

**Негосударственное аккредитованное некоммерческое
частное образовательное учреждение высшего образования
«АКАДЕМИЯ МАРКЕТИНГА И СОЦИАЛЬНО-ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ – ИМСИТ»
(г. Краснодар)**

Кафедра математики и вычислительной техники

Рассмотрено и одобрено на заседании
кафедры Математики и вычислительной
техники Академии ИМСИТ, протокол
№8 от 19 марта 2018 года,
зав. кафедрой



Н.С. Нестерова



УТВЕРЖДЕНО
Научно-методическим советом академии
протокол №8 от 16 апреля 2018 года
Председатель НМС,
проректор по учебной работе,
профессор

Н.Н. Павелко

**ПРОЕКТИРОВАНИЕ И АРХИТЕКТУРА
ПРОГРАММНЫХ СИСТЕМ**

Методические указания по выполнению курсового проекта
для студентов направления подготовки
09.03.04 Программная инженерия
Направленность (профиль) программы:
«Информационно-вычислительные системы»
Квалификация
Бакалавр

**г. Краснодар
2018**

Составитель: В.В. Бужан



к.ф.-м.н., доцент.

ПРОЕКТИРОВАНИЕ И АРХИТЕКТУРА ПРОГРАММНЫХ СИСТЕМ: Методические указания по выполнению курсового проекта для студентов очной и заочно-ной форм обучения направления подготовки 09.03.04 Программная инженерия/ В.В. Бужан. – Краснодар: ИМСИТ, 2018. – 29 с.

В методических указаниях рассмотрены основные требования к структуре, содержанию и оформлению курсового проекта. Представлены примерные темы работ. Для улучшения восприятия основные разделы сопровождаются примерами. Описан порядок защиты работ и указаны критерии их оценки.

Рассмотрено и одобрено на заседании кафедры математики и вычислительной техники от 19.03.2018 г., протокол №8.

Утверждено научно-методическим советом Академии от 16.04.2018 г., протокол №8.

Рецензенты:

Левченко В.И., к.т.н., доцент, доцент кафедры автоматизации производственных процессов КубГТУ

Суриков А.И., директор ООО «1С-КОНСОЛЬ»

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Содержание курсового проекта	4
2 Методические указания по курсовому проекту	6
2.1 Введение и постановка задачи	6
2.2 Используемый математический аппарат.....	6
2.3 Программная реализация решения задачи.....	8
2.4 Руководство пользователя	9
2.5 Руководство программиста.....	12
3 Правила оформления курсового проекта.....	14
4 Порядок защиты курсового проекта	15
5 Критерии оценки курсового проекта	16
6 Тематика курсовых проектов.....	17
7 Оценочные средства для проведения аттестации уровня сформированности компетенций обучающихся при выполнении курсового проекта	18
8 Тезаурус.....	18
9 Выполнение курсового проекта лицами с ограниченными возможностями здоровья.....	20
10 Рекомендуемая литература	21
10.1 Основная литература	21
10.2 Дополнительная литература	22

ВВЕДЕНИЕ

Выполнение курсового проекта – один из важнейших этапов применения теоретических знаний для разработки программного обеспечения.

Студент обязан овладеть методологией программирования на языке высокого уровня, а также научиться грамотно оформлять текстовую и программную документацию.

В процессе выполнения курсового проекта студент повышает качество своих знаний, учится работать с научно-методической литературой, знакомится с новыми для него предметными областями.

Курсовой проект, содержащие элементы исследований, могут быть отнесены к студенческим научно-исследовательским работам и представлены на студенческие конкурсы. Варианты заданий на курсовой проект могут быть предложены самими студентами по согласованию с преподавателем.

Методические указания предназначены для студентов направления подготовки 09.03.04 Программная инженерия и используют полученные ими к моменту написания курсового проекта знания по дисциплине «Проектирование и архитектура программных систем» и другим смежным дисциплинам.

Методические указания направлены на формирование следующих компетенций:

1. способность к самоорганизации и самообразованию (ОК-7);
2. способность осваивать методики использования программных средств для решения практических задач (ОПК-2);
3. способность разрабатывать бизнес-планы и технические задания на оснащение отделов, лабораторий, офисов компьютерным и сетевым оборудованием (ОПК-3);
4. способность участвовать в настройке и наладке программно-аппаратных комплексов (ОПК-4);
5. способность разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов «человек - электронно-вычислительная машина» (ПК-1);

способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования (ПК-2).

1 Содержание курсового проекта

В процессе написания курсового проекта студент обязан:

- грамотно сформулировать постановку задачи;
- разработать алгоритм решения задачи;
- воспроизвести алгоритм в одной из принятых форм записи алгоритмов;
- использовать один из языков программирования высокого уровня;

- освоить структурный и объектно-ориентированный подходы к программированию;
- применить один из пакетов профессионального программирования (Borland C++ Builder, Turbo C++ Explorer, Microsoft Visual Studio и т.п.);
- закодировать алгоритм на языке программирования высокого уровня;
- воспользоваться стандартными библиотеками пакета профессионального программирования, выбранного в качестве инструмента разработки программы;
- создать программу, в соответствии с заданием на курсовой проект.

Работу необходимо выполнить в сроки, отведенные кафедрой, и представить в виде пояснительной записки.

Структура пояснительной записки следующая:

1. Титульный лист.
2. Задание на курсовой проект.
3. Реферат.
4. Содержание.
5. Введение.
6. Основная часть.
 - 6.1. Постановка задачи.
 - 6.2. Описание используемого математического аппарата (если необходимо).
 - 6.3. Программная реализация решения задачи (созданные студентом программные блоки и их описание).
 - 6.4. Руководство пользователя.
 - 6.5. Руководство программиста
7. Заключение (краткие выводы).
8. Список использованных источников и литературы.
9. Приложения.

При необходимости по согласованию с преподавателем приведенный перечень может быть расширен за счет включения дополнительных разделов.

Пояснительная записка должна быть сохранена:

- на твердом носителе в виде страниц текста, напечатанного с одной стороны на отдельных пронумерованных листах бумаги;
- на электронном носителе в любом из распространенных форматов текстовых документов (DOC, RTF, PDF и т. д.).

Загрузочный и исходные модули разработанного программного проекта (вместе со всеми вспомогательными данными), а также файл с пояснительной запиской сохраняют на одном физическом носителе данных (гибкий магнитный диск, CD или DVD-диск, флэш-карта памяти и т. п.). Сам физический носитель вместе с твердой копией пояснительной записки представляют на кафедру в заранее оговоренные сроки.

Материал курсового проекта должен быть изложен в строгой научной форме, а сведения, заимствованные из других источников следует осмыс-

лить и переработать в соответствии со стилем пояснительной записки с обязательным указанием источника. Не допускаются вставки электронных копий фрагментов используемых источников. Исключение составляют графические изображения в качестве иллюстраций.

2 Методические указания по курсовому проекту

2.1 Введение и постановка задачи

В разделе «Введение» необходимо отразить в общих чертах проблему, которой посвящена работа, обосновать её актуальность (важность), кратко перечислить цели курсового проекта.

В работе должна быть чётко сформулирована постановка задачи. Этому посвящён соответствующий раздел, в котором требуется точно сформулировать задачу с описанием входной и выходной информации.

Приведём пример постановки задачи по теме курсового проекта «Разработка математического пакета векторной алгебры».

Реализовать программно основные операции над произвольными векторами: сложение, умножение на число, скалярное и векторное произведение. Для векторов определить новый абстрактный тип данных – класс. Данные-члены (в терминологии C#) – три переменные вещественного типа для хранения координат вектора. Функции-члены – два конструктора: без инициализации данных и с инициализацией данных. Для операций скалярного умножения векторов и умножения вектора на число осуществить перегрузку операции $*$, а для их сложения – операции $+$.

Входной информацией могут быть два вектора или вектор и действительное число в зависимости от типа операции.

2.2 Используемый математический аппарат

Если в программе применяются расчётные формулы, их необходимо представить в названном разделе.

В качестве примера опишем расчётные формулы в задаче, сформулированной ранее.

Операции над двумя произвольными векторами из векторного пространства R_3 : $\vec{a} = (a_1, a_2, a_3)$ и $\vec{b} = (b_1, b_2, b_3)$ реализованы в виде следующих математических зависимостей

а) сложение:

$$\vec{a} + \vec{b} = (a_1 + b_1, a_2 + b_2, a_3 + b_3) \quad (2.1)$$

б) скалярное произведение:

$$(\vec{a}, \vec{b}) = a_1 b_1 + a_2 b_2 + a_3 b_3 \quad (2.2)$$

с) умножение на число:

$$c\vec{a} = (ca_1, ca_2, ca_3). \quad (2.3)$$

Другой пример иллюстрирует использование метода решения системы линейных алгебраических уравнений:

«Метод Гаусса или метод исключения неизвестного используется для решения линейных систем уравнений

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2; \\ &\dots\dots\dots \\ a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1. \end{aligned} \quad (2.4)$$

с невырожденной матрицей коэффициентов

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, |A| \neq 0 \quad (2.5)$$

Метод основан на использовании элементарных преобразований над уравнениями системы. К элементарным преобразованиям относятся три действия:

1. Если поменять местами любые два уравнения системы, то получим систему уравнений, эквивалентную первой (имеющую то же множество решений).

2. Если любое уравнение системы умножить на не равное нулю число, то получим систему уравнений, эквивалентную первой.

3. Если к любому уравнению системы прибавить другое уравнение, умноженное на число, то получим эквивалентную систему уравнений.

Метод Гаусса решения системы линейных уравнений состоит из двух этапов – *прямого хода метода Гаусса* и *обратного хода метода Гаусса* [7].

Прямой ход метода Гаусса. Предположим, что $a_{11} \neq 0$. В противном случае поменяем местами первое уравнение и уравнение с ненулевым коэффициентом a_{i1} (элементарное преобразование 1). Первое уравнение системы

$$\sum_{j=1}^n a_{1j}x_j = b_1 \quad (2.6)$$

делим на a_{11} . В результате получим уравнение

$$x_1 + \sum_{j=2}^n a_{1j}^1x_j = b_1^1 \quad (2.7)$$

(верхним индексом обозначен номер шага преобразования). Затем из каждого i -го уравнения системы ($i = 2, \dots, n$) вычитаем первое уравнение, умноженное на соответствующий коэффициент a_{i1} . В результате эти уравнения преобразуются к виду

$$\sum_{j=2}^n a_{ij}^1 x_j = b_i^1, \quad i = 2, \dots, n. \quad (2.8)$$

Таким образом, первое неизвестное исключено из всех уравнений, кроме первого. Далее, в предположении, что $a_{22}^1 \neq 0$, делим второе уравнение на a_{22}^1 и исключаем неизвестное x_2 из всех уравнений, начиная с третьего. В результате последовательного исключения неизвестных система уравнений преобразуется в систему уравнений с верхней треугольной матрицей коэффициентов

$$x_i + \sum_{j=i+1}^n a_{ij}^i x_j = b_i^i, \quad i = 1, \dots, n, \quad (2.10)$$

которая эквивалентна исходной системе, так как получена с помощью цепочки элементарных преобразований.

Обратный ход метода Гаусса. Из n -го уравнения полученной системы определяем x_n , из $(n-1)$ -го уравнения определяем x_{n-1} и т. д. до x_1 ».

2.3 Программная реализация решения задачи

В данной части курсового проекта необходимо выполнить следующее.

1. Указать используемый язык программирования и среду разработки программного продукта;

2. Обосновать выбор языка программирования и среды разработки. Например, если использован язык программирования C# и среда разработки Visual Studio, этот выбор можно обосновать следующим образом. Язык программирования C# является современным объектно-ориентированным языком программирования, сравнительно простым в использовании, который в то же время позволяет разрабатывать приложения любой сложности. C# относится к мультиплатформенным языкам программирования. Созданные на нём программы выполняются не самой операционной системой (ОС), а виртуальной машиной .NET. Поэтому программы C# могут работать на любом компьютере с любой ОС, на котором установлена .NET. Данное обстоятельство снижает стоимость программного продукта, исключая его адаптацию к другим платформам и, экономя время на сопровождении, решает проблему совместимости со средой выполнения заказчика. Профессиональная среда разработки программного обеспечения Microsoft Visual Studio является в настоящее время одной из самых распространенных среди аналогов других фирм – производителей программного обеспечения. Она существенно ускоряет процесс создания приложений самого разного предназначения за счёт использования обширной библиотеки визуальных компонентов и развитой интегрированной среды разработки.

3. Указать подход к программированию (структурный, или объектно-ориентированный), использованный в работе.

В основе *структурного подхода* лежит *декомпозиция* (разбиение на части) сложных систем для последующей их реализации в виде отдельных небольших блоков (40-50 операторов). Вначале разрабатывают программу, которой в работе отводится роль головной. Из неё должен быть организован вызов всех остальных подпрограмм. Их роль на начальной стадии проектирования выполняют так называемые «заглушки» – подпрограммы, которые имитируют работу будущих подпрограмм. В процессе работы, «заглушки» замещают готовыми подпрограммами. Такой способ проектирования называется «сверху-вниз». Из современных наиболее распространённых языков программирования структурный подход поддерживают C/C++ и Pascal.

В основе *объектно-ориентированного подхода* лежат три принципа, призванные отождествить объекты программирования с объектами реального мира: *инкапсуляция*, *наследование* и *полиморфизм*. *Инкапсуляция* означает объединение данных (переменных и констант) с функциями, которые могут работать только с этими данными, что повышает их защищённость от случайного изменения другими подпрограммами. Типы данных, инкапсулирующие данные и функции получили название *классы*. *Наследование* означает приобретение одними классами некоторых признаков другого класса. Такой подход имеет аналог в классификации видов живых организмов. Он позволяет ускорить процесс программирования, избавляя от необходимости в новых классах дублировать элементы из других классов. И, наконец, *полиморфизм* предусматривает возможность одинаково идентифицировать различные объекты, предоставляя компилятору право самому выбирать тот из них, параметры которого соответствуют требованиям, предъявляемым при использовании объекта. Примером может служить перегрузка функций.

4. Представить описание алгоритма, который реализует программа. При выборе способа описания следует исходить из выбранного подхода к программированию. При структурном подходе для записи алгоритма можно использовать схемы алгоритмов в соответствии с ГОСТ 19.701-90. Пример записи алгоритма решения системы линейных алгебраических уравнений методом Гаусса представлен на рисунке 2.1.

2.4 Руководство пользователя

В разделе должно быть освещено следующее.

1. Системные требования к среде выполнения программы. Например: Для нормальной работы с данным программным продуктом требуется IBM-совместимый компьютер следующей конфигурации: тактовая частота процессора не менее 500 МГц, оперативная память не менее 32 МВ, видеоадаптер SVGA, объём свободного места на жёстком диске не менее 5 МВ, операционная система Microsoft Windows 95/98/NT/2000/XP, устройство для чтения компакт-дисков.

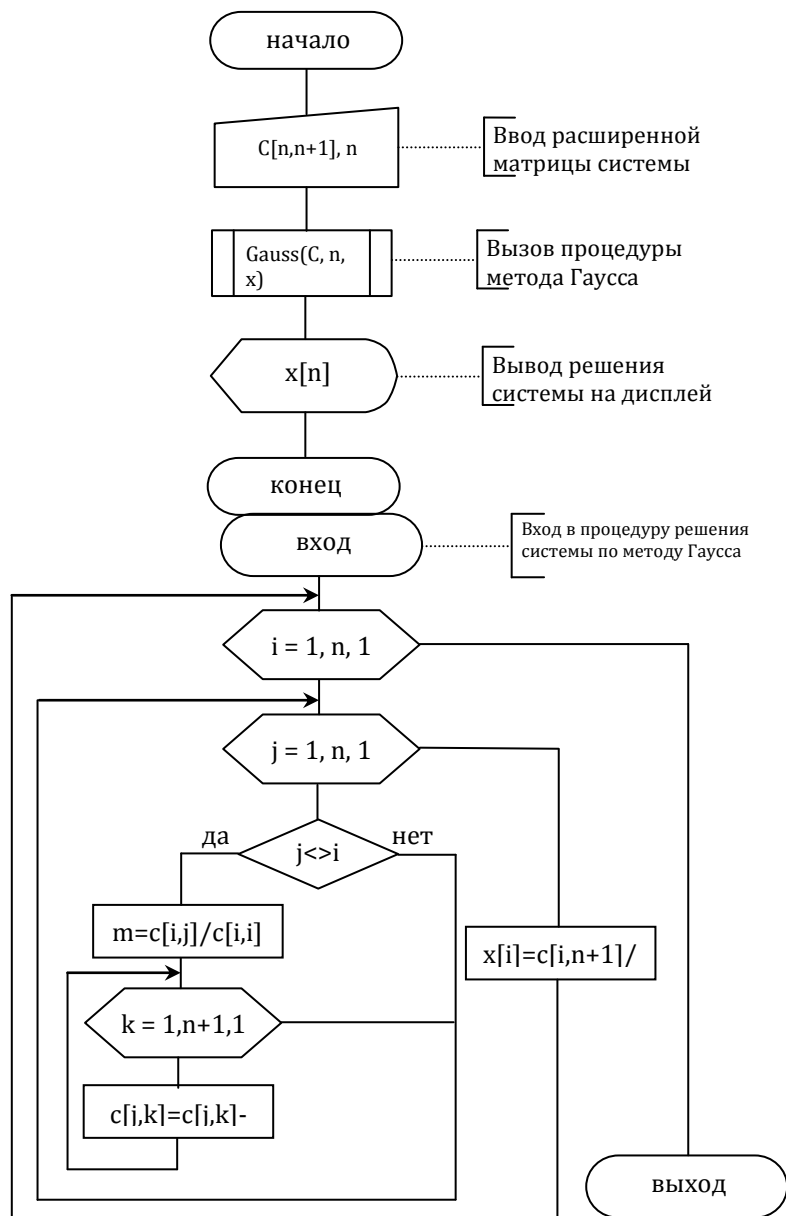


Рисунок 2.1 – Схема алгоритма решения системы уравнений методом Гаусса

2. Установка программы на компьютер пользователя. Здесь следует описать этапы инсталляции программы: с помощью программы-инсталлятора или простым копированием. В любом случае должен быть указан источник с программным продуктом (компакт-диск, дискета и т.п.), способ хранения программного продукта (в форме установочного пакета (дистрибутива) или программного файла, готового к использованию), способ установки (запуск программы-инсталлятора в случае дистрибутива или простым копированием). Так, для установки приложения из рассмотренного ранее примера решения систем уравнений можно указать следующее.

Установка программного продукта производится с компакт-диска на компьютер. Для этого требуется:

- а) создать папку с именем «GAUSS» в директории C:\Program Files\;

- b) вставить CD-ROM с программным продуктом в устройство чтения компакт-дисков;
- c) произвести копирование программного продукта в папку C:\Program Files\GAUSS (программный продукт представлен в виде единственного exe-файла)».

3. Работа с программой. В данном пункте нужно описать использование программы на каком-либо примере. Для ясности рекомендуется использовать скриншоты (снимки экрана) окон программы, отражающих состояние вводимых и выводимых данных на каждом шаге демонстрации. Рассмотрим в качестве примера демонстрацию программы GAUSS.

Работа пользователя с программой начинается с открытия файла gauss.exe в окне проводника (либо иного файлового менеджера) из папки с установленной программой (по умолчанию – это C:\Program Files\GAUSS). В результате откроется окно приложения рисунок 2.2.

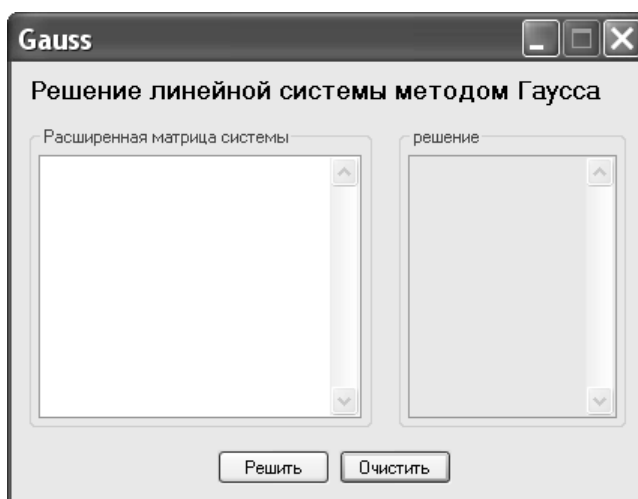


Рисунок 2.2 – Окно приложения после загрузки

В левую панель следует вводить коэффициенты расширенной матрицы системы по строкам, разделяя числа пробелами и соблюдая принятые в региональных настройках требования к вводу чисел (рисунок 2.3).

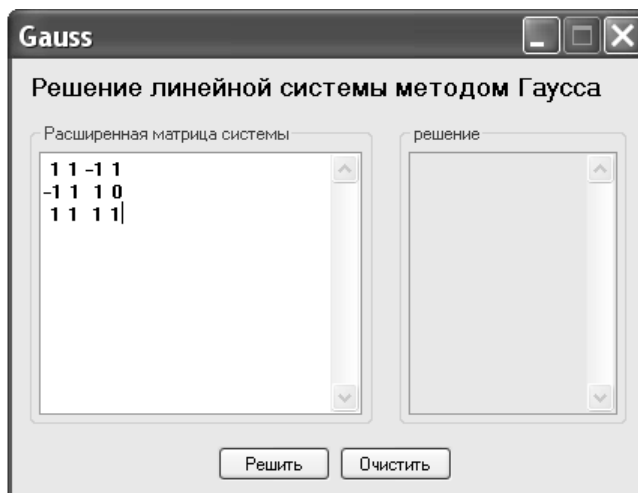


Рисунок 2.3 – Ввод данных

Во избежание некорректной работы программы не допускается ввод пустых строк.

Нажатие на кнопку «Решить» инициирует процесс вычислений. По прошествии некоторого времени, результат отобразится на правой панели (рисунок 2.4).

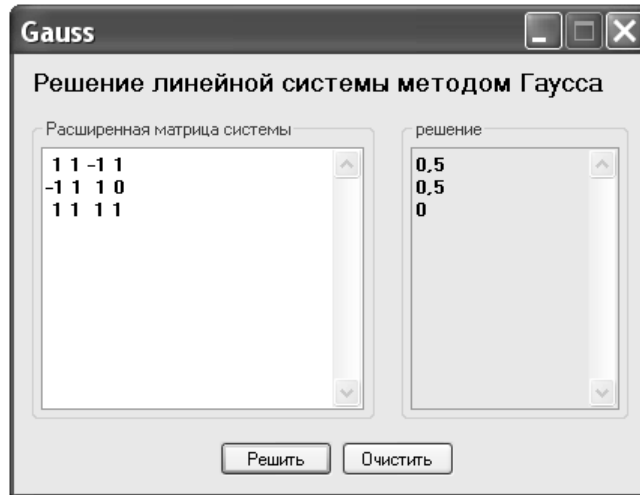


Рисунок 2.4 – Отображение результата

4. Удаление программы. Здесь необходимо представить инструкцию по безопасному удалению программы с компьютера пользователя.

Замечание: Если программа установлена простым копированием, то удаление (деинсталляция) заключается в прекращении работы программы с последующим перемещением её значка из окна проводника в корзину. Если для установки был использован инсталлятор, то для удаления, возможно, понадобится представить подробную инструкцию либо предоставить и описать программу деинсталляции.

2.5 Руководство программиста

Данный раздел должен содержать краткие сведения о программном продукте и описания его основных объектов с точки зрения разработчика.

В *сведениях* обычно указывают компилятор и/или среду разработки программного обеспечения, в которой выполнено задание, перечисляют имена загрузочного модуля (программного файла) или сборки, статических или динамических библиотек (если используется), файлов с исходными модулями (по желанию разработчика). Все перечисленные модули должны также присутствовать и на электронном носителе с курсового проекта. При этом, если использованы библиотеки сторонних производителей и их присутствие необходимо для функционирования программы, они могут быть включены при условии обладания открытой лицензией. В данном случае

необходимо указать производителя и включить текстовый файл с лицензией в состав электронного носителя.

В описания объектов программного продукта входят объявления основных переменных и констант, использованных в программе, и описание пользовательских типов данных (массивов, структур, классов, перечислений и т.д.).

Например, руководство к программному продукту для решения систем линейных уравнений может содержать следующие сведения (в терминах C#):

Программный продукт Gauss разработан на языке программирования высокого уровня C# в среде Visual Studio 2008 и предназначен к использованию на любой платформе, где установлена среда .NET Framework версии не ниже 3.5 или Mono Framework версии не ниже 2.0. Программный продукт состоит из файлов сборок gauss.exe и lineq.dll.

В программе используются переменные:

const int n=10; – целочисленная константа, равная максимально допустимому числу уравнений системы, которую может решить подпрограмма;

double[] C = new double[n,n+1]; – двумерный массив коэффициентов расширенной матрицы системы;

int i, j, k; – целочисленные переменные, содержащие значения индексов двумерного массива.

Другой пример иллюстрирует описания объектов из задания по разработке программного пакета векторной алгебры:

В программе описан абстрактный тип данных – класс Vector, в котором инкапсулированы данные-члены – вещественные переменные x, y и z для хранения координат вектора и функции-члены – свойства для доступа к данным, конструктор класса и две перегруженные операции скалярного произведения векторов и умножение вектора на число:

```
class Vector
{
    double X, Y, Z; // координаты вектора
    public double x //свойство
    {
        get
        {
            return X;
        }
        set
        {
            X=value;
        }
    }
    public double y //свойство
    {
```

```

    get
    {
        return Y;
    }
    set
    {
        Y = value;
    }
}

public double z //свойство
{
    get
    {
        return Z;
    }
    set
    {
        Z = value;
    }
}

public Vector() { } //конструктор

public static double operator *(Vector vl, Vector vr)
{// скалярное произведение векторов
    return vl.x * vr.x + vl.y * vr.y + vl.z * vr.z;
}

public static Vector operator *(double a, Vector v)
{// умножение вектора на число
    Vector r = new Vector();
    r.x = a * v.x;
    r.y = a * v.y;
    r.z = a * v.z;

    return r;
}
}»

```

79. Руководство программиста оформляют в соответствии с ГОСТ 19.504-

3 Правила оформления курсового проекта

Работу выполняют на белой нелинованной бумаге формата А4 (210 x 297 мм) на компьютере через 1,5 интервала (количество знаков в строке - 64, включая пробелы; высота букв - не менее 1,8 мм (кегель не менее 12)). Полу-жирный шрифт не применяется.

Текст работы должен быть кратким, четким и не допускать различных толкований. В нем должны применяться научно-технические термины, обозначения и определения, установленные соответствующими стандартами, а при их отсутствии - общепринятые в научно-технической литературе.

4 Порядок защиты курсового проекта

Защита курсового проекта заключается в изложении постановки задачи, актуальности темы, теоретического обоснования построенной модели базы данных, перечислении основных результатов работы и в демонстрации работоспособности приложения базы данных.

Работу выносят на защиту в электронной (презентация) и печатной (рукописной) форме.

Печатная форма – распечатанный и переплетённый текст работы (рукопись), оформленный в соответствии с требованиями к курсовому проекту.

Электронная форма представляет собой презентацию, используемую в докладе в качестве иллюстрации функциональности созданного программного обеспечения.

Для создания презентации можно использовать специальные программные средства, например, Microsoft Power Point.

Презентация может содержать графические изображения (схемы, графики, диаграммы, таблицы, скриншоты и т.д.), облегчающие верное восприятие комиссией сути доклада. Желательно, чтобы презентация состояла из нескольких страниц, каждая из которых содержит отдельный пункт доклада. Каждый пункт должен сопровождаться кратким текстовым комментарием. Например, наименованием этапа, перечнем основных результатов, подписью к рисунку и т.д.

Защиту начинают с доклада студента по теме работы. Доклад сопровождается презентацией. Члены комиссии задают дополнительные вопросы по теме курсового проекта.

В докладе необходимо отразить следующее. Если приложение разработано в какой-либо среде программирования (Delphi, C++ Builder, JBuilder, Visual C++ и т.п.), обосновать выбор именно этого языка и среды программирования. Например, указать на использование специальных библиотек (VCL, MFC и т.д.) для разработки интерфейса приложения баз данных, возможность создавать с применением программирования гибкой программы для манипулирования данными, отметить преимущества выбранного языка по сравнению с другими.

Если приложение создано с помощью каких-либо средств визуального проектирования (например, в среде Access), то обосновать выбор именно этого подхода. Например, указать на то, что среда разработки приложений Access имеет следующие достоинства:

- обладает удобной в использовании панелью инструментов (инструментарием);
- поддерживает встроенный механизм транзакций в отличие от сред программирования, где может потребоваться проектирование транзакций вручную;
- имеет удобные для работы программы-мастера форм и запросов, которые позволяют реализовывать модель базы данных, не отвлекаясь на программирование методов управления данными;
- широкую распространённость приложений Microsoft Office, что обуславливает хорошую совместимость реализованной в работе базы данных со всеми системами с установленным программным пакетом Microsoft Office.

5 Критерии оценки курсового проекта

Работу оценивают отметкой «5» (отлично), если студент:

- выполнил все пункты задания по теме работы;
- полностью раскрыл содержание темы работы;
- изложил материал грамотно в логической последовательности, точно использовал терминологию;
- показал умения иллюстрировать теоретические положения конкретными практическими примерами;
- продемонстрировал усвоение изученных сопутствующих вопросов, сформированность и устойчивость знаний;
- отвечал самостоятельно на дополнительные вопросы комиссии;
- проявил достаточно высокую активность на стадии выполнения курсового проекта (регулярно посещал консультации руководителя работы, самостоятельно изучил учебную литературу и освоил техническую документацию по теме работы, спроектировал модель данных и разработал приложение базы данных). Возможны одна – две неточности при освещении второстепенных вопросов.

Работу оценивают отметкой «4» (хорошо), если она, в основном, удовлетворяет требованиям отметки «5», но при этом имеет один из недостатков:

- допущены один – два недочета в докладе по курсовому проекту, исправленные по замечанию членов комиссии;
- допущены ошибки или недочеты в работе, легко исправленные по замечанию членов комиссии.

Отметку «3» (удовлетворительно) ставят в следующих случаях:

- неполно или непоследовательно раскрыта тема работы, но показано общее понимание целей работы и получены результаты, предусмотренные заданием;
- допущены ошибки в определении понятий, выкладках, рассуждениях, исправленные после нескольких наводящих вопросов членов комиссии;
- при знании теоретического материала по теме работы, выявлен недостаток основных умений и навыков.

Отметку «2» (неудовлетворительно) ставят, если:

- не раскрыто основное содержание темы работы;
- обнаружено незнание и непонимание студентом целей работы;
- допущены ошибки в определении понятий, при использовании терминологии, которые не исправлены после нескольких наводящих вопросов членов комиссии. Допущены ошибки в освещении темы работы во время доклада.

6 Тематика курсовых проектов

В данном разделе представлены примерные темы курсовых проектов по дисциплине «Проектирование и архитектура программных систем».

1. Проектирование обучающих систем для образовательной организации.
2. Проектирование тестирующих систем для образовательной организации.
3. Проектирование автоматизированных информационных систем и автоматизированных систем управления предприятием.
4. Проектирование и разработка хранилищ и баз данных предприятия.
5. Проектирование учебно-методических комплексов для изучения дисциплин математического цикла.
6. Проектирование подсистем автоматизированной системы управления вузом.
7. Проектирование мультимедийных учебных курсов для образовательной организации.
8. Проектирование программного обеспечения для ЭВМ и компьютерных сетей.
9. Проектирование прикладных программных систем с использованием баз данных.
10. Проектирование программного обеспечения для автоматизации документооборота предприятия.
11. Проектирование корпоративного хранилища данных предприятия.
12. Проектирование программного обеспечения, использующего параллельные вычисления.
13. Проектирование программного обеспечения, использующего технологии СОМ.
14. Проектирование программного обеспечения с возможностью автоматического обновления.
15. Проектирование программного обеспечения, реализующего распределённые вычисления.
16. Проектирование распределённых баз данных.
17. Проектирование транслятора языка программирования.
18. Проектирование программ-генераторов электронных задачник по учебным дисциплинам.
19. Проектирование автоматизированной системы учёта перемещения материальных ценностей.
20. Проектирование информационной системы типа складов данных.

21. Проектирование информационных систем аналитического характера.
22. Проектирование информационных OLAP-систем.
23. Проектирование программного обеспечения для систем мобильной связи.
24. Проектирование программного обеспечения для вычислений с повышенной степенью точности.
25. Проектирование информационной системы для метеостанций.

7 Оценочные средства для проведения аттестации уровня сформированности компетенций обучающихся при выполнении курсового проекта

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, осваивающих дисциплину «Проектирование и архитектура программных систем».

Оценочные средства включают контрольные материалы для проведения текущего контроля по проблемным вопросам и промежуточной аттестации в форме экзамена и зачёта.

Оценочные средства прилагаются к методическим указаниям.

8 Тезаурус

Алгоритм – последовательность действий, которую необходимо выполнить для получения нужного результата.

Данные-члены – члены класса, которые содержат данные класса – поля, константы, события.

Динамическая библиотека – совокупность подпрограмм, обычно редко используемых, которые хранятся в одном DLL-файле на внешнем носителе и загружаемые в память компьютера во время выполнения программы по мере надобности и выгружаемые из памяти по завершении своей работы.

Виртуальная машина – интерпретатор специального промежуточного кода (байт-кода), реализующий завершающую ступень в платформо-независимом программировании (предшествующая ступень – компиляция байт-кода).

Задача – выполняемая процессором программа.

Загрузочный модуль – готовая к выполнению, прошедшая компиляцию и компоновку, программа.

Класс – абстрактный тип данных, в котором инкапсулируются (объединяются) данные и функции для их обработки.

Компилятор – транслятор с языка программирования высокого уровня. Обработывая исходную программу, компилятор создаёт эквивалентную программу на машинном языке, которая называется также объектной программой или объектным кодом.

Компоновщик – специальная программа, объединяющая в исполняемый модуль объектные коды.

Константа – неизменная величина.

Конструкторы (constructor) – специальные функции, вызываемые автоматически при инициализации объекта.

Идентификатор – имя программного объекта.

Исполняемый модуль – см. *загрузочный модуль*.

Исходный модуль – текст программы на языке программирования.

Интерпретатор – транслятор, который, распознавая исходную программу, не формирует машинный код в явном виде, а для каждой операции исполняет заранее заготовленную машинную команду или команды.

Методы (method) – функции, ассоциированные с некоторым классом.

Модуль – динамическая библиотека без атрибутов сборки.

Модульное программирование – технология, предполагающая выделение групп подпрограмм, использующих одни и те же глобальные данные в отдельно компилируемые модули.

Объект – тип (экземпляр) класса.

Объектная программа – программа на машинном языке, созданная компилятором из исходной программы на языке программирования высокого уровня.

Объектный код – см. *объектная программа*.

Объектный модуль – прошедшая компиляцию программа.

Объектно-ориентированное программирование – технология создания сложного программного обеспечения, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого типа (класса), а классы образуют иерархию с наследованием свойств.

Переменная – именованная область памяти, хранящая данные определённого типа.

Перегрузка функций – использование нескольких функций с одним и тем же именем, но с различными типами параметров.

Перегрузка операций – переопределение действий операций так, чтобы при использовании с объектами конкретного класса они выполняли заданные функции.

Программа – совокупность описаний (объявлений) и операторов, объединённая единым алгоритмом.

Простой тип – тип данных, не инкапсулированный в какой-либо класс.

Процесс – см. *Задача*.

Рекурсия – процесс вычислений, инициирующий сам себя (прямая рекурсия) или группа процессов, вызывающих друг друга (косвенная рекурсия). На практике рекурсия реализуется в виде функций.

Сборка (assembly) – логическая группировка одного или нескольких управляемых модулей и/или файлов ресурсов.

Свойства (property) – наборы функций, которые могут быть доступны таким же способом, как общедоступные поля класса.

Структурное программирование – совокупность рекомендуемых технологических приёмов, охватывающих выполнение всех этапов разработки программного обеспечения. В его основе лежит декомпозиция сложных систем с целью последующей реализации в виде отдельных небольших (до 40-50 операторов) подпрограмм.

Ссылочный тип (reference type) – класс, все экземпляры которого хранят адреса ячеек в специально отведённой для этих целей области памяти, называемой «кучей».

Скриншот – копия фрагмента экранного изображения, сохранённая в одном из графических форматов файлов.

Транслятор – программа, получающая на входе исходный модуль программы и получающая на выходе функционально-эквивалентный исходному модулю объектный модуль.

Тип данных – концепция языков программирования высокого уровня, которая определяет внутреннее представление данных в памяти компьютера, множество значений, которые могут принимать величины этого типа и применимые к ним операции и функции.

Тип значений (value type) – класс, экземпляры которого хранят данные непосредственно в памяти.

Указатель – переменная, предназначенная для хранения адреса ячейки памяти.

Файл – логически связанная совокупность данных определённой длины, имеющая имя.

Форма – объект программы, в котором разработчик размещает элементы управления, служащие для ввода, отображения и изменения данных в полях.

Функции-члены – члены класса, которые обеспечивают некоторую функциональность для манипулирования данными класса.

Член класса – переменная, константа или функция, инкапсулируемая в классе.

Экземпляр класса – см. объект.

9 Выполнение курсового проекта лицами с ограниченными возможностями здоровья

При выполнении работы в аудитории обеспечивается соблюдение следующих общих требований:

- проведение обучения для лиц с ограниченными возможностями здоровья в одной аудитории совместно со студентами, не имеющими ограниченных возможностей здоровья, если это не создает трудностей для них в процессе обучения;
- присутствие в аудитории ассистента, оказывающего обучающимся необходимую техническую помощь с учётом их индивидуальных особенностей (занять рабочее место, передвигаться, прочесть и оформить

задание, общаться с преподавателем);

- пользование необходимыми обучающимся техническими средствами при выполнении практических и других работ в соответствии с учебным планом с учётом их индивидуальных особенностей;

В зависимости от индивидуальных особенностей обучающихся с ограниченными возможностями здоровья образовательная среда Академии обеспечивает выполнение следующих требований при обучении и проведении промежуточной и итоговой аттестации:

а) для слепых:

- задания и иные материалы для аттестации зачитываются ассистентом;
- письменные задания надиктовываются обучающимся ассистенту;

б) для слабовидящих:

- задания и иные учебно-методические материалы оформляются увеличенным шрифтом; обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
- при необходимости обучающимся предоставляется увеличивающее устройство, допускается использование увеличивающих устройств, имеющихся у обучающихся;

в) для глухих и слабослышащих, с тяжёлыми нарушениями речи:

- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающимся предоставляется звукоусиливающая аппаратура индивидуального пользования;
- по их желанию аттестационные испытания проводятся в письменной форме;

г) для лиц с нарушениями опорно-двигательного аппарата (тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания надиктовываются ассистенту;
- по их желанию все аттестационные испытания проводятся в устной форме.

10 Рекомендуемая литература

10.1 Основная литература

1. Бужан В.В. Программирование: учебно-методическое пособие для студентов очной и заочной форм обучения направления подготовки 230100.62 «Программная инженерия». – Краснодар: ИМСИТ, 2013. – 84 с.
2. Бужан В.В. Объектно-ориентированное программирование: учебно-методическое пособие для студентов очной и заочной форм обучения направления подготовки 230100.62 «Программная инженерия». – Краснодар: ИМСИТ, 2013. – 52 с

10.2 Дополнительная литература

1. Давыдова, Н. А. Программирование [Электронный ресурс] : учебное пособие / Н. А. Давыдова, Е. В. Боровская. - 2 изд. (эл.). - М. : БИНОМ. Лаборатория знаний, 2012. - 238 с. Режим доступа: <http://www.znanium.com>.
2. Голицына, О.Л., Попов, И.И. Программирование на языках высокого уровня: учеб. пособие для СПО. – М.: Форум, 2011. – 496 с.
3. Костюк, Ю. Л. Основы разработки алгоритмов [Электронный ресурс] : учебное пособие / Ю. Л. Костюк, И. Л. Фукс. - М. : БИНОМ. Лаборатория знаний, 2010. - 286 с. Режим доступа: <http://www.znanium.com>.
4. Информатика и программирование. Компьютерный практикум: учеб. пособие для вузов / Гуда, А.Н., Бутаков, М.А., Нечитайло, Н.М., Чернов, А.В. – М.: Дашков и К, 2010. – 240 с.
5. Окулов, С. М. Основы программирования [Электронный ресурс] / С. М. Окулов. - 5-е изд., испр. - М. : БИНОМ. Лаборатория знаний, 2010. - 440 с. : Режим доступа: <http://www.znanium.com>.
6. Корнелл Г., Моррисон Дж. Программирование на VISUAL STUDIO: учебный курс.- СПб.: Питер, 2008.-205с.
7. Алексеев Е.Р. Программирование на Visual C++ и Turbo C++ Explorer.- М.:ИТ Пресс, 2007. – 352 с.
8. Пахомов Б.Н. C/C++ и MS Visual C++ 2005 для начинающих. – СПб.: БХВ-Петербург, 2007. – 464 с.
9. Сергеев А.П., Терен А.Н. Программирование в Microsoft Visual C++ 2008. – М.:Издательский дом «Вильямс», 2008. – 352 с.
- 10.Хортон А. Visual C++ 2005: Базовый курс – М.:Издательский дом «Вильямс», 2007. – 1152 с.
- 11.ГОСТ 19.103 – 77. ЕСПД. Обозначение программ и программных документов.
- 12.ГОСТ 19.105 – 78. ЕСПД. Общие требования к программным документам.
- 13.ГОСТ 19.401 – 78. ЕСПД. Текст программы. Требования к содержанию и оформлению.
- 14.ГОСТ 19.402 – 78. ЕСПД. Описание программы.

- 15.ГОСТ 19.404 – 78. ЕСПД. Пояснительная записка. Требования к содержанию и оформлению.
- 16.ГОСТ 19.502 – 78. ЕСПД. Описание применения. Требования к содержанию и оформлению.
- 17.ГОСТ 19.504 – 79. ЕСПД. Руководство программиста.
- 18.ГОСТ 19.701 – 90. ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

ПРИЛОЖЕНИЕ А

(обязательное)

Форма титульного листа пояснительной записки

АКАДЕМИЯ МАРКЕТИНГА И СОЦИАЛЬНО-ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ - ИМСИТ (г. Краснодар)

Факультет информатики и вычислительной техники

Кафедра математики и вычислительных систем

Курсовой проект по дисциплине «Проектиро-
вание и архитектура программных систем»

на тему: «Проектирование базы данных для подведения итогов спор-
тивных соревнований»

направление 09.03.04 Программная инженерия

Работу выполнил
студент 3 курса очной
формы обучения
группы 10-ИВТ-02
Ермоленко Александр
Владимирович

Научный руководитель:

канд. физ.-мат. наук, доцент _____ В.В. Бужан

Работа защищена с оценкой «_____»

Краснодар
2018

ПРИЛОЖЕНИЕ Б
(обязательное)

Форма задания на курсовую работу
АКАДЕМИЯ МАРКЕТИНГА И СОЦИАЛЬНО-ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ - ИМСИТ (г. Краснодар)

Кафедра математики и вычислительных систем
УТВЕРЖДАЮ Зав. кафедрой

« ____ » _____ 20__ г.

З А Д А Н И Е на
курсовой проект

Студенту: _____ группы _____ курса
(Ф.И.О.) (№ группы и курса)

факультета _____
направления _____

(шифр и наименование)

Тема работы: _____

Содержание задания: _____

Объем работы:

а) пояснительная записка к работе _____ с.

б) задачи

Рекомендуемая литература: _____

Срок выполнения работы: с " ____ " _____ по " ____ " _____ 20__ г.

Срок защиты: " ____ " _____ 20__ г.

Дата выдачи задания: " ____ " _____ 20__ г.

Дата сдачи проекта на кафедру: " ____ " _____ 20__ г.

Руководитель работы _____
(подпись, ф.и.о., звание, степень)

Задание принял студент _____
(подпись, дата)

ПРИЛОЖЕНИЕ В

(справочное)

Примеры библиографических описаний документов

Примеры библиографических описаний книг

- 1) одного, двух или трех авторов однотомного издания Шандала, М. Г. Охрана и оздоровление окружающей среды в условиях научно-технической революции [Текст] / М. Г. Шандала. - М. : Наука, 1982. - 75 с.
Исаев, Л. К. Обеспечение качества. Стандартизация, единство измерений, оценка соответствия [Текст] / Л. К. Исаев, В. Д. Малиновский ; под общ. ред. Л. К. Исаева. – М. : ИПК Изд-во стандартов, 2001. – 274 с.
Бахвалов, Н. С. Численные методы [Текст] : учеб. пособие для физ.-мат. специальностей вузов / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кулаков ; под ред. Н. И. Тихонова. – 2-е изд. – СПб. : Невский диалект, 2002. – 630 с.
- 2) четырех и более авторов однотомного издания Проектирование систем автоматизации технологических процессов [Текст] : справочное пособие / А. С. Ключев [и др.] ; под ред. А. С. Ключева. – М. : Энергоатомиздат, 1990. - 464 с.
- 3) коллективного автора однотомного издания "Внутривузовские системы обеспечения качества подготовки специалистов", всероссийская конференция (2002 ; Краснодар). Всероссийская конференция "Внутривузовские системы обеспечения качества подготовки специалистов", 2002 г. [Текст] / редкол. : В. Т. Иванченко [и др.] – Краснодар : Изд-во КубГТУ, 2002. – 125 с.
- 4) одного автора –отдельного тома из многотомного издания Савельев, И. В. Курс общей физики [Текст]. В 3 т. Т. 1. Механика. Молекулярная физика: учеб. пособие для студентов вузов / И. В. Савельев. -2-е изд., перераб. - М. : Наука, 1982. - 432 с.
- 5) законодательные материалы

Трудовой кодекс Российской Федерации [Текст]. - М. : Наука, 2002. – 223 с.

Примеры библиографических описаний технических документов

Правила безопасности при обслуживании гидротехнических сооружений [Текст] : РД 153-34.0-03.205-2001 : утв. М-вом энергетики Рос. Федерации 13.04.01 : введ. 01.11.01. – М. : ЭНИАС, 2001. – 158 с.

Оптовые цены на редукторы и муфты соединительные [Текст] : прейскурант № 19-08 : утв. Госкомцен РФ 12.08.90 : введ. 01.01.91. – М. : Прейскурантиздат, 1990. – 60 с.

Оборудование классных комнат общеобразовательных школ [Текст] : каталог / М-во образования РФ, Моск. гос. пед. ун-т. – М. : МГПУ, 2002. – 235 с.

Примеры библиографических описаний диссертационных и патентных материалов

Вишняков, И. В. Модели и методы оценки коммерческих банков в условиях неопределенности [Текст] : дисс. ... канд. экон. наук : 08.00.13 : защищена 12.02.02 : утв. 24.06.02 / Вишняков Илья Викторович. – М. , 2002. – 234 с.

Пат. 2187888 Российская Федерация, МПК7 Н 04 В 1/38. Приемопередающее устройство [Текст] Чугаева В. И. ; заявитель и патентообладатель Воронеж. науч.-исслед. ин-т связи. - № 2000131763/09 ; заявл. 18.12.00 ; опубл. 20.08.02 , Бюл. № 22. – 3 с.

А. с. 107970 СССР, МКИЗ В 25 J 15/00. Устройство для захвата деталей [Текст] / В. С. Ваутин, В. Г. Кемайкин (СССР). - № 3366585/25 – 08 ; заявл. 23.11.81 ; опубл. 30.03.83, Бюл. № 12. – 2 с.

Примеры библиографических описаний составных частей документов

1) статья из книги или другого разового издания Пшуков, Ю. Г. О нормировании качества жидких электролитов

[Текст] / Ю. Г. Пшуков // Научно-технический прогресс и оптимизация технологических процессов: тез. докл. Всесоюз. науч. конф. , 21 - 22 мая 1987 г. – Тамбов. 1987.— С. 200 - 205.

2) статьи из сериального издания Богатова, Е. Г. Актуальные вопросы обучения студентов технических

вузов [Текст] / Е. Г. Богатова // Сб. науч. тр. Моск. гос. пед. ин-та. Вып. 18, Педагогика высшей школы.— 1987.—С. 33 - 47.

Спиридонов, Э.А. Разработка системы качества вуза [Текст] / Э. А. Спиридонов, Е. В. Савина // Стандарты и качество. – 2002. - № 4. – С. 46 – 47.

Пример библиографического описания методических указаний Метрология и измерения [Текст] : программа и методические указания для студентов заочной формы обучения специальности 210100 - Управление и информатика в технических системах / Сост. В. В. Осокин ; Кубан. гос. технол. ун-т. Каф. автоматизации производственных процессов. – Краснодар: Изд-во КубГТУ, 2000. - 14 с.

Примеры библиографических описаний электронных ресурсов удаленного доступа

Российская Федерация. Законы. О техническом регулировании [Электронный ресурс] : федер. закон : [принят Гос. Думой 27 декабря 2002 г.]. Режим доступа : <http://www.gost.ru>.

Исследования в России [Электронный ресурс] : многопредмет. науч. журн. / Моск. физ.-техн. ин-т. – электрон. журн. – Долгопрудный : МФТИ, 1998 - . - Режим доступа : <http://www.zhurnal.mipt.rssi.ru>.

ПРИЛОЖЕНИЕ Г
(обязательное) Пример
оформления реферата

РЕФЕРАТ

Курсовой проект 27 с., 13 рис., 6 табл., 5 источника, 3 прил.

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ, ТАБЛИЦА, ОТНОШЕНИЕ, ПОЛЕ, АТ-
РИБУТ, ПРИЛОЖЕНИЕ, БАЗА ДАННЫХ, ФОРМА, ОТЧЁТ

Объектом исследования является база данных для подведения итогов спортивных соревнований.

Цель работы состоит в приобретении навыков построения модели данных, их практическом применении в построении базы данных средствами СУБД и разработке приложения базы данных в соответствии с темой.

К полученным результатам относятся база данных для подведения итогов спортивных соревнований и программное обеспечение для автоматизации управления данными. Теоретическая часть работы оформлена в виде пояснительной записки.

Новизна результатов заключается в построенной модели данных, в полной мере отражающей сведения об итогах спортивных соревнований. Достоверность сведений обеспечивается соблюдением целостности данных, предусмотренной в модели.

Областью применения системы являются соревнования и турниры спортивных команд.

Эффективность работы характеризуется простотой интерфейса и автоматическим формированием списка призёров на предварительном этапе соревнований.