

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

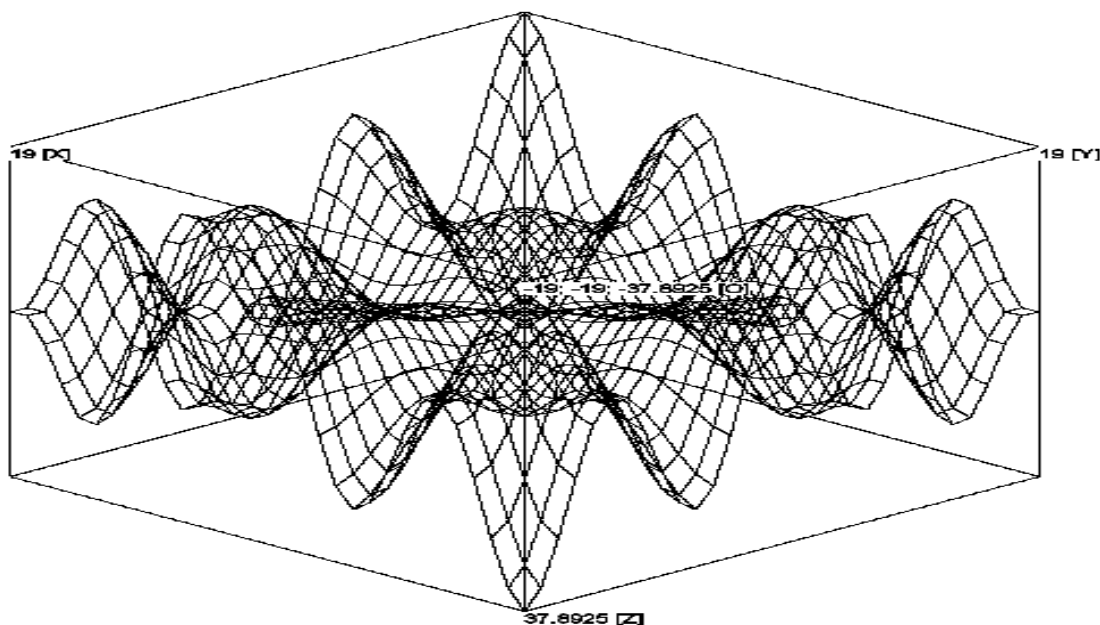
**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

Кафедра компьютерных систем в управлении и проектировании (КСУП)

С. Ю. Дорофеев

**Геометрическое моделирование в системах
автоматизированного проектирования**

Учебно-методическое пособие по курсовой работе



Томск – 2011

Рецензенты:

Дорофеев С.Ю.

Геометрическое моделирование в системах автоматизированного проектирования: Учебно-методическое пособие по курсовой работе – Томск : Кафедра Компьютерных систем в управлении и проектировании, 2011. – 16 с.

Приводятся рекомендации по выполнению курсовой работы. Учебно-методическое пособие предназначено для студентов любых форм обучения, в том числе и с применением дистанционных образовательных технологий.

© Дорофеев С. Ю., 2011

© Кафедра Компьютерных систем в
управлении и проектировании, 2011

1	Курсовое проектирование	5
1.1	Темы курсовых работ	5
1.2	Условия выполнения курсового проекта	7
1.2.1	Требования к техническому заданию	8
1.2.2	Проект системы	8
1.2.3	Диаграммы классов	9
1.2.4	Пояснительная записка	12
	Литература	16

1 Курсовое проектирование

Курсовое проектирование необходимо для закрепления материала, полученного в результате выполнения лабораторных работ, основанных на лекционном курсе.

1.1 Темы курсовых работ

Для выполнения курсовых работ необходимо объединиться в группы по 2-3 человека, на каждую группу выбирается одна из нижеперечисленных тем. Все работы предполагают плагинную архитектуру (plug-in — независимо компилируемый программный модуль, динамически подключаемый к основной программе, предназначенный для расширения и/или использования её возможностей. При этом, перекомпиляции основной программы не требуется.).

1. Написать редактор электрических схем. Все элементы должны быть представлены в виде плагинов. Обеспечить работу с номиналами элементов. Обеспечить многоступенчатую систему отмены-повтора (Undo/Redo).

2. Написать векторный графический редактор: квадрат, эллипс, растровое изображение и т.п. Все графические элементы должны быть представлены в виде плагинов. Обеспечить многоступенчатую систему отмены-повтора (Undo/Redo). Отображать список «слоёв» и позволять его редактировать (изменять их взаимное положение). Позволять редактировать выделенный в списке слоёв объект.

3. Реализовать построение трехмерного изображения графика произвольной функции от двух переменных $f(x, y)$ с возможностью изменения ракурса и масштаба просмотра. Функции для графика должны динамически загружаться из различных DLL. Обеспечить изменение точек на графике (построение графика должно производиться по некоторой дискретной сетке, должна быть возможность потянуть мышкой произвольную точку данной

сетки, изменяя одну из её координат). Обеспечить многоступенчатую систему отмены-повтора (Undo/Redo).

4. Написать растровый графический редактор – загружается растровое изображение, затем, с помощью инструментов (оформляемых в виде плагинов), производится его модификация. Инструменты используются пользователем посредством мыши. Все инструменты реализуются в виде квадратной кисти некоторого размера (размер задаётся пользователем), однако реализуют собственную функциональность.

Предусмотреть инструменты: рисование круглой кистью, затемнение, осветление, аэрограф (напыление), размытие и т.п. Обеспечить систему отмены-повтора (Undo/Redo).

5. Создание просмотрщика-редактора фотографий, обеспечивающего следующие функции: обрезать, поворот, перемасштабирование, надпись выбранного цвета, операции с гистограммой (яркость, контраст). Все операции должны обеспечиваться отдельными плагинами. Должна быть поддержка показа в режиме слайд-шоу. Обеспечить систему отмены-повтора (Undo/Redo).

6. Модуль печати, обеспечивающий работу с принтером, размещение на листе бумаги отдельных растровых изображений, поворот на 90 градусов, оптимальное заполнение листа копиями, масштабирование, отступы от краёв листа. Вся работа с устройством вывода (получение размеров листа, разрешения, имени устройства и т.п.) должна быть вынесена в отдельный модуль. Обеспечить вывод итогового растрового изображения на монитор и на принтер.

Перечисленные темы являются стандартными. Для получения нестандартных тем, связанных с разработками, выполняемыми на кафедре КСУП в данный момент, обратитесь к преподавателю.

Если работа, выполняемая в рамках ГПО, имеет отношение к сфере геометрического моделирования или САПР, тема ГПО может быть принята так

же в виде курсовой. Однако, это не исключает этапов составления технического задания, проекта, пояснительной записки и сдачи непосредственно программы.

Приветствуется использование технологий .NET, C#, DirectX, OpenGL, GDI+ и других современных инструментов. Также приветствуется создание кроссплатформенных систем.

1.2 Условия выполнения курсового проекта

Процесс разработки ПО (который часто называют *жизненным циклом* ПО) включает в себя следующие стадии:

1. Постановка задачи (формирование ТЗ).
2. Проект системы.
3. Реализация системы.
4. Испытание системы.
5. Сопровождение (модификация) системы.

На всех предыдущих этапах готовится соответствующая документация.

Все эти этапы необходимо выполнить в процессе выполнения курсового проекта.

Сроки, необходимые для успешной сдачи курсового проекта (за несвоевременную сдачу будут сниматься баллы).

- 2 недели — **Техническое задание** (не менее двух страниц).
- 1 Контрольная точка — **Проект системы** (не менее 10 страниц), пишется индивидуально — общими частями могут являться только описания интерфейсов взаимодействия компонентов.
- 2 Контрольная точка — **Реализация системы, испытания системы** — сдаётся программа, идеальная на взгляд студента, но не прошедшая контроль со стороны преподавателя. Программа должна быть с читаемыми комментариями и соответствовать международным стандартам (как минимум – венгерская нотация). Преподаватель подсказывает советы — как сделать программу

лучше, студент обязан устранить недостатки (стадия **сопровождения системы**).

- Зачётная неделя — исправленная программа + пояснительная записка (итоговое документирование).

Техническое задание, проект системы и итоговый отчёт принимаются только в твёрдой копии. Проект системы и итоговый отчёт (пояснительная записка) выполняются и защищаются индивидуально каждым студентом группы.

1.2.1 Требования к техническому заданию

Подробное техническое задание должно содержать следующие пункты:

- Требования к функциональности программы;
- Сфера применения;
- Требования к аппаратной и программной частям, необходимых для функционирования программной системы.

1.2.2 Проект системы

Проект системы является основным документом, описывающим будущую программную систему. Должен содержать следующие разделы:

Логические схемы:

- Иерархия классов (наследование)
- Диаграмма структуры (агрегация, композиция)
- Диаграмма использования (все оставшиеся связи)
- Блок-схемы типовых алгоритмов

Физические схемы:

- Разделение на модули

- Работа с файлами (использование файлов, их создание, форматы файлов)

Интерфейс пользователя – нарисованный в любом визуальном редакторе (например, Visual Studio или Microsoft Visio) интерфейс пользователя (GUI — Graphic User Interface): диалоги, панели, меню и.т.п.

Обосновать выбранный тип архитектуры каркаса приложения — (Dialog, SDI, MDI), привлекаемые технологии и алгоритмы. Обратить внимание на обоснование интерфейсов — почему именно такое расположение и именно таких элементов управления. Проектирование пользовательского интерфейса должно быть выполнено максимально тщательно.

1.2.3 Диаграммы классов

Визуализация структуры программы с помощью UML-диаграмм позволяет отобразить связи, которые существуют между объектами.

Взаимосвязь — это особый тип логических соединений в диаграммах классов и объектов. В UML представлены следующие виды отношений:

Агрегация (aggregation)

Агрегация — (“has a” association) встречается, когда один класс является коллекцией или контейнером других. Причём, время существования содержащихся классов не зависит от времени существования содержащего их класса. Если контейнер будет уничтожен, то его содержимое — нет.

Графически агрегация представляется пустым ромбиком на блоке класса и линией, идущей от этого ромбика к содержащемуся классу.

Пример: Список студентов у преподавателя. Студенты могут существовать независимо от того, занесены они в список или нет.

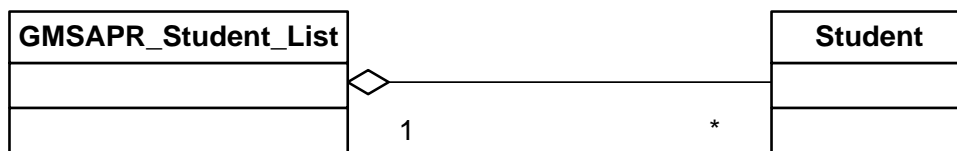


Рис. 1. Агрегация между двумя классами

Композиция (composition)

Композиция — более строгий вариант агрегации.

Композиция имеет жёсткую зависимость времени существования экземпляров класса контейнера и экземпляров содержащихся классов. Если контейнер будет уничтожен, то всё его содержимое будет также уничтожено.

Графически представляется как агрегация, но с закрашенным ромбиком.

Пример: Двигатель в автомобиле. Нет смысла создавать двигатель, если некуда его вставить (нет автомобиля), и наоборот. Поэтому, с точки зрения моделирования, объект «двигатель» логично встроить в объект «автомобиль».



Рис. 2. Связь «Композиция»

Обобщение (generalization)

Обобщение (“is a”) показывает, что один из двух связанных классов (подтип) является частной формой другого (супертипа или надтипа). Связь обобщения также известна как связь наследования.

Графически обобщение представляется линией с пустым треугольником у супертипа.

Пример: Животные — супертип млекопитающих, которые, в свою очередь, являются супертипом хищников, и так далее. Эта взаимосвязь легче всего описывается фразой «А — это Б» (хищники — это млекопитающие, млекопитающие — это животные).

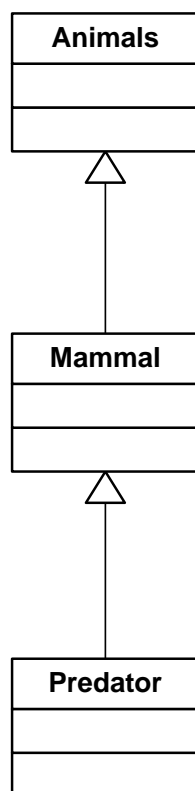


Рис. 3. Связь «Обобщение»

Зависимость (Dependency)

Зависимость существует между двумя элементами, если изменение одного влечёт за собой изменение другого.

Графически представляется пунктирной стрелкой, идущей от зависимого элемента к тому, от которого он зависит.

Например: зависимость типа «использует» («uses») — Человек учится в университете, то есть человек использует университет с целью получения знаний.

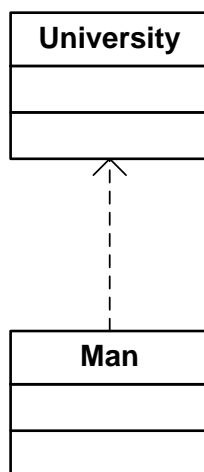


Рис. 4. Связь «Зависимость»

Для описания количества взаимосвязанных объектов одного типа используются *мультипликаторы*.

Нотация	Объяснение	Пример
0..1	Ноль или один экземпляр	У кошки может быть, а может и не быть хозяина.
1	Обязателен один экземпляр	У кошки должна быть мать, причём только одна
0..* или *	Ноль или более экземпляров	У кошки могут быть котята. А могут и не быть.
1..*	Один или более экземпляров	У кошки есть хотя бы одно место, где она спит. А может их несколько.

Примечание: Композиция должна иметь мультипликатор 0..1 или 1 (часть является частью только одного целого). В агрегации же может быть любой мультипликатор.

1.2.4 Пояснительная записка

Пояснительная записка к курсовой работе оформляется **согласно требованиям ГОСТ и СТП ТУСУР**.

Пояснительная записка (ПЗ) к курсовому проекту (работе) должна включать в себя следующие разделы:

1. **Титульный лист** – пример оформления титульного листа представлен в приложении А.

2. **Реферат** – содержит краткое описание (реферат) выполненной работы (2-3 предложения). Перечисляются ключевые слова, указывается количество страниц и приложений. Реферат размещают на отдельной странице. Заголовком служит слово "Реферат", написанное прописными буквами по центру страницы.

3. **Задание на проектирование** – содержит постановку задачи для курсового проекта.

4. **Содержание ПЗ** – нумерованный по страницам список разделов ПЗ. Нумерация страниц ПЗ – сквозная: титульный лист имеет первый номер.

5. **Введение** – содержит общую информацию по проекту: краткую характеристику решаемой задачи, назначение и возможности программы..

6. **Постановка и анализ задачи** – состояние проблемной области на момент начала работы: обзор существующих программ, их достоинства и недостатки. Анализируется поставленная задача, обосновывается выбранный путь решения, рассматриваются достоинства и недостатки выбранного решения.

7. **Описание алгоритма** – описываются математические модели, формулы, блок схемы типовых алгоритмов.

8. **Описание реализации** – логические и физические схемы программы, структура и типы специфических для решения задачи данных, диаграммы классов (с пояснениями), исходные коды с подробными комментариями, основные методы, их назначение и взаимосвязь, входные и выходные данные программы и т.п.

9. **Описание программы для пользователя** – общая инструкция по работе с программой, вид диалогов, описание меню.

10. **Тестирование программы** - описание тестовых задач и компьютерных систем, на которых происходило тестирование, результаты тестирования (качество работы и/или время работы).

11. **Заключение** – в данном разделе формулируем основные итоги работы: сопоставление желаемых и полученных результатов, встретившиеся проблемы, возможные направления для улучшения программы.

12. **Список литературы** – список источников, используемых при работе над проектом. Может содержать не только литературные источники, но и ссылки на различные ресурсы в сети ИНТЕРНЕТ. (не менее 5 наименований), ссылки по тексту обязательны.

13. **Приложения** – дополнительная информация, на усмотрение составителя документации.

Указания к оформлению ПЗ:

ПЗ пишется в редакторе MS Word или Open Office шрифтом Times New Roman, размером не менее 12 пунктов, на листе формата А4. Отступы: верхний 1,5, левый 3, нижний 2, правый 1 см. Нумерация страниц должна быть сквозной, первой страницей является титульный лист (номер страницы на титульном листе не ставится). Номер страницы проставляется вверху справа. Заголовки разделов пишутся прописными буквами по середине текста. Заголовки подразделов пишутся с абзаца строчными буквами, кроме первой прописной. В заголовке не допускаются переносы слов. Точка в конце заголовка не ставится.

Пример оформления списка литературы:

Список использованных источников

1. Иванов И.И., Петров П.П. Методы и алгоритмы компьютерной графики. – М.: Радио и связь, 2007. – 123с.
2. Сидоров С. Visual Studio для студентов-чайников. - СПб.: "Вильямс", 2004. – 234 с.

3. [http:// www.metka.cbuilder.ru](http://www.metka.cbuilder.ru)

Необходимо обратить внимание на следующее:

- 1) наличие нумерации и подписей к рисункам, нумерация и заголовки таблиц;
- 2) наличие заголовков приложений и их сквозная нумерация;
- 3) оформление списка литературы в соответствии с требованиями ГОСТа;
- 4) Обязательно: наличие в тексте ссылок на используемую литературу (в тексте ПЗ в квадратных скобках пишется номер источника).

Оценка за курсовой проект выставляется с учетом качества выполнения пояснительной записки.

Литература

1. Тригуб С.Н. Стандарты программирования на C++. // Пер. с англ. — М.: Издательский дом "Вильямс" ", 2005.— 224 с.
2. Скотт Мейерс. Наиболее эффективное использование C++. 35 новых рекомендаций по улучшению ваших программ и проектов // Пер. с англ. — СПб.: "ДМК Пресс" · 2000 . – 300 с.
3. Мартин Фаулер. Рефакторинг – улучшение существующего кода. // Пер. с англ. — СПб.:Символ-Плюс, 2003. — 432с.
4. Буч Гради. Объектно-ориентированное проектирование и анализ с примерами на C++. 2-е изд.
5. Гудман С., Хидетниемеи С. Введение в разработку и анализ алгоритмов. - М.: Мир, 1981. 368с.
6. Поляков А., Брусенцев В. Программирование графики: GDI+ и DirectX – СПб.: BHV, 2005. – 357с.
7. Ли. К. Основы САПР (CAD/CAM/CAE). – Спб.:«Питер», 2004. – 560с.
8. Аветисян, Амаякович. Автоматизация проектирования электротехнических систем и устройств: Учебное пособие для вузов/ Д. А. Аветисян. – М.: Высшая школа, 2005. – 510 с.: ил.
9. Порев В. Компьютерная графика: Учебное пособие/ В. Н. Порев. – СПб.: БХВ-Петербург, 2005. – 428 с.: ил.. – (Учебное пособие)