовый спрос на

страховки, а также диапазон изменения случайной величины – количества страховых случаев.

В ходе игры менеджеру компании должна быть доступна информация о базовом спросе на страховки и о состоянии дел компании (общий капитал, условия страховок, количество проданных страховок каждого вида, выплаты по страховым случаям и др.).

Возможны следующие усложнения рассмотренной игры:

- новые виды страховок и подвиды старых (ОСАГО, КАСКО);
- несколько игроков, конкурирующих между собой за спрос на страховки.

Моделирование инвестиций в строительство

Требуется разработать программную систему для игры, в которой участвуют N $(2 \le N \le 5)$ игроков-людей или игроков-программ. Каждый из участников игры считается управляющим *инвестиционного фонда*, осуществляющего вложение капитала в строительство жилых домов и супермаркетов в некотором городе.

Для рассматриваемого города известны средние стоимости и сроки строительства домов (можно рассмотреть несколько типов домов – панельные, монолитные, кирпичные) и супермаркетов, например: стоимость строительства дома – от 8 млн. у.е., срок строительства – от 7 месяцев; стоимость строительства супермаркета – от 2.5 млн. у.е., срок – от 5 месяцев. Известен также обычный *спрос на жилье* (количество квартир в месяц), который зависит от сезона года (возрастает с весны и максимален осенью), а также средний *уровень продаж* (прибыль) в супермаркетах, который также зависит от сезона (возрастает с осени и максимален зимой).

В начале игры каждый игрок получает во владение одинаковую сумму денег, например, 37 млн. у.е., и определяет, в какие объекты и в каком количестве он будет инвестировать деньги. Предполагается, что дома и супермаркеты он строит в одном микрорайоне, так что строительство каждого дополнительного супермаркета может повышать спрос на рядом строящееся жилье, а строительство дополнительного дома может увеличивать уровень продаж в соседних супермаркетах. Цель игры – получить максимальную прибыль для вложенного капитала.

Каждый шаг игры соответствует одному месяцу, в начале которого выполняются следующие *операции* для каждого игрока:

1. выплата месячных затрат на строительство объектов в текущем месяце: предполагается, вложенная сумма равномерно расходуется в течение всего срока строительства каждого объекта;
2. получение дохода от продажи (в прошлом месяце) жилья в строящихся и уже построенных домах по ценам, заявленным в прошлом месяце;
3. получение прибыли от продажи (в прошлом месяце) товаров в уже действующих супермаркетах, согласно текущему уровню продаж.
4. определение объема выставляемого на продажу (в текущем месяце) жилья в строящихся и уже построенных домах и цен на кв.м. жилья (обычно цена растет в ходе строительства);

5. определение затрат на *рекламу* продаваемого жилья: считается, что каждая потраченная на рекламу 1 тыс. у.е. увеличивает объем продаваемого в этом и следующем месяце жилья на 0,5 %;
6. определение затрат на рекламу товаров в уже построенных супермаркетах: считается, что каждая потраченная на рекламу 0.5 тыс. у.е. увеличивает прибыль от продаж на 3%;

Важно, что продажа выставленного игроками жилья в строящихся и уже построенных домах происходит через одно *риэлторское агентство*, которое удовлетворяет текущий спрос на жилье (для каждого типа домов) в зависимости от заявленной игроками цены 1 кв.м. жилья, их затрат на рекламу и других рассмотренных выше факторов. Если в целом предложение превышает спрос, то предложение жилья с более низкой ценой удовлетворяется в первую очередь.

Общий капитал каждого игрока складывается из стоимости строящихся объектов (она равна потраченной сумме денег на их строительство), стоимости непроданного жилья в построенных домах (рассчитывается по себестоимости 1 кв.м.), стоимости построенных супермаркетов (на 60% выше потраченной на их строительство суммы) и вырученной в результате всех продаж суммы денег.

Игра заканчивается после установленного числа M шагов $(6 \le M \le 24)$, при этом побеждает тот игрок, который имеет наибольший общий капитал. В изменяемые параметры игры целесообразно включить M и N, а также параметры строительства объектов разного типа, например, сроки и стоимость.

Описанная игра может быть усложнена за счет *задержек строительства* объектов, происходящие с некоторой вероятностью.

Создаваемая игровая программа должна включать:

- модуль, управляющий ходом моделирования и контролирующий соблюдение игроками правил игры;
- модуль, реализующий работу риэторского агентства;
- несколько модулей-игроков, реализующих разные стратегии игры.

В ходе игры игрокам должна быть доступна информация о строящихся и построенных объектах других игроков и проданном ими жилье.

Система управления инвестиционным портфелем

Создаваемая система реализует экономическую игру, участник которой – менеджер, управляющий работой некоторого *инвестиционного фонда*. Фонд осуществляет различные вложения собранных денежных средств с целью получения прибыли. Возможны вложения в срочные *депозиты* банков (валютные и рублевые), в *драгоценные металлы* (в золотые слитки и др.), в государственные *облигации*, в *акции* предприятий – все эти виды вложений различаются *доходностью* и *риском* (обычно доход пропорционален риску).

В начале игры устанавливается общий капитал фонда (например, 560 тыс. у.е.), и определяется его *портфель*, т.е. какая часть капитала фонда куда будет вкладываться. В портфеле не обязательно присутствуют все виды вложений (например, нет вложений в депозиты, если они малодоходны), в то же время допускается несколько разных вложений одного вида (например,

вкладываются разные суммы в акции разных предприятий или в депозиты одного или нескольких банков).

В начале игры определена *внешняя конъюнктура* – возможные в текущий момент виды вложений и их условия, к примеру, известен процент дохода по годовым депозитам некоторого банка, стоимость акций некоторой компании и их доходность и т.п.

Игра моделирует работу фонда в течение M месяцев $(12 \le M \le 30)$. Шаг моделирования – один месяц, в конце каждого месяца выполняется:

1. подсчет доходности по всем составляющим инвестиционного портфеля, определение общей суммы прибыли и процента доходности за этот месяц;
2. выплата государству налога на доход фонда (например, 17% от суммы прибыли);
3. учет новых поступивших денежных средств фонда (в частности, за счет продажи населению паев фонда);
4. учет расходов фонда (например, в случае возврата паев их держателями);
5. реструктуризация портфеля с учетом изменённой (на шагах 3 и 4) общей суммы капитала фонда и с учетом изменений во внешней конъюнктуре (например, новой стоимости акций).

Операции 1 и 2 выполняются автоматически, операция 5 – игроком-менеджером, а операции 3 и 4 могут выполняться как автоматически, так и игроком. Обычно поступление новых денежных средств в фонд (шаг 3) зависит от его доходности: чем выше доходность за месяц, тем больше спрос на паи фонда и наоборот – падение доходности влечет за собой обратную продажу (возврат) паев (шаг 4). *Реструктуризация* инвестиционного портфеля может включать, например, продажу части акций или покупку новых, а также вложения в новые депозиты.

При подсчете доходности фонда за месяц (шаг 1) учитывается, что доходность депозита и облигаций известна заранее, а доходность акций и драгоценных металлов определяется внешней конъюнктурой (текущей ценой). Текущую цену следует моделировать как цену прошлого месяца, скорректированную случайной величиной, изменяющейся в некотором диапазоне по определенному вероятностному закону. Аналогичным образом можно моделировать изменение процентных ставок по депозитам по окончании их срока и другие показатели внешней конъюнктуры.

Цель моделирования – выявление пропорций инвестиционного портфеля, позволяющих устойчиво получать прибыль и наращивать общий капитал инвестиционного фонда. В изменяемые параметры целесообразно включить число M, размер исходного капитала инвестиционного фонда, первоначальную структуру инвестиционного портфеля, процент налога на доход, а также диапазоны разброса случайных величин, от которых зависит изменение внешней конъюнктуры.

На каждом шаге игры игроку-менеджеру должны быть доступны все данные о состоянии дел инвестиционного фонда: суммарный капитал, общий доход и доход по отдельным статьям портфеля и др., а также информация о внешней конъюнктуре: цена акций, процентные ставки по депозитам и т.п. По окончании игры можно предусмотреть вывод дополнительных статистических данных о работе фонда, например, уровень продажи и возврата паев фонда.

Модель работы рыбоводческого хозяйства

Моделируется работа хозяйства *по разведению и продаже* нескольких видов *рыбы* (форели, карпа и др.). Для разведения рыбы служат несколько *(2 ≤ K ≤ 7) прудов* хозяйства, причем в каждом пруду в любой момент времени разводится не более одного вида рыбы.

Система моделирования реализует экономическую игру, участник которой – управляющий хозяйством. В начале моделируемого периода управляющий располагает определенным денежным капиталом (например, 560 тыс. рублей) для закупки сухого корма для рыб и, при необходимости, закупки мальков для развода. Управляющий также заключает *контракт* с торговым домом сроком на *M* недель *(6≤M ≤24)*, по которому обязуется еженедельно покупать сухого корма на определенную сумму и поставлять (продавать) определенное число килограммов рыбы. Контракт фиксирует стоимости кормов и рыбы на каждые очередные 3 недели контрактного периода. В случае невыполнения обязательств по контракту управляющий будет обязан выплатить оговоренную в контракте *неустойку*, например, 1 тыс. руб. за каждый непроданный килограмм рыбы.

Шаг моделирования соответствует *производственному циклу* в 1 неделю, в течение которого происходит откорм рыбы в каждом из действующих прудов и соответственно – рост популяций рыб. Количественные изменения популяций измеряются в килограммах и моделируются с помощью следующих рекуррентных соотношений:

$$Ny^{\circ} = \alpha * Na$$

$$Na^{\circ} = \beta * Ny - \delta * Na$$

где Ny, Na – количества соответственно молодых и взрослых рыб в начале недели, а Ny°, Na° – эти количества в конце недели; α и β – коэффициенты рождаемости и выживаемости молодняка; δ – коэффициент смертности взрослых особей. Указанные коэффициенты могут различаться для разных видов рыб. Стоимость кормов, необходимых для откорма рыб в течение недели, определяется по формуле

$$P = Q*(Ny/2 + Na)$$

где Q – стоимость сухого корма, необходимого для откорма в течение недели 1 кг взрослых рыб соответственно. Если в пруд закладывается количество корма меньше, чем требуется численностью популяции, то происходит гибель ее части, пропорционально нехватке корма (при этом в равной мере погибают молодые и взрослые рыбы).

Цель моделирования – выявление стратегий производства, позволяющих по окончании срока контракта приумножить общий капитал хозяйства. Общий капитал складывается из стоимости всей имеющейся в настоящий момент рыбы (по контрактной цене последнего трехнедельного периода) и наличных денег. В ходе моделирования хозяйство может обанкротиться – это происходит, когда оно не может выполнить всех обязательств, определенных контрактом.

В модели следует учесть, что каждый пруд должен время от времени проходить *очистку* – при этом он на неделю должен быть освобожден от рыбы. Дополнительно можно учесть разные *неблагоприятные случайные события* (например, экстремальные колебания температуры пруда), приводящие к гибели некоторой части рыбы. Процент потери целесообразно моделировать как

случайную величину, изменяющуюся в некотором диапазоне (например, 5-10% от всей популяции пруда).

В параметры моделирования следует включить количество прудов K и первоначальное количество рыбы в них, длительность контракта M, все данные самого контракта, размер исходного капитала хозяйства, диапазон разброса случайной величины – процента гибели рыбы при неблагоприятных факторах.

В ходе моделирования должна быть доступна вся информация о текущем состоянии дел хозяйства и о действующем контракте.

Моделирование работы животноводческой фермы

Рассматривается работа животноводческой фермы, на которой в начальный момент имеется определенное количество животных (коров или овец), например: 90 голов взрослых животных, 70 голов молодняка и 85 старых животных. Система моделирования реализует экономическую игру, пользователь программы – владелец фермы.

В начале моделируемого периода владелец фермы имеет определенный денежный капитал, например, 80 тыс. у.е., для закупки кормов для животных. Владелец фермы заключает *контракт* с товарной биржей сроком на K лет $(3 \le K \le 5)$, по которому обязуется ежегодно покупать кормов на определенную сумму, и ежегодно продавать определенное число голов молодняка, взрослых и старых животных. Животные разного возраста продаются по разной цене, которая фиксируется в контракте; цена может меняться год от года. В случае невыполнения обязательств по контракту владелец фермы должен выплатить определенную *неустойку* (оговариваемую в контракте), например, 9 тыс. у.е. за каждое непроданное животное.

Шаг моделирования соответствует *производственному циклу* в 1 год, включающему откорм животных, рост их поголовья и продажу на бирже. Следует считать, что численность поголовья животных в начале и конце года определяется следующими рекуррентными соотношениями

$$Ny^\circ = \alpha * Na + \beta * No$$
$$Na^\circ = \delta * Ny$$
$$No^\circ = Na + (1 - \rho) * No$$

где Ny, Na, No – количество соответственно молодняка, взрослых и старых животных в начале года, а $Ny^\circ, Na^\circ, No^\circ$ – их количество в конце года;

α и β – коэффициенты рождаемости молодняка у взрослых и старых животных; δ – коэффициент выживаемости молодняка; ρ – коэффициент смертности старых животных.

Стоимость кормов, необходимых для питания животных в течение года, определяется по формуле $P = R * (Ny/2 + Na + No/3)$

где R – стоимость корма, необходимого взрослому животному в течение одного года. Если на текущий год корма закуплено меньше, чем требуется текущим поголовьем скота, то происходит частичный падеж скота, пропорциональный нехватке кормов (причем в равной мере погибают животные всех возрастов).

Цель моделирования – выявление стратегий контрактации и производства на ферме, позволяющих владельцу фермы приумножить общий капитал по

окончании срока контракта. В изменяемые параметры моделирования следует включить длительность контракта *K*, все данные самого контракта (отдельно по каждому году), размер исходного денежного капитала владельца фермы, а также текущее количество на ферме молодняка, взрослых и старых животных.

Общий капитал фермы складывается из стоимости всех имеющихся на настоящий момент животных (по контрактной цене текущего года) и наличных денег. В ходе моделирования ферма может обанкротиться – это происходит, когда фермер не может выполнить всех обязательств, определенных контрактом.

В модели следует учесть возможные *неблагоприятные события* (например, суровые погодные условия), происходящие с определенной вероятностью за период контракта и приводящие к гибели некоторой части скота. Процент погибающих животных следует моделировать как случайную величину, изменяющуюся в диапазоне 5- 20% от общего поголовья.

В ходе моделирования должны быть доступны все основные данные о текущем состоянии дел фермы и ее рентабельности.

Модельные системы контроля

Модельная система регулирования домашнего отопления

Система отопления представляет собой водный обогреватель, работающий на газе и нагревающий воду в батареях отопления, установленных в *комнатах дома* (включая гостиную, рабочий кабинет, кухню, ванную комнату и другие помещения). В каждой комнате есть специальный *клапан*, регулирующий поступление горячей воды в батареи этой комнаты, и, тем самым – температуру в этой комнате. Возможные положения клапана (полностью открыт/закрыт/полуоткрыт) могут быть установлены дистанционно. В каждой комнате находятся также *датчик текущей температуры* и *инфракрасный датчик присутствия людей*, их значения используются для регулирования температуры.

Основное назначение системы автоматического регулирования отопления – поддержание в каждой из комнат дома нужной температуры путем установки соответствующих положений клапанов обогревателя. Пользователи системы (жильцы дома) могут включать и выключать систему регулирования, а также задавать *рабочую температуру* в каждой комнате, т.е. температуру, которая должна быть в комнате в случае присутствия в ней людей. В случае же их отсутствия в целях экономии должна поддерживаться *температура ожидания* – она определяется на M градусов $(1 \le M \le 5)$ ниже рабочей в этой комнате.

Системе известно обычное *недельное расписание* пребывания людей в комнатах дома – для того, чтобы заранее, к моменту ожидаемого появления в конкретной комнате людей, начать прогревать ее до рабочей температуры.

Регулирование температуры системой основано на показаниях датчиков температуры и присутствия людей, заданных величинах рабочей температуры в каждой комнате и недельного графика пребывания людей в доме, а также на показаниях *таймера*, который обеспечивает непрерывный поминутный отсчет текущего времени. Если температура в каком-либо помещении опускается или

поднимается на N градусов $(1 \le N \le 5)$ ниже или выше требуемой, то система формирует команду на изменение положения клапана в этой комнате.

Модельная система регулирования отопления подсчитывает также общий *расход топлива на обогрев* (в условных единицах), при условии, что на обогрев 1 кв. м комнаты за 1 минуту расходуется величина $P=C\times K$, где значение K соответствует положению клапана обогревателя в этой комнате: $K=0$ – закрыт, $K=2$ – полуоткрыт, $K=5$ – открыт полностью; а C – некоторая заранее заданная для каждой комнаты константа (зависит от площади батарей в этой комнате).

Цель моделирования – изучение зависимости величины расхода топлива от параметров M и N и недельного расписания занятости комнат дома.

Для проведения экспериментов необходимо программно эмулировать показания датчиков текущей температуры и присутствия людей в комнатах. Следует считать, что при закрытом клапане обогревателя температура в каждой комнате медленно падает (линейно по времени, коэффициент линейной зависимости определяется временем суток), при открытом – растет (по аналогичному закону), при полуоткрытом – сохраняется постоянной. При моделировании показаний датчика присутствия людей в комнате может использоваться вычисляемая тем или иным образом случайная величина – отклонение от известного расписания пребывания людей в рассматриваемой комнате дома (например, жильцы дома могут по каким-либо причинам раньше обычного уйти с утра из дома).

В ходе моделирования должны быть изображены: план дома, наличие в каждой комнате людей, положение клапанов обогревателя, а также указаны температура в каждой комнате, время суток и день недели, расход топлива на текущий момент. Кроме начального задания недельного расписания следует допустить также возможность его изменения в ходе экспериментов.

Модель контроля городской экологической обстановки

Требуется создать компьютерную модель слежения за *экологической ситуацией* в городе, где работает N $(5 \le N \le 12)$ промышленных предприятий, а также зарегистрировано K тысяч $(30 \le K \le 90)$ автомобилей. Экологическая обстановка в городе зависит от общего объема вредных *промышленных выбросов* действующих предприятий и *выхлопов автомобилей*, а также от погодных условий (дождь и ветер убыстряют рассасывание вредных веществ в атмосфере). Известны площадь города, местоположение предприятий и расстояния между ними, налоговые отчисления каждого предприятия в городскую казну, а также допустимый объем их выбросов в атмосферу.

Городской департамент экологии добивается улучшения экологической обстановки в городе несколькими способами. Он может уменьшить число автомобилей на дорогах города (и соответственно, суммарный их выхлоп), введя на определенный период *специальный режим движения* (например, по четным дням в городе могут ездить только автомобили с четными номерами, по нечетным – с нечетными номерами). Департамент может применять *штрафные санкции* к предприятиям, превысившим допустимую норму выбросов вредных веществ в атмосферу. Санкции включают *денежные штрафы* и полную или частичную *приостановку работы* предприятия на один или несколько дней.

Выплаченные штрафы пополняют денежный фонд города, из которого предприятиям могут субсидироваться средства на установку *очистных фильтров*. Установка одного фильтра требует определенной суммы (например, 30 тыс. у.е.) и выполняется за 7-10 дней. Фильтр уменьшает объем выброса на 7%. Денежный фонд пополняется также за счет налоговых отчислений работающих предприятий (измеряется в у.е.).

Цель компьютерного моделирования – исследовать влияние различных штрафных санкций, ограничений и субсидий на улучшение экологической обстановки в городе. В изменяемые параметры модели целесообразно включить числа *K* и *N*, начальный размер денежного фонда, значимые характеристики каждого предприятия (налоговые отчисления, допустимый выброс). Период моделирования – несколько месяцев, шаг моделирования – один день.

Каждый шаг включает следующие действия:

1. Замеры текущих объемов выбросов для всех работающих предприятий; расчет (замер) концентрации вредных веществ в фиксированных точках города и сравнение ее с установленной для города допустимой величиной;
2. Расчет текущей суммы денежного фонда, при этом учитывается его остаток от ранее предоставленных субсидий и прирост за счет штрафов и налоговых отчислений работающих в рассматриваемый день предприятий;
3. Принятие решения о штрафных санкциях (штрафах и приостановке работы) по отношению к предприятиям, превысивших допустимую норму выбросов;
4. Принятие решения о введении специального режима движения автомобилей и срока его действия;
5. Принятие решения о субсидировании (на установку фильтров) предприятий, часто нарушающих допустимые объемы выбросов.

Операции 1 и 2 модели выполняются автоматически; операции 3-5 – автоматически или пользователем системы моделирования.

При расчете концентрации вредных веществ в атмосфере в рассматриваемый день следует учитывать *остаточную концентрацию* за прошлый день и суммарное загрязнение атмосферы за текущий день, которое дают выхлопы автомобилей и выбросы работающих в этот день предприятий. Загрязнение от автомобилей распределяется равномерно над городом, а от предприятий – равномерно падает с увеличением расстояния от него. Следует учесть также колебания дневного выброса вредных веществ на каждом предприятии – их можно рассматривать как случайную величину из некоторого диапазона. Считать также, что каждый день в городе двигаются только 75% от общего числа автомобилей, которым разрешено движение, и известен средний дневной объем выхлопа двигающегося автомобиля. Остаточная концентрация рассчитывается как часть от концентрации прошлого дня, причем эта часть тем меньше, чем больше сила ветра и дождя (погодные условия также можно моделировать статистически).

Визуализация экологической обстановки должна включать изображение карты города, показ местоположения каждого предприятия и точек, где производятся замеры концентрации вредных веществ, а также разную расцветку карты в зависимости от степени загрязнения атмосферы. В ходе моделирования

пользователю системы должна быть доступна информация об экологической обстановке за прошедшие дни, о принятых штрафных санкциях, выделенных субсидиях и введенных ограничениях движения транспорта.

Моделирование распространения вирусного заболевания

Необходимо создать компьютерную модель распространения вирусного заболевания в стране из N *(*$5 \le N \le 10$*) городов трех типов*, различающихся численностью населения (мегаполисов, средних городов и поселков городского типа). Распространение вируса в городе зависит от нескольких факторов:

- численности населения рассматриваемого города и его типа;
- *насыщенности* в нем внутригородского и междугороднего *транспортного сообщения*;
- процента *заболевших людей* в городе;
- процента населения, у которого сделаны *профилактические прививки*;
- сезона года (заболеваемость растет с сентября, а с марта она уменьшается).

Цель моделирования – выявление стратегий проведения прививок, которые позволяют минимизировать число заболевших вирусным заболеванием и избежать эпидемии в крупных городах (порог эпидемии – 45% заболевших). Период моделирования – M месяцев $(6 \le M \le 24)$, шаг моделирования – одна неделя. В параметры модели следует включить числа M и N, месяц начала моделирования (например, сентябрь), численность населения городов страны, насыщенность транспортного сообщения в каждом городе (можно задать ее как величину в интервале от 0 до 1), начальный процент заболевших людей и сделанных прививок в каждом городе, а также стоимость одной прививки и первоначальный размер денежного фонда страны, предназначенного для прививок (измеряются в у.е.).

На каждом шаге моделирования производятся следующие действия:

1. Принимается решение, в каких городах и в каком количестве сделать профилактические прививки (их количество должно допускаться текущим состоянием денежного фонда); профилактические прививки не делаются заболевшим, а сделанная прививка начинает действовать через три недели.
2. Пересчитывается процент заболевших в каждом городе и процент заболевших в стране, при этом, во-первых, учитывается выздоровление определенной части заболевших (считается, что заболевание длится 2 недели у 60% больных, у 15% – 3 недели, а у остальных – 1 неделя); во-вторых, учитывается распространение вируса и заболевание новых людей – их процент определяется всеми вышеперечисленными факторами.
3. Высчитывается текущий размер денежного фонда, при этом учитывается его убыль от сделанных (на этом шаге моделирования) прививок и выплат по временной нетрудоспособности для болеющего населения, а также прирост фонда за счет налоговых отчислений здоровой и работающей части населения городов (считается, что работоспособное население составляет 65% от всего населения города).

Шаги 2 и 3 выполняются автоматически, а шаг 1 – автоматически или пользователем системы моделирования. На шаге 2 процент заболевших людей

определяется как случайная величина, закон распределения которой зависит от всех вышеперечисленных факторов. В частности, чем больше население города и насыщеннее его транспортное сообщение, тем выше процент заболевающих в нем при прочих равных условиях; а чем больше сделано профилактических прививок, тем меньше заболевающих.

Визуализация процесса распространения вируса должна предусматривать изображение городов страны в виде кругов (размер круга зависит от численности населения города), показ текущей заболеваемости (на каждом круге определенным цветом высвечивается часть, пропорциональная числу заболевшего населения), а также особую расцветку городов, в которых началась эпидемия. На каждом шаге моделирования должна быть также доступна более подробная информация о заболеваемости в каждом городе (число привитых, заболевших, здоровых людей и др.).

Графические редакторы

Специализированный графический редактор

Графический экранный редактор позволяет оперировать *плоскими фигурами N* типов *(1≤ N ≤5)*, точнее, их изображениями. Назначение редактора – формирование нужной пользователю *конфигурации фигур* (изображений) на основе допустимых *операций* с ними.

Пользователь экранного редактора должен иметь возможность:

- добавлять и удалять фигуры разных типов;
- изменять местоположение фигур и поворачивать их;
- *группировать* несколько фигур (изображений) в более крупную единицу (изображение) – с тем, чтобы затем оперировать всей группой (перемещать, поворачивать, удалять ее);
- определять (вычислять) некоторые характеристики получившейся конфигурации (зависящие от специализации редактора);
- запоминать получившуюся конфигурацию (изображение) на дисковом файле и считывать ранее сохраненные конфигурации из файла в рабочее окно;

Требуется, чтобы указанные действия пользователь мог производить в произвольном, удобном для него порядке.

Визуализация формируемой пользователем конфигурации должна предусматривать показ фигур и их групп, а при необходимости – увеличение размера изображений и скроллинг. Вычисленные характеристики сформированной конфигурации (изображения) должны выдаваться по запросу пользователя. Возможны следующие специализации графического редактора:

- ✓ Построение различных электрических цепей (электрических схем) постоянного или переменного тока из составных элементов – лампочек, замыкающих ключей, резисторов, конденсаторов, источников тока и др. Редактор позволяет задавать внутренние характеристики указанных элементов (например, емкость конденсатора, сопротивление лампочки) и вычислять конфигурационные характеристики цепи (к примеру, число разветвлений тока).

- ✓ Расстановка на заданной схеме дома, состоящего из нескольких комнат, предметов мебели нескольких видов (стулья, столы и проч.). Редактор по запросу пользователя вычисляет и показывает расстояния между предметами мебели и между предметами и стенками комнат. Можно предусмотреть предварительное задание пользователем схемы дома (например, мышью указываются углы комнат, которые потом соединяются, отмечаются также двери и окна).

Модельная система укладки плитки

Рассматривается задача укладки *плитки*, имеющей разный цвет и геометрическую форму: квадрат, правильный треугольник, шестиугольник, восьмиугольника. Основная функция создаваемой системы – помощь пользователю в создании на поле (экране компьютера) нужного разноцветного *орнамента* (узора) плитки, при ее плотной укладке.

Пользователь системы должен иметь возможность:

- добавлять в нужное место поля плитку указанного цвета и геометрической формы; изменять местоположение или удалять уже уложенную плитку;
- *группировать* несколько плиток – с тем, чтобы затем оперировать целиком полученной группой (перемещать, поворачивать, удалять, изменять цвет);
- выделив некоторую плитку или группу плиток, *размножить* ее на поле (при этом система циклически повторяет на поле выделенную плитку/группу);
- запоминать получившийся орнамент на дисковом файле и считывать ранее сохраненный орнамент из файла в рабочее окно;
- открыть несколько окон с разными орнаментами и создавать новый орнамент путем копирования фрагментов других.

Требуется, чтобы указанные действия пользователь мог производить в произвольном, удобном для него порядке. При необходимости следует предусмотреть скроллинг отображаемого орнамента.

3. Методические указания

Моделирование процессов и событий во времени

При моделировании и визуализации процессов (событий), изменяющихся (происходящих) во времени обычно применяется *метод пошаговой фиксации* (пересчета) всех характеристик рассматриваемого процесса или события. При этом выбирается шаг – отрезок времени, существенный для этого процесса и измеряемый в тех или иных единицах (например, в часах или минутах), через который пересчитываются изменяемые характеристики процесса и производятся необходимые изменения в его изображении на экране компьютера (в ряде задач целесообразно выполнять полную перерисовку изображения).

Необходимое пошаговое изменение времени реализует программный цикл, шаг которого соответствует выбранному отрезку времени моделируемого процесса. В простейшем случае шаг цикла не связан непосредственно с реальным временем. Как следствие, визуализация изменений процесса (событий)

при методе пошаговой фиксации может быть слишком быстрой или медленной для пользователя системы (зависит от производительности компьютера). Поэтому более гибкое решение – выполнять шаг пересчета характеристик моделируемого процесса по *системному таймеру*, увязав тем самым отрезок реального времени (измеряемый в мсек) с отрезком времени этого процесса/события. В таком случае можно опытным путем подобрать шаг, приемлемый для визуализации процесса, или даже включить этот шаг в параметры моделируемого процесса, позволив тем самым пользователю устанавливать удобный для него темп визуализации процесса.

Проектирование пользовательского интерфейса

Пользовательский интерфейс обычно разрабатывается средствами визуального программирования, предоставляемыми большинством современных интегрированных сред разработки (*IDE*). Важно, что перед непосредственным программированием интерфейса на базе этих средств необходимо выполнить *общее проектирование пользовательского интерфейса*, в ходе которого определяется количество диалоговых окон, их назначение и взаимодействие. Для каждого окна подбираются входящие в его состав *элементы управления* (кнопки, текстовые поля, списки и др.). По итогам проектирования составляется *эскиз пользовательского интерфейса*.

В большинстве вариантов задания достаточно двух диалоговых окон: для задания параметров моделирования и для визуализации моделируемого процесса. Если число управляющих элементов каждого окна не очень велико, эти окна можно объединить в одно, сгруппировав логически связанные управляющие элементы в отдельные *функциональные блоки*, а также сделав элементы первого окна *неактивными* после задания параметров моделирования. В ряде вариантов целесообразно завести отдельное диалоговое окно для вывода итоговой статистики.

Главное требование к пользовательскому интерфейсу – он должен быть понятен пользователю, не знакомому с деталями реализации системы и особенностями моделируемого процесса. Поэтому важны как названия элементов управления (кнопок, полей, списков), так и их группировка и расположение внутри соответствующего окна. Следует предусмотреть также возможность выхода из системы в любой момент ее работы.

Объектный анализ и проектирование системы

Объектный анализ – начальная задача объектно-ориентированного проектирования, в ходе которой происходит выявление (идентификация) объектов и классов программной системы.

Объект – сущность, имеющая четко определенное функциональное назначение в рассматриваемой задаче, например: “небесное тело”, “орбита” (в модели Солнечной системы). Для выявления объектов полезно составить так называемый *словарь проблемной области* (задачи), в который входят все основные понятия (ключевые абстракции), как процедурные (глаголы), так и непроцедурные (существительные), например: “планета”, “вращаться” и т.п.

Объект характеризуется *состоянием* и *поведением*. Состояние объекта определяется перечнем (обычно статическим) всех *атрибутов* (свойств) данного

объекта и текущими (динамическими) значениями каждого из этих свойств. Поведение объекта характеризуется выполняемыми над ним *операциями* и его состоянием (некоторые операции имеют побочное действие – они изменяют состояние). Например, для объекта “небесное тело” атрибутами являются его координаты, размер, масса, а операциями – “изобразить”, “переместить” и др.

Структура (атрибуты) и операции схожих объектов определяет общий для них *класс* (тип). Объекты – *экземпляры* соответствующих классов. Обычно операции объявляются как методы класса, к объектам которого они относятся.

Поскольку в процессе работы одни объекты связаны с другими, эти связи моделируются как отношения между соответствующими классами. Сначала фиксируются общие семантические связи (зависимости) классов – *ассоциации*. В ходе дальнейшего проектирования эти связи конкретизируются. Будем различать следующие виды отношений между двумя классами:

- ***наследование***: отношение *обобщения*, или общего и частного: например, “небесное тело” и “планета”; в общем случае *подклассом* наследуется структура и поведение более общего класса (*суперкласса*), в то же время подкласс обладает дополнительными/уточненными атрибутами и операциями.
- ***агрегация***: отношение *включения*, или целого и части между экземплярами классов: например: “Солнечная система” и “планета”; обычно объект-часть включается в состояние объекта-целого.
- ***использование***: связь между экземплярами классов, при которой один пользуется услугами другого (отношение *клиент-сервер*): например, использование операции.

Все перечисленные связи – направленные (в отличие от ассоциации).

Составление диаграмм и спецификаций

Выявленные связи классов необходимо зафиксировать в ***диаграмме классов***. Класс обычно обозначается прямоугольником, в котором записывается его имя (часто записываются также его атрибуты и операции). Связи классов изображаются соединяющими их линиями. Наследование графически изображается в виде стрелки с полым треугольником, идущей от подкласса к суперклассу, агрегация – в виде незакрашенного ромба у класса-агрегата, а использование – в виде пунктирной стрелки, идущей от клиента к серверу.

На диаграмме полезно указывать *мощность* (множественность) связей, определяющая, сколько экземпляров одного класса могут быть связаны с одним экземпляром другого класса. На практике важно различать три случая: *один к одному*, *один ко многим* (*1 : N*) и *многие ко многим* (*M : N*).

Нередко при проектировании классов необходимо выбирать между отношениями наследования, агрегации и использования, как при выявлении связи между “планетой” и “спутником” в модели Солнечной системы.

Отношения между классами *A* и *B* можно считать наследованием, если объект класса *A* может одновременно рассматриваться и как объект *B*, в ином случае лучше ввести другой вид связи.

При выявлении агрегации классов важно понять, существуют ли их экземпляры по отдельности и независимо друг от друга. *Композиция* – это

разновидность агрегации [4], при которой часть не существует без целого и объект-агрегат несет полную ответственность за создание и уничтожение своих частей (графически композиция изображается в виде закрашенного ромба у класса-агрегата). Распознать этот вид агрегации важно для правильного определения конструкторов и деструкторов классов, входящих в агрегацию.

В общем случае объект-целое и объект-часть существуют независимо друг от друга, и их можно создавать/уничтожать по отдельности. Заметим, что в программном коде агрегация превращается либо во *включение по значению*, при котором один объект является атрибутом другого, либо во *включение по ссылке*, при котором один объект содержит ссылку на другой.

При связи использования метод одного класса или вызывает операцию другого класса, или порождает экземпляр другого класса, или использует объект другого класса как параметр, или же возвращает его в качестве своего значения. Отношения использования слабее и менее ограничительны, в то же время агрегация часто предпочтительнее, т.к. позволяет скрыть части в целом.

Для более полного описания проектных решений, касающихся *ответственности классов*, необходимо составить *текстовые спецификации классов* – они должны содержать то, что не отображено в диаграмме, но удобнее записать в текстовом виде. Спецификация класса есть по сути описание его *интерфейса*, т.е. его видимых извне операций (методов). Текстовые спецификации могут быть записаны на используемом объектно-ориентированном языке программирования с добавлением *комментариев*, содержащих дополнительную информацию об атрибутах и операциях классов (например: статический, виртуальный, чисто виртуальный и т.п.).

Диаграмма классов должна быть дополнена ***диаграммой объектов***, показывающей взаимодействие основных объектов программной системы. Классы статичны, т.е. их содержание определено в процессе компиляции программного кода, и диаграмма классов описывает структуру программной системы. Напротив, объекты динамичны – они создаются и уничтожаются в процессе выполнения программы, и диаграмма объектов характеризует ее поведение, т.е. совместное функционирование объектов.

Объекты на диаграмме изображаются прямоугольниками с закругленными углами, внутри которых указывается имя объекта (а нередко и его класс). Связи объектов на диаграмме показывают пути передачи сообщений между ними (вызовов операций), направление сообщения (вызова) обозначается стрелкой, указывающей на объект, который предоставляет операцию. Связи целесообразно помечать именем вызываемой операции.

Важно, что диаграммы классов и объектов должны быть согласованы: если объект *X* посылает объекту *Y* сообщение, то в соответствующей диаграмме классов должна быть надлежащая связь между классами этих объектов.

Отметим, что объектное проектирование – итеративный процесс. На практике очень часто невозможно завершить проектирование, не закончив программирование классов – таким образом, эти процессы выполняются последовательно и итеративно. В итоге *логический класс* превращается в *программный класс*, т.е. объявление класса в конкретном языке программирования.

Особенность всех описанных вариантов задания практикума в том, что они допускают несколько различных способов проектирования, равно приемлемых и отображаемых в разных итоговых диаграммах и спецификациях.

Проектирование файловой структуры системы

В ходе проектирования и программирования решается также вопрос о *распределении классов по модулям (файлам)* программной системы.

Ряд объектных языков и интегрированных сред разработки приложений навязывают определенный способ разбиения программного кода на модули-файлы. В тех же случаях, когда остается свобода выбора, разумно сгруппировать в отдельные файлы внутренне (логически) связанные классы, например, выделить в отдельные модули: основную расчетную часть (математическую модель процесса или явления), процедуры визуализации моделируемого процесса или явления, процедуры организации интерфейса с пользователем.

В больших проектах следует избегать вариантов модульной структуры, при которых каждый класс помещен в отдельном файле. В то же время нежелательна и другая крайность, когда все спроектированные классы объединяются в одном файле – такое решение допустимо для небольших задач.

Отчет о выполнении задания практикума

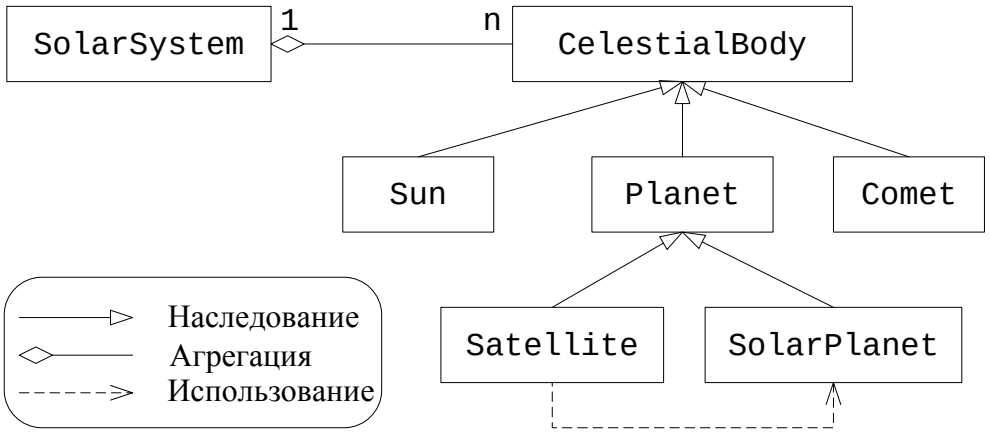
Текст отчета должен включать следующие разделы:

1. Уточнение постановки задачи для выбранного варианта задания, при необходимости – описание математической модели процесса или явления.

   Например: Демонстрационная модель солнечной системы.
   Смоделировать *на основе законов Кеплера* движение планет Солнечной системы и *пролет кометы* через нее с заданной скоростью.

2. Диаграмма классов, показывающая выделенные классы и связи между ними.

   Например:



3. Текстовые спецификации интерфейса основных классов системы.

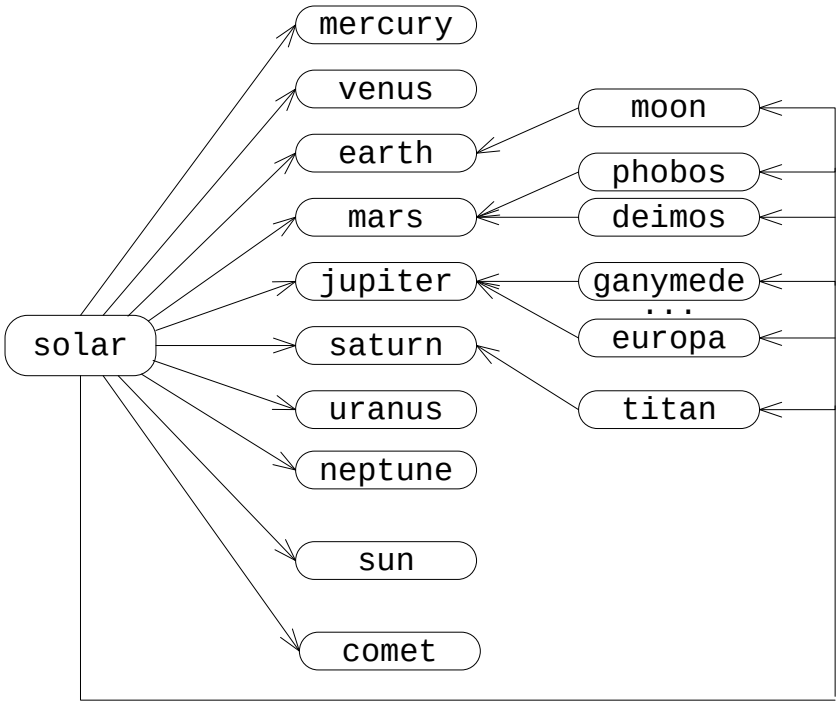
   Например, для одного из классов:

```
//Интерфейс класса, представляющего Солнечную систему
class CSolarSystem
{  public:
```

```
        //итераторы для объектов системы
        typedef TBodies::iterator TIterator;
        TIterator begin();
        TIterator end();
        //конструктор и деструктор
        CSolarSystem(void);
        ~CSolarSystem(void);
        //пересчет положения объектов системы
        void nextFrame(double days_per_frame);
        //добавление тела p в систему
        void addBody(CCelestialBody * p);
        //удаление тела p из системы
        void removeBody(CCelestialBody * p);
};
```

4. Диаграмма объектов, показывающая основные объекты и их связи.

   Например, для модели Солнечной системы (для наглядности на стрелках не указываются названия вызываемых методов):



5. Инструментальные средства, использованные при выполнении задания: язык программирования, интегрированная среда, библиотеки.

   Например: Язык разработки – C++
   Среда разработки – Microsoft Visual Studio .NET 2003
   Используемые библиотеки – OpenGL, glut.

6. Описание файловой структуры системы: перечень всех файлов программы с указанием классов, описанных в каждом из них.

   Например:

   `solar.cpp` – главный файл, интерфейс;

`Planets.h` и `Planets.cpp` – объявление и описание классов `CPlanets`, `CSolarPlanet` и `CSatellite`;
`Comet.h` и `Comet.cpp` – объявление и описание класса `CComet`;
`Sun.h` – описание класса `CSun`;
`CelestialBody.h` – объявление класса `CCelestialBody`;
`CelestialBody.cpp` – описание класса `CCelestialBody`;
`SolarSystem.h` – объявление класса `CSolarSystem`;
`SolarSystem.cpp` – описание класса `CSolarSystem`.

7. Пользовательский интерфейс: вид *диалоговых окон* для ввода параметров моделирования и для визуализации моделируемого процесса или явления; описание функций основных элементов управления этих окон.

   Например: клавиши управления визуализацией Солнечной системы:

   W – увеличение масштаба изображения;
   S – уменьшение масштаба изображения;
   A – движение налево, *D* – движение направо области просмотра;
   Мышью при зажатой левой кнопке – повороты тел;
   + и **-** – увеличение и уменьшение скорости движения тел;
   O – отображение/скрытие орбит;
   N – отображение/скрытие названий тел Солнечной системы;
   1..9 – инициирует пролет кометы со скоростью от 1 до 9 скоростей Земли в точке восхождения;
   E – выход из системы.

4. Литература

1. Буч Г. Объектно-ориентированное проектирование с примерами применения. – М.: Конкорд, 1992.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. – М.: Бином, 1998.
3. Буч Г., Максимчук Р., Энгл М. и др. Объектно-ориентированный анализ и проектирование с примерами приложений. – М.: Изд. дом «Вильямс», 2008.
4. Буч Г., Якобсон И., Рамбо Дж. UML. Классика CS. 2-е изд. - СПб.: Питер, 2006.
5. Гамма Э., Хэлм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.: Питер, 2010.
6. Гультяев А.К., Машин В.А. Проектирование и дизайн пользовательского интерфейса. – СПб.: КОРОНА, 2000.
7. Рамбо Дж., Блаха М. UML 2.0. Объектно-ориентированное моделирование и разработка. 2-е изд. – СПб.: Питер, 2006.