

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
"ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ УПРАВЛЕНИЯ"

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ КУРСОВЫХ ПРОЕКТОВ**

по дисциплине

**Алгоритмизация, языки и объектно-ориентированное
программирование**

для студентов специальности

Математические методы в экономике — 080116

Составитель: к.ф.-м.н., доцент К. Ю. Федоровский

МОСКВА – 2008

Тематика курсового проектирования по дисциплине "Алгоритмизация, языки и ООП" – это *"Разработка специального программного обеспечения"*.

Задание на курсовое проектирование (вычислительную практику)

В качестве задания на курсовое проектирование студенту предлагается создать прикладной программный продукт, решающий некоторую поставленную задачу. Важным моментом в постановке задачи на курсовой проект является то, что задача ставится максимально неформально, формулировка задачи предполагает использование только терминологии предметной области, из которой берется задача. Это соответствует тому, какие задачи возникают при разработке программного обеспечения в реальной ситуации. Уточнение того, как будет выглядеть и как будет работать программный продукт (т.е. постановка формализованной задачи на разработку интерфейсов и функциональностей), происходит в ходе выполнения первого этапа курсового проекта – при разработке технического задания на разработку программного продукта.

Этапы выполнения курсового проекта

Первый этап. На основании задания на курсовой проект студент разрабатывает подробное техническое задание на разработку требуемого программного продукта.

Техническое задание должно содержать подробное описание будущего программного продукта, включая все его функциональные особенности и полный макет интерфейса. В техническом задании должны быть описаны все штатные и нештатные режимы функционирования программного продукта. Для всех нештатных режимов функционирования, должна быть специфицирована методика восстановления после возникновения соответствующих ошибок. Кроме того, в техническом задании должны быть указаны все специфические требования на вычислительную и/или пространственную эффективность программного продукта.

По завершению этого этапа сдается первый отчет по выполнению курсового проекта — техническое задание. Техническое задание должно содержать примерно 2-3 страницы текста (без учета объема графических изображений и схем, описывающих внешний вид интерфейса). После утверждения технического задания преподавателем, оно становится основным документом, определяющим дальнейшую разработку программного продукта в рамках курсового проекта.

Преподавателем могут быть внесены изменения в техническое задание, направленные как на увеличение функциональных возможностей продукта, так и на упрощение этого продукта. В дальнейшем, программный продукт, не соответствующий техническому

заданию, не может быть представлен в качестве результата выполнения курсового проекта.

Второй этап выполнения проекта предполагает разработку (проектирование) необходимых для программирования задачи структур данных и алгоритмов.

На этом этапе, на основании технического задания, формулируются требования к структурам данных и алгоритмам, при необходимости устанавливаются априорные нижние оценки временной и пространственной сложности соответствующих алгоритмов. В соответствии с этими требованиями, студент разрабатывает структуры данных и алгоритмы, анализирует их на оптимальность и, в случае необходимости, перерабатывает.

В конце этапа сдается отчет, в котором описываются соответствующие структуры данных и алгоритмы, и обосновывается оптимальность предложенного решения. В этот же отчет включаются объявления основных классов и функций, которые реализуют соответствующие структуры данных и алгоритмы. Объем отчета составляет примерно 2-3 страницы (без учета программного кода).

Третий этап — это этап собственно программирования (кодирования) продукта. По завершению этого этапа студентом должен быть разработан программный продукт, полностью функционирующий в соответствии с техническим заданием и требованиям к алгоритмам и структурам данных, разработанными в ходе выполнения предыдущего этапа.

Написанный программный код должен быть аккуратно документирован. При этом особое внимание необходимо уделить документированию основных функций и классов.

Сдача работы на этом этапе состоит в демонстрации работоспособности разработанного программного продукта в полной функциональности и предъявлении для проверки (в электронном виде) документированного программного кода.

Разработанный программный продукт и его документированный программный код не распечатывается, сохраняются и сдаются впоследствии в электронном виде.

Четвертый этап. На этом этапе студент разрабатывается инструкция пользователя (объем которой составляет примерно 3-4 страниц, не считая графических изображений и примеров ввода/вывода программного продукта) и готовит отчет, в котором обосновывает соответствие разработанного программного продукта техническому заданию (объем отчета 2-3 страницы без учета графических изображений, фрагментов программного кода и другого сопроводительного материала).

Отчет о выполнении курсового проекта

По завершении выполнения курсового проекта студент предоставляет письменную работу с названием "Курсовой проект", которая состоит из титульного листа, оглавления и следующих содержательных разделов:

1. Описание задания, полученное от преподавателя
2. Техническое задание (отчет по первому этапу выполнения курсового проекта)
3. Описание алгоритмов и структур данных, обоснование оптимальности этих алгоритмов и структур данных (отчет по второму этапу курсового проекта)
4. Примеры работы программного продукта в штатных и нештатных режимах (наборы исходных и обработанных данных, screen shots и т.д.) Здесь необходимо привести примеры работы программы при возникновении нештатных ситуаций и показать, как в этих ситуациях ведет себя разработанный программный продукт.
5. Инструкция пользователя и отчет о соответствии разработанного программного продукта техническому заданию (отчет по четвертому этапу выполнения проекта)
6. Библиографический список использованной литературы.

К работе необходимо приложить собственно программный продукт, созданный в соответствии с заданием (прилагается электронный носитель, как правило, CD-ROM, на котором записаны все необходимые исполняемые файлы и библиотеки). Все тексты (отчеты, инструкция пользователя, текст курсовой работы) и исходные коды программ также представляются на соответствующем носителе в электронном виде.

Основные требования к разработке программного продукта

1. Используемые структуры данных адекватны задаче, их реализация выполнена с учетом требований методологии объектно-ориентированного программирования.
2. Предложенный алгоритм имеет наилучшую оценку временной сложности в среднем (этот факт должен быть формально строго обоснован) и грамотно закодирован.
3. Интерфейс пользователя удовлетворяет стандартным требованиям к интерфейсу Windows-приложений и не содержит неожиданностей (т.е. стандартные пункты меню и т.н. "горячие" клавиши функционируют общепринятым образом).
4. Продукт должен содержать справочную систему и подробное руководство пользователя в текстовом виде.
5. Все этапы выполнения курсового проекта должны быть завершены в отведенное для этого время. Завершением этапа считается **успешная** сдача той части работы, которая оговорена выше.

Сроки выполнения работы. Методика оценки работы.

Срок завершения первого этапа работы: 8 учебная неделя. Срок завершения второго этапа: 10 учебная неделя. Срок представления законченного программного продукта: 15 учебная неделя. Срок окончательной сдачи работы на проверку и допуск к защите: конец семестра, начало зачетной сессии. Защита курсового проекта происходит во время зачетной сессии.

Работа на первом, втором и четвертом этапах оценивается по 5-бальной шкале в случае своевременной сдачи работы и по 3-бальной шкале в противном случае. Работа на третьем этапе оценивается по 10-бальной и 5-бальной шкале соответственно. Сумма баллов, набранных на всех этапах выполнения проекта, приводится к стандартной оценке (неудовлетворительно, удовлетворительно, хорошо, отлично). ***Работа, сданная после окончания семестра (во время зачетной сессии или позже), оценивается целиком по 2-х бальной системе (удовлетворительно или неудовлетворительно).***

При сдаче работы возможно (и приветствуется) использование электронных средств коммуникации студента и преподавателя.

Сдача работы до определенной недели означает, что работы должна быть **получена** преподавателем до завершения соответствующей недели.

В ходе защиты курсового проекта возможна корректировка оценок, полученных на всех этапах, как в сторону их увеличения, так и в сторону уменьшения.

Задания

Часть 1. Разработать структуру данных:

1. линейный однонаправленный список;
2. кольцевой однонаправленный список;
3. линейный двунаправленный список;
4. кольцевой двунаправленный список.

Во всех 4-х вариантах предусмотреть параметризацию создаваемых классов по типу хранимых данных, реализацию требуемой структуры данных в виде общего контейнерного типа (разработать все методы доступа и обработки элементов, присущие заданной структуре данных и методы, присущие общему контейнерному типу, включая итератор).

Разработать методы, позволяющие визуализировать работу созданной структуры данных и необходимый интерфейс пользователя.

Часть 2. Используя структуру данных, созданную при выполнении первой части задания разработать тип данных "Многочлен от одной комплексной переменной", при реализации которой многочлен представляется как список своих мономов (т.е. слагаемых вида $a_n x^n$). Каждый моном храниться как запись, содержащая целое поле, соответствующее показателю степени и поле, содержащее комплексный коэффициент a_n . Реализовать алгоритмы сложения, вычитания, умножения многочленов, деления многочленов с остатком, нахождения значения многочлена в заданной точке. Изучить, реализовать и использовать алгоритм быстрого преобразования Фурье многочленов. Разработать методы, визуализирующие представление многочленов и иллюстрирующие работу с ними.

Предусмотреть ввод многочленов в режиме диалога с пользователем, построение интерполяционных многочленов для функций, принадлежащих к следующим семействам

1. Алгебраические функции и экспоненты,
2. Тригонометрические функции,
3. Гиперболические функции,
4. Обратные тригонометрические функции.

Кроме того, необходимо предусмотреть генерацию показателей степеней и коэффициентов многочленов следующим образом

A. Коэффициенты выбираются случайным образом из последовательности

$(3^{2n} \pmod{100})_{n=1}^{\infty}$, а показатели степеней мономов формируются так: пусть (m_j)

– некоторая случайно выбранная подпоследовательность последовательности

$(37n \pmod{10})_{n=1}^{\infty}$. Тогда степени мономов – это числа $n_k = \sum_{j=1}^k m_j$

B. Коэффициенты выбираются случайным образом из последовательности

$(5^{2n+1} \pmod{100})_{n=1}^{\infty}$, а показатели степеней мономов формируются так: пусть (m_j)

– некоторая случайно выбранная подпоследовательность последовательности

$(19n \pmod{5})_{n=1}^{\infty}$. Тогда степени мономов – это числа $n_k = \sum_{j=1}^k m_j$

C. Коэффициенты выбираются случайным образом из последовательности

$(7^{2n} \pmod{100})_{n=1}^{\infty}$, а показатели степеней мономов формируются так: пусть (m_j)

– некоторая случайно выбранная подпоследовательность последовательности

$(137n \pmod{7})_{n=1}^{\infty}$. Тогда степени мономов – это числа $n_k = \sum_{j=1}^k m_j$

D. Коэффициенты выбираются случайным образом из последовательности

$(6^{2n+1} \pmod{100})_{n=1}^{\infty}$, а показатели степеней мономов формируются так: пусть (m_j)

– некоторая случайно выбранная подпоследовательность последовательности

$(17n \pmod{4})_{n=1}^{\infty}$. Тогда степени мономов – это числа $n_k = \sum_{j=1}^k m_j$

Часть 3. Разработать Windows приложение, демонстрирующее выполнение всех частей задания и иллюстрирующее работу созданных типов и структур данных.

Шаблоны функций и классов, способы визуализации работы типов и структур данных, выбор используемых алгоритмов, интерфейс приложения (из части 3) и другие детали реализации в обязательном порядке согласовываются с преподавателем на первых двух этапах выполнения работы.

Таблица определения вариантов:

Номер по списку группы	Вариант (часть 1)	Вариант (часть 2)
1, 32	1	1,А
2, 31	1	2,А
3, 30	1	3,А
4, 29	1	4,А
5, 28	2	1,В
6, 27	2	2,В
7,26	2	3,В
8, 25	2	4,В
9,24	3	1,С
10,23	3	2,С
11,22	3	3,С
12,21	3	4,С
13,20	4	1,Д
14,19	4	2,Д
15,18	4	3,Д
16,17	4	4,Д