



**Министерство образования и науки Украины
Севастопольский национальный технический университет**

**Методические указания
к курсовому проектированию по дисциплине
«Объектно-ориентированное программирование»**

**для студентов дневной и заочной формы обучения
направления 050101 – «Компьютерные науки»**

Часть 1

**Севастополь
2011**

УДК 004.42 (075.8)

Методические указания к курсовому проектированию по дисциплине «Объектно-ориентированное программирование» для студентов дневной и заочной формы обучения направления 050101 — «Компьютерные науки» / СевНТУ; сост. **Ю.В. Доронина, Т.И. Сметанина, А.К. Забаштанский.** — Севастополь: Изд-во СевНТУ, 2011. — 24 с.

Цель указаний: оказать методическую помощь студентам дневной и заочной формы обучения направления 050101 — «Компьютерные науки» в выполнении курсовой работы по дисциплине «Объектно-ориентированное программирование».

Методические указания составлены в соответствии с требованиями программы дисциплины «Объектно-ориентированное программирование» для студентов дневной и заочной формы обучения направления 050101 — «Компьютерные науки» и утверждены на заседании кафедры Информационных систем протоколом № 7 от 23 февраля 2011 г.

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензент: Кожаев Е.А., канд. техн. наук, доцент кафедры кибернетики и вычислительной техники.

СОДЕРЖАНИЕ

1. Цель и задачи курсового проектирования	4
2. Тематика курсового проектирования	4
3. Выбор варианта задания.....	5
4. Порядок выполнения курсовой работы	5
5. Краткая характеристика основных этапов разработки программы	6
6. Разработка программных документов.....	7
6.1. Единая система программной документации.....	7
6.2. Требования к содержанию пояснительной записки и правила оформле- ния.....	8
7. Порядок защиты курсовой работы	10
Библиографический список	11
Приложение А. (обязательное) Варианты заданий к курсовой работе	12
Приложение Б. (справочное) Пример оформления титульного листа	20
Приложение В. (обязательное) Бланк задания на курсовое проектирование....	21

1. ЦЕЛИ И ЗАДАЧИ КУРСОВОГО ПРОЕКТИРОВАНИЯ

Целями курсового проектирования является изучение современного подхода к программированию на основе объектно-ориентированной технологии, приобретение навыков написания программ на языке с поддержкой ООП (C++, Java и Object Pascal) на примере написания программы согласно варианту задания.

Для достижения цели на разных этапах курсового проектирования должны быть решены следующие **задачи**:

- выбор варианта задания и языка реализации, детализация поставки задачи;
- абстрагирование, разработка классов и их иерархии;
- написание текста (кодирование) разработанных классов на выбранном языке;
- разработка тестовых примеров;
- тестирование и отладка программы;
- разработка программных документов в соответствии с действующими стандартами.

По окончании курса студенты должны овладеть объектно-ориентированным подходом решения практических задач, овладеть инструментами, сопровождающими разработку программных систем с использованием этого подхода.

2. ТЕМАТИКА КУРСОВОГО ПРОЕКТИРОВАНИЯ

Объектный подход — один из современных методов реализации программных систем (ПС). Наиболее показательна эффективность применения объектного подхода для больших ПС со сложным характером взаимодействия значительного количества элементов.

Тематика курсовой работы продиктована популярностью динамических моделей реального времени. В качестве примера можно привести системы моделирования работы электронных схем (язык *VHDL*), симуляция работы динамической системы пакетом *Simulink*. По схожему принципу организуется работа стратегических игр, которые развивают у пользователя стратегическое мышление, обдумывание действий на несколько шагов вперед, предвидение и тактику действий.

Рассматриваемые модели учитывают влияние большого количества факторов на многие элементы системы, и разработка таких систем требует от проектировщика продуманных шагов и взвешенных решений.

Таким образом, основная задача курсовой работы — детальное проектирование и программная реализация системы, моделирующей процессы создания и взаимодействия группы объектов.

3. ВЫБОР ВАРИАНТА ЗАДАНИЯ

Выбор варианта осуществляется по согласованию с преподавателем.

Все варианты разбиты на 4 группы (Приложение А):

- в первой группе заданий необходимо на базе объектного подхода спроектировать и разработать модель игры по заданному плану-варианту;
- во второй группе заданий предлагается разработать на базе теории систем массового обслуживания модель по заданному плану-варианту;

- в третьей группе заданий необходимо разработать систему сбора статистической информации по заданному плану-варианту;
- в четвертой группе заданий необходимо на базе объектного подхода спроектировать и разработать модель сложной системы по заданному плану-варианту.

Примечание: при выполнении работы возможно дополнение или незначительное изменение исходных условий постановки задачи в целях более качественной реализации проекта. По сути задания представляют собой лишь канву тех или иных описаний поведения одиночных объектов или их групп. Внесение изменений согласовывается с преподавателем.

4. ПОРЯДОК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Курсовая работа (КР) по дисциплине ООП выполняется в течение семестра. Защита курсового проекта происходит в комиссии на 13-14 неделе семестра по утвержденному графику. Чтобы получить допуск к защите, необходимо предъявить следующие документы и программные продукты:

- программу;
- пояснительную записку (ПЗ);
- графические материалы (плакат формата А1).

Общий порядок работы над КР определяется календарным планом, приведенным в бланке задания на курсовое проектирование.

Курсовая работа может выполняться индивидуально или группой студентов 2-3 человека, в зависимости от сложности задания. В качестве языка программирования выбираются языки Object Pascal, Си++, Java и др. (выбор языка программирования ограничивается требованием **абсолютно полной** поддержки возможностей объектного подхода).

Выполнение работы разбивается на несколько этапов:

1. Объектно-ориентированный анализ. Описание объектов, классов и их атрибутов. Описание цели игры, стратегии. Построение информационной модели (ИМ) как представления предметной области разработки в статике. Построение жизненного цикла системы (программы и объекта) (ЖЦ) и диаграммы переходов состояний (ДПС) как динамического представления системы. Построение диаграммы потоков данных и действий (ДПДД), как представления внутренней организации программной системы.

2. Объектно-ориентированное проектирование. Разработка иерархии классов в терминах программной среды.

3. Объектно-ориентированное программирование. Описание входных и выходных данных, методов, функций (процедур).

4. Пользовательский интерфейс, описание работы программы.

5. Анализ программы. Критерии качества ПС. Краткая характеристика содержания каждого из этапов изложена в следующем разделе.

Во время защиты КР студент должен сделать доклад, продемонстрировать работу программы на персональном компьютере, ответить на вопросы, относящиеся как к работе программы, так и к основным теоретическим положениям дисциплины ООП.

Курсовая работа оценивается по пятибалльной и рейтинговой шкалам (ECTS). Критериями, влияющими на результирующую оценку, являются:

- полнота реализации требований к программе;
- удобство программного интерфейса;
- стиль написания программного кода;
- тщательность тестирования программы;
- качество оформления пояснительной записки и графических материалов;
- полнота и правильность ответов на вопросы;
- соблюдение календарного плана выполнения курсового проектирования.

Критерии оценок:

- **отлично:** полностью разработанная ПС в соответствии с предложенным или самостоятельно разработанным (предварительно утвержденным ведущим преподавателем) заданием, положительные критерии качества ПС;

- **хорошо:** полностью разработанная ПС в соответствии с предложенным заданием или самостоятельно разработанным заданием (предварительно утвержденным ведущим преподавателем), но критерии качества ПС имеют недостаточный уровень;

- **удовлетворительно:** полностью спроектированная ПС в соответствии с предложенным заданием (исследована предметная область, построены иерархии классов, сделаны все необходимые проектные решения и предположения, полностью описан результат функционирования ПС, описаны пути обеспечения критериев качества ПС), но ПС реализована частично;

- **неудовлетворительно:** ПС спроектирована, но реализована частично, критерии качества ПС не соблюдены.

5. КРАТКАЯ ХАРАКТЕРИСТИКА ОСНОВНЫХ ЭТАПОВ РАЗРАБОТКИ ПРОГРАММЫ

Создание любой программы начинается с постановки задачи. Описание общих требований к программе и постановка задачи на курсовое проектирование изложены в разделах 2 и 3. На этом этапе также определяется среда, в которой будет выполняться программа: требования к аппаратуре, используемая операционная система и другое программное обеспечение.

Объектный подход предусматривает следующие этапы:

- *OOA (object oriented analysis)* — объектно-ориентированный анализ;
- *OOD (object oriented design)* — объектно-ориентированное проектирование;
- *OOP (object oriented programming)* — объектно-ориентированное программирование.

Рассмотрим кратко эти ключевые понятия (определения Г. Буча)[2]:

Объектно-ориентированный анализ — это методология, при которой требования к системе воспринимаются с точки зрения классов и объектов, выявленных в предметной области.

Объектно-ориентированное проектирование — это методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы.

В данном выше определении содержатся два важных свойства объектно-ориентированного проектирования:

- 1) оно основывается на объектно-ориентированной декомпозиции;
- 2) использует многообразие приемов представления моделей, отражающих логическую (классы и объекты) и физическую (модули и процессы) структуру системы, а также ее статические и динамические аспекты.

Объектно-ориентированное программирование — это методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

Программа будет объектно-ориентированной только при соблюдении трех требований:

- 1) *ООП использует в качестве базовых элементов объекты, а не алгоритмы;*
- 2) *каждый объект является экземпляром какого-либо определенного класса;*
- 3) *классы организованы иерархически.*

6. РАЗРАБОТКА ПРОГРАММНЫХ ДОКУМЕНТОВ

6.1. Единая система программной документации (ЕСПД)

Разработка документации на программу ответственный этап проектирования. От качества выполнения документации в значительной степени зависит не только эффективность использования программы, но и пригодность её к дальнейшему развитию и сопровождению.

Разработка программных документов выполняется в соответствии с ЕСПД, которая представляет собой комплекс государственных стандартов, устанавливающих взаимоувязанные правила разработки, оформления и обращения программ и программной документации. Перечень стандартов, входящих в ЕСПД, приведен в приложении Б. В соответствии с ЕСПД на разных стадиях проектирования программы составляются следующие программные документы (ГОСТ 19.101-77): техническое задание, пояснительная записка, текст программы, описание программы, программа и методика испытаний, различные эксплуатационные документы (формуляр, описание применения, руководство оператора, руководство программиста и др.). Общие требования к программным документам изложены в ГОСТ 19.105-78. *При выполнении курсового проекта разрабатывается пояснительная записка.*

6.2. Требования к содержанию пояснительной записки и правила оформления

В рамках настоящего *учебного* проекта совмещены несколько стадий проектирования (эскизное, техническое и рабочее проектирование). Поэтому разрабатываемая *пояснительная записка* к курсовому проекту будет объединять в себе несколько документов, требования к которым определяются ГОСТ 19.404-79 (требования собственно к пояснительной записке), ГОСТ 19.505-79 (требования к руководству оператора) и ГОСТ 19.401-78 (требования к тексту программы). ПЗ к курсовому проекту должна содержать следующие структурные элементы и разделы:

- титульный лист;
- техническое задание (на специальном бланке);
- аннотация;
- содержание;
- перечень сокращенных и условных обозначений (при необходимости);
- введение;
- список исполнителей (при необходимости);
- постановка задачи;
- проектное решение;
- программная реализация;
- выводы;
- перечень ссылок;
- приложения (обязательное приложение — текст программы).

Титульный лист пояснительной записки оформляют в соответствии с приложением В.

Техническое задание - документ на специальном бланке, в соответствии с которым выполняется курсовое проектирование. Образец заполнения бланка задания представлен в приложении Г. Бланк задания заполняется в начале проектирования и подписывается студентом и руководителем проекта.

В **аннотации** приводят сведения о назначении документа (в данном случае пояснительной записки) и краткое изложение его сути в виде 6 — 8 предложений. В **содержании** указываются наименования разделов и подразделов пояснительной записки (кроме титульного листа и технического задания), а также номера соответствующих страниц.

Перечень сокращений и условных обозначений содержит все используемые в тексте пояснительной записки малораспространенные условные обозначения, символы, единицы, сокращения и термины. Независимо от этого *при первом появлении этих элементов в тексте необходимо привести их расшифровку*. Элементы перечня должны быть упорядочены по алфавиту.

Во **введении** следует указать наименование программы, а также документы, на основании которых ведется разработка с указанием организации и даты утверждения. В рамках настоящего курсового проектирования документ – это техническое задание, организация – Севастопольский национальный технический университет. Наименование программы и дата утверждения, указанные в разделе «Введение», должны соответствовать значениям, указанным на листе технического задания. В этом разделе обосновывается **актуальность** решаемой задачи, указываются **цель** и перечисляются **задачи**, которые необходимо решить для достижения поставленной цели. Во введении также приводится краткое изложение последующих разделов пояснительной записки.

При формулировке цели следует указать наименование программы, важнейшие её характеристики и назначение. Рекомендованный объем введения — 1 страница.

Раздел **«Список исполнителей»** приводится в случае выполнения задания группой студентов. В разделе приводится список исполнителей с подробным указанием вклада каждого из них.

Раздел **«Постановка задачи»** должен содержать следующие подразделы:

- цель разработки (например: разработать программную систему на базе объектного подхода, реализующую заданные функции);
- описательная постановка задачи (описание стратегии работы системы, предполагаемого интерфейса, взаимодействия с пользователем);
- ограничения, условия выполнения, функционирования.

Рекомендуемый объем этого раздела 1 – 2 страницы.

Раздел «**Проектное решение (разработка объектной модели)**» должен содержать следующие подразделы:

- абстрагирование и выделение объектов;
- построение иерархии классов (структура);
- построение информационной модели;
- описание ЖЦ (описание жизненного цикла программы и жизненного цикла одного из объектов);
- ДПС (диаграмма переходов состояний);
- ДПДД (диаграмма потоков данных и действий).

Рекомендуемый объем этого раздела 8 ... 10 страниц.

Раздел «**Программная реализация**» должен содержать следующие подразделы:

- обоснование выбора языка (системы) программирования;
- общее описание взаимодействия программных модулей (структурная схема программы);
- описание реализации основных классов и их методов (продемонстрировать использование наследования, полиморфизма, инкапсуляции);
- интерфейс пользователя (подробное описание работы программы со снимками экрана);
- критерии качества программной системы.

Рекомендуемый объем раздела 15...20 страниц.

В **выводах** указываются основные и смежные отрасли использования спроектированной программы, делаются выводы о соответствии её техническому заданию и о путях дальнейшего усовершенствования.

Выводы формулируют в прошедшем времени («получено», «достигнуто», «разработано» и т.п.) в обратном порядке по отношению к порядку следования задач во введении, т.е. от решенных задач к цели. При этом следует не только констатировать факт решения задачи (например, «разработана схема алгоритма»), но дать краткую характеристику результату. Отдельным абзацем приводят фразу о факте достижения цели проекта, формулируют достигнутую цель и приводят рекомендации о дальнейшем использовании результатов проекта.

Рекомендуемый объем выводов — 1 страница.

В разделе «**Перечень ссылок**» необходимо поместить перечень научно-технических публикаций, нормативно-технических документов и других научно-технических материалов, на которые есть ссылки в основном тексте ПЗ.

В обязательном **приложении** приводится **текст программы с комментариями**. Текст программы содержит символическую запись программы на языке программирования с комментариями, отражающими структуру и назначение частей программы. Помимо текста программы, в приложение могут быть включены и другие документы, использованные при разработке.

Графический материал представляется в виде 1 плаката, на котором схематически изображается описание проектного решения: иерархия объектов, их взаимодействие, структура модулей программой системы.

Правила оформления пояснительной записки приведены в [8].

В случае выполнения курсовой работы группой студентов, пояснительная записка оформляется в единственном экземпляре, как и плакат (один на группу).

7. ПОРЯДОК ЗАЩИТЫ КУРСОВОЙ РАБОТЫ

Курсовая работа защищается перед ведущим преподавателем.

Порядок защиты следующий:

1. За три-пять дней до назначенного срока защиты сдается на проверку пояснительная записка, полностью оформленная и сшитая вместе с макетами плакатов или чертежей.

2. В назначенный срок разработчиками демонстрируется реализованный программный продукт в окончательном варианте (Защищаемая программная система должна содержать в обязательном порядке руководство пользователя, удобный интерфейс).

3. Наряду с демонстрацией разработанной программной системы студенты отвечают на вопросы комиссии, касающиеся как конкретного проекта и его реализации, так и объектного подхода в целом.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Белецкий Я. Энциклопедия языка Си / Я.Белецкий. — М.: Мир, 1992. — 687 с.
2. Буч Г. Объектно-ориентированное проектирование с примерами применения / Г.Буч. — М.: Конкорд, 1992. — 519 с.
3. Гради Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 2-е изд. / Г.Буч. — М.: "Издательство Бином", СПб:"Невский диалект", 1998. — 278 с.
4. Р. Вайнер. C++ изнутри / Р.Вайнер. Л.Пинсон. — Киев: НПИФ "ДиаСофт", 1993. — 304 с.
5. Айра Пол. Объектно-ориентированное программирование на C++ /Айра Пол. — М. : "Издательство Бином", СПб: "Невский диалект", 2001. — 464 с.
6. Г. Шилдт. Самоучитель C++. 3-издание / Г. Штлдт. — БХВ-Петербург, 2003. — 687 с.
7. Х. М. Дейтел, П. Дж. Дейтел. Как программировать на C++ (полное издание) / Х. М. Дейтел, П. Дж. Дейтел.; под ред. В.В.Тимофеева. — М.: Бином, 2008. — 1454 с.
8. Методические указания к курсовому проекту по дисциплине «Основы программирования и алгоритмические языки» для студ. дневной и заочной формы обучения направления 050101 — «Компьютерные науки» / СевНТУ ; сост. В. Н. Бондарев, Л. Ф. Давиденко, Т. И. Сметанина, В. Н. Хоролич. — Севастополь: Изд-во СевНТУ, 2009. — 76 с.

ПРИЛОЖЕНИЕ А

(обязательное)

Варианты заданий к курсовой работе

Таблица А.1 — Стратегические игры

№ варианта	Наименование	Функции системы
1	2	3
1.1	«Противоборство – <i>strategy</i> »	<p>В системе моделируется противоборство двух колоний. Колония состоит из строений и юнитов (живых существ). Юнит обладает набором характеристик: на сколько клеток игрового поля он «видит», уровень «здоровья» (когда ≤ 0 – юнит умирает), сколько «здоровья» он теряет во время боя за один удар и др.</p> <p>Юниты разделяются на два типа: строящие и воюющие. Строители наносят врагу незначительный урон и обладают низкой жизнестойкостью. Воины наносят более существенный урон и более защищены. Урон, наносимый воином, меняется случайно в незначительных пределах.</p> <p>Юнит обладает характерным поведением: строитель – если видит врага, – убегает, воин вступает в бой и т.д.</p> <p>У строений есть свои характеристики: размер, количество ударов, после которых строение рухнет и т.д.</p> <p>Строения могут быть двух видов: дома юнитов и защитные укрепления. Каждый дом обеспечивает жизнедеятельность N юнитов. Если вместимости домов не хватает, новый юнит не может быть порожден.</p> <p>Если дом разрушен противоборствующей стороной, то юниты пострадавшей стороны начинают голодать. Дома можно строить без ограничений, на строительство дома нужно достаточно много времени. Юнит создается быстрее, количество ограничено вместительностью строений.</p> <p>Игра происходит на игровом поле. Юнит занимает одно место, здание – несколько. Юнит может перейти на соседнее место, если оно не занято зданием или другим юнитом.</p> <p>Игра может проходить в двух режимах: из некоторого начального положения без участия человека (на начало игры уже созданы все здания и юниты) и с участием человека (он управляет юнитами).</p>
1.2	«Противоборство – <i>shooter</i> »	<p>Моделируется противостояние игрока и компьютера.</p> <p>Суть: главный герой (ГГ) противостоит множеству вражеских юнитов.</p> <p>Характеристики ГГ: количество жизней, величина урона, защита, скорость и др.</p> <p>ГГ может собирать некоторые бонусы, выпадающие случай-</p>

Продолжение таблицы А.1.

1	2	3
		<p>ным образом или из побежденных противников. Возможна смена оружия, способностей или характеристик героя со временем или после побед над врагами.</p> <p>Юниты имеют характеристики, сходные с ГГ.</p> <p>Юниты делятся на несколько типов, каждый из которых отличается особыми параметрами и способностями.</p> <p>Цель игрока – сохранить ГГ от столкновений с юнитами или уничтожить наибольшее их количество. Цель юнитов – уничтожение ГГ.</p> <p>Игра завершается победой ГГ над всеми юнитами, управляемыми компьютером, либо достижением максимального количества оценочных баллов игры (опыт, очки, время и др), либо смертью ГГ.</p>
1.3	«Головоломка – <i>logic puzzle</i> »	<p>Существует множество объектов различного вида и типа, которые взаимодействуют между собой.</p> <p>Задача: перемещая и комбинируя определенным образом объекты, необходимо добиться заданного графически ожидаемого результата. Достижение результата представляется возможным в случае единственно правильного расположения и взаимодействия используемого множества объектов.</p>

Таблица А.2 — Системы массового обслуживания

№ варианта	Наименование	Функции системы
1	2	3
2.1	«Почтовые сервера»	<p>Рассматривается работа системы, состоящей из почтового сервера (S) и его клиентов (K). Сервер связан с клиентами каналом связи ограниченной пропускной способностью, и не может обслуживать одновременно больше ($x < n$) клиентов. Пропускная способность канала 19200 бод.</p> <p>В исходном состоянии сервер свободен, ни одно соединение с клиентом не установлено. После того, как клиент запросит соединение, сервер проверяет, может ли он предоставить соединение.</p> <p>Если соединение устанавливается, клиент передает серверу общее количество писем, их заголовки и размер. Получив заголовок очередного письма, сервер проверяет правильность адреса (на это уходит 2-3 секунды), если адрес правильный, начинается передача тела письма.</p> <p>В случае, если адрес неправильный – клиент получает сообщение об ошибке и соединение с ним закрывается.</p> <p>Время, необходимое для передачи тела письма зависит от пропускной способности канала и размера письма.</p>

Продолжение таблицы А.2.

1	2	3
		<p>Если канал свободен, то время составит: (размер тела в битах)/ (пропускная способность канала).</p> <p>Если к серверу подключены несколько клиентов, то пропускная способность делится между ними поровну.</p> <p>Если соединение с сервером не может быть установлено, клиент ждет некоторое время, после чего пытается повторить передачу.</p> <p>Письма классифицируются следующим образом: 1 вид – малые (до 2 Кб), 2 вид – средние (до 150 Кб), 3 вид – большие (все остальные).</p> <p>У каждого клиента есть уникальное распределение генерирующих писем, т.е. сколько процентов от общего количества генерируется первого вида, сколько второго, и т.д. (Например: клиент генерирует 80% писем малых, 10% - средних, 10%- больших). Общий поток писем характеризуется интенсивностью- количеством писем в единицу времени (Например, интенсивность 1.2 означает, что за 5 единиц времени придет 6 писем).</p> <p>Каждый клиент обладает своей интенсивностью. Необходимо промоделировать работу системы в течение задаваемого количества времени, определить следующие характеристики:</p> <ul style="list-style-type: none"> - процент занятости сервера, - количество отказов каждого клиента, - среднюю загрузженность канала, - среднее время передачи письма.
2.2	«Маршрутное такси»	<p>Смоделировать работу сети маршрутных такси на базе теории СМО. По маршруту движется не более N маршрутных такси (МТ). Каждое МТ имеет 18 посадочных мест.</p> <p>Водители МТ не нарушают ПДД и не берут дополнительных пассажиров. На маршруте имеется L остановок, на каждой из которых случайным образом генерируется некоторое количество пассажиров. Количество пассажиров, скопившихся на остановке, пропорционально времени ожидания.</p> <p>Для каждого пассажира случайным образом генерируется число остановок P, которое он хочет проехать $P \leq 2 * L / 3$.</p> <p>Время посадки/высадки 1 пассажира занимает 7 секунд. Время стоянки на остановке, если есть свободные места, 10 секунд, в противном случае МТ не останавливается.</p> <p>МТ движется по маршруту со средней скоростью 40 км/ч</p>

Продолжение таблицы А.2.

1	2	3
		<p>(скорость с учетом светофоров, пробок и т.п.). Расстояние между всеми остановками заранее известно.</p> <p>Необходимо промоделировать работу системы в течение задаваемого количества времени, определить следующие характеристики:</p> <ul style="list-style-type: none"> - процент занятости всех МТ на маршруте, - среднее число пропускаемых МТ остановок за 1 прохождение маршрута, - среднюю загруженность МТ, - среднее время проезда пассажира.

Таблица А.3 — Системы сбора статистической информации

№ варианта	Наименование	Функции системы
1	2	3
3.1	«Мониторинг погоды»	<p>Разработать систему сбора данных для мониторинга погоды.</p> <p>Система состоит из датчиков скорости и направления ветра, температуры, давления, влажности воздуха и др. Значения каждой величины в момент снятия показаний генерируется случайным образом из указанного диапазона с учетом общих закономерностей (например, изменение температуры в течение суток).</p> <p>Система должна снимать показания каждые 5 сек., усреднять по каждому часу, вычислять относительные изменения измеряемых величин, выводить результаты на экран, выводить данные за последние 24 часа, сохранять результаты работы в файле.</p> <p>Программа должна иметь: графический интерфейс и возможность указания начальных условий моделируемого процесса.</p> <p>Количество и тип каждого из датчиков, диапазон значений в которых изменяются измеряемые величины, интервал времени моделирования в часах и коэффициент соотношения реального времени со временем системы разрабатываются студентом и согласовываются с преподавателем.</p>
3.2	«Компьютерные сети»	<p>Сеть состоит из серверов (видео, аудио, файлообменников и др.), машин пользователей, оргтехники (принтер, сканер, копир, факс и др.). Сеть имеет некую топологию (звезда, кольцо, дерево).</p> <p>В системе существует несколько типов сообщений: запросы на обслуживание, ответы от обслуживающей техники, системные сообщения, сообщения включения / выключения</p>

1	2	3
		<p>серверов, запросы на получение данных и пакеты передаваемых данных и др.</p> <p>Машины пользователей генерируют запросы к серверам и оргтехнике. Запросы генерируются случайно с некоторой вероятностью их появления (вероятность появления запроса, а также распределение вероятностей появления по типам запросов разрабатываются студентом).</p> <p>Оргтехника способна обрабатывать только один запрос в единицу времени, на обработку запроса тратится N единиц времени. Существует некоторая вероятность отказа техники и некоторая вероятность отказа в обслуживании запроса. Результаты работы оргтехники сообщаются в ответном сообщении.</p> <p>Сервера могут обслуживать одновременно несколько запросов от пользователей. Запрос к серверу может быть «на получение информации», «на отправку информации», «на поиск информации» и др. Запросы «на информацию» инициализируют серию сообщений между сервером и клиентом.</p> <p>Если техника либо соединение вышли из строя, то их восстановление проводится строительно-ремонтной бригадой с заданной производительностью (то есть определенное время). Время ремонта определяется коэффициентом сложности оборудования.</p> <p>Программа должна отображать топологию сети, выводить статистические данные по запросам в сети и работе оборудования каждый час моделируемого времени системы, выводить во внешний файл средние значения за сутки.</p>

Таблица А.4 — Модели сложных систем

№ варианта	Наименование	Функции системы
1	2	3
4.1	«Океан»	<p>В первобытном океане существует две формы жизни: хищники и существа, питающиеся растительной пищей.</p> <p>Океан вырабатывает растительную пищу следующим образом: к количеству вещества на предыдущем шаге (x) добавляется некоторая часть (например, $0.5x$), если итоговая масса не превышает некоторого предела – порога (Y), т.к. океан не может вырастить больше некоторого количества растительной пищи на единицу поверхности.</p> <p>Потребление растительной пищи происходит следующим</p>

1	2	3
		<p>образом: если клетка поля занята «травоядным» существом, то оно потребляет 0.1х за один ход, 0.2х за второй, 0.3х за третий и т.д., т.к. в благоприятных условиях популяция травоядных возрастает.</p> <p>Размеры колонии травоядных увеличиваются на 1 ряд с каждой стороны за ход, если колония не голодает.</p> <p>Хищники, в отличие от травоядных, индивидуальны.</p> <p>Один хищник занимает целое поле и за 1 ход уничтожает половину его содержимого. Хищник находится в постоянном движении.</p> <p>В случае, если хищник в течение 5 ходов не «голодал» (получал не менее 50% от Y пищи), он раздваивается (делится).</p> <p>Два хищника не могут занимать одну клетку. В случае, если хищник не может перейти на клетку, не занятую другим хищником, один из них погибает (более «сытый»).</p> <p>Хищник живет 30 ходов, «голодающий» хищник в течение трёх ходов погибает.</p>
4.2	«Улей»	<p>Разработать программную систему, моделирующую жизнь улья. Существует 3 вида пчел — собиратели, строители и матка. Каждый из этих видов отвечает за выполнение определенных действий. У каждого вида пчел есть уровень жизни. Каждая пчела употребляет определенное количество меда в единицу времени, в зависимости от состояния в котором находится.</p> <p>Собиратели – отвечают за облет окружающей территории и сбор нектара. Когда Собиратель покидает улей, он перестает есть мед и начинает терять уровень жизненных сил. Интенсивность потери им жизненных сил зависит от того, какое количество нектара он несет с собой.</p> <p>Если он ничего не несет – 1 единица жизни за 1 ход. Каждая единица веса нектара увеличивает расход жизненных сил на 0.2 за 1 ход. Таким образом, если собиратель несет с собой 5 единиц нектара – он теряет $1 + 0.2 \cdot 5 = 2$ единицы жизненных сил за 1 ход. После возвращения в улей он начинает есть с интенсивностью в 4 раза больше, чем строитель в спокойном состоянии. После пополнения сил Собиратель может вновь отправиться за нектаром.</p> <p>Строители — отвечают за строительство улья и преобразование нектара в мед. Например, 1 строитель за 1 ход может преобразовать 10 единиц нектара в 7 единиц меда. Строительство дополнительных сот улья занимает определенное длительное время.</p>

1	2	3
		<p>Строитель может находиться в одном из 3-х состояний. Спокойное — малое потребление меда. Преобразование нектара в мед — удвоенное потребление меда. Строительство — утроенное потребление меда.</p> <p>Матка — отвечает за порождение новых членов колонии. При создании нового улья Матка порождается из Строителя (достаточно длительный процесс). В одном улье не может быть более одной Матки. В спокойном состоянии Матка потребляет столько же меда, сколько Строитель в спокойном состоянии. При создании новой пчелы она ест в 5 раз больше.</p> <p>Цветы, дающие нектар, появляются случайным образом на игровом поле. Длительность жизни цветка ограничена и не зависит от того, собирают ли с него нектар или нет.</p> <p>На обеспечение жизнедеятельности каждого насекомого колонии должно быть определенное количество сот в улье. Перед порождением нового члена колонии должна осуществляться проверка на возможность его жизнеобеспечения в улье.</p> <p>Если такой возможности нет, то улей должен быть достроен определенным количеством сот. Улей не может быть больше определенного размера.</p> <p>Если колония не голодает, может расти, но улей переполнен, то 30 процентов колонии покидают улей и образуют новый в другой части игрового поля. При этом они должны взять с собой определенное (достаточно большое) количество меда.</p> <p>Если в улье не хватает меда, то жители начинают голодать. Во время голодания скорость потери жизненных сил увеличивается. Работающие пчелы теряют уровень жизненных сил быстрее. Как только уровень жизненных сил пчелы становится ≤ 0, пчела погибает.</p>
4.3	«Муравейник»	<p>Разработать программную систему, моделирующую жизнь муравейника, основываясь на принципах жизни насекомых и иерархии особей: личинка, матка, рабочий муравей, воин.</p> <p>Матка порождает личинки, которые в свою очередь превращаются в рабочих муравьев. Матка может породить новую личинку только при достаточном количестве пищи и места в муравейнике.</p> <p>Воины охотятся, добывая еду муравейнику, и защищают муравейник от опасностей (атак других насекомых).</p>

Продолжение таблицы А.4.

1	2	3
		<p>Муравейник строится в виде центральной шахты и боковых ответвлений. Если длина бокового ответвления превышает определенные пределы, то должна быть построена очередная шахта. Строительство боковых тоннелей процесс, требующий не очень много еды. Данный процесс занимает много времени. На создание нового муравья требуется много времени и еды. Еда - это насекомые других видов (несколько разновидностей) отличающихся силой, размерами и количеством пищи, в которое они могут превратиться, если будут убиты муравьями. Соответственно и количество муравьев, необходимых для убийства насекомого и доставку его в муравейник разнится в зависимости от вида насекомого. Еда появляется случайным образом в произвольных точках игрового поля и может самостоятельно перемещаться по нему, в течение произвольного времени или пока не будет убита.</p> <p>Поглощение муравьями еды внутри муравейника и вне его аналогично потреблению пчелами меда в задании про улей.</p>

ПРИЛОЖЕНИЕ Б
(справочное)
Пример оформления титульного листа

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
Севастопольский национальный технический университет

Кафедра
информационных систем

КУРСОВАЯ РАБОТА ПО ДИСЦИПЛИНЕ
«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

ПРОГРАММА (название)

Пояснительная записка
UA.02070973.28315-01 81 14

Листов 30

ПРОВЕРИЛ
Доцент кафедры ИС
(подпись) Ю.В. Доронина
«___» _____ 2011

РАЗРАБОТАЛ
Студент группы И-31д
(подпись) И.И. Иванов
«___» _____ 2011

Севастополь
2011

ПРИЛОЖЕНИЕ В
(обязательное)
Бланк задания на курсовое проектирование

Форма № У 6.01

Севастопольский национальный технический университет
(название высшего учебного заведения)

Кафедра _____
Дисциплина _____
Специальность _____
Курс _____ Группа _____ Семестр _____

ЗАДАНИЕ
на курсовую работу студента

(фамилия, имя, отчество)

1. Тема работы _____

2. Срок сдачи студентом законченного проекта (работы) _____
3. Входные данные к работе _____

4. Содержание расчетно-пояснительной записки (перечень заданий, подлежащих разработке) _____

5. Перечень графического материала (с точным указанием обязательного) _____

6. Дата выдачи задания _____

КАЛЕНДАРНЫЙ ПЛАН

№ этапа	Название этапа курсовой работы	Срок выполне- ния этапа работы	Примечания

Студент _____
(подпись)

(фамилия, имя, отчество)

Руководитель _____
(подпись)

(фамилия, имя, отчество)

« _____ » _____ 20__ г.

Заказ № _____ от « _____ » _____ 2011 г. Тираж _____ экз.
Изд-во СевНТУ