

ФГБОУ ВО «Воронежский государственный технический
университет»
Кафедра компьютерных интеллектуальных технологий
проектирования

14-2016

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению курсового проекта по дисциплине
“Объектно-ориентированное программирование” для студентов
направления 09.03.01 «Информатика и вычислительная
техника» (все профили) очной формы обучения



Воронеж 2016

Составитель: канд. техн. наук А.Н. Юров

УДК 004.9

Методические указания к выполнению курсового проекта по дисциплине “Объектно-ориентированное программирование” для студентов направления 09.03.01 «Информатика и вычислительная техника» (все профили) очной формы обучения/ ФГБОУ ВО «Воронежский государственный технический университет»; сост. А.Н. Юров. Воронеж, 2016. 23 с.

Методические указания содержат материал по разработке программных решений с графическим интерфейсом и выполнению курсового проекта по дисциплине «Объектно-ориентированное программирование».

Предназначены для студентов 2 курса.

Методические указания подготовлены в электронном виде в текстовом редакторе MS Word 2010 и содержатся в файле МУ 2016_ООПК.docx.

Ил. 1. Библиогр.: 9 назв.

Рецензент канд. физ.-мат. наук, доц. Н.А. Тюкачев

Ответственный за выпуск зав. кафедрой д-р техн. наук, проф. М.И. Чижов

Издается по решению редакционно-издательского совета Воронежского государственного технического университета

© ФГБОУ ВО «Воронежский
государственный технический
университет», 2016

Введение

Разработка целевых программных решений всегда являлась востребованной продукцией для предприятий и организаций, нуждающихся в современном, практичном и функциональном решении для повседневной работы. Однако правильно разработать программный продукт, учесть все потребности заказчика и реализовать проект в кратчайшие сроки является достаточно трудоемкой задачей. В методических указаниях приводятся базовые сведения, которые знакомят студентов с отдельной группой вопросов, возникающих при разработке и позволяют сформировать базовые навыки для последующей профессиональной деятельности.

Подготовка студентов базируется на теоретическом материале и практическом выполнении курсовых проектов согласно выданному индивидуально техническому заданию. Реализация проектов позволит усвоить концепции объектно-ориентированного программирования на практике.

Подготовка к курсовому проектированию

Целью курсового проекта является приобретение учащимися практических навыков проектирования, отладки, тестирования и документирования программных решений с графическим интерфейсом согласно заданию.

Работа над курсовым проектом является завершающим этапом изучения дисциплины “Объектно-ориентированное программирование”, которая позволяет принять решение об усвоении студентом теоретических основ изученной дисциплины, применению полученных знаний для решения поставленных задач при разработке программного обеспечения. Курсовой проект - это полностью самостоятельная работа учащихся, опыт работы, приобретенный на данном этапе обучения, должен быть использован при подготовке выпускного дипломного проекта.

Задание на курсовой проект выдается преподавателем и утверждается на заседании кафедры “Компьютерные интеллектуальные технологии проектирования”.

Студентом может быть выбрана иная тематика курсового проекта. В этом случае задание должно быть представлено в письменной форме с четкой формулировкой (содержательной и математической) существа задачи, описания входных и выходных данных, требованиями по оформлению готовой программы и документации, списком рекомендуемой литературы и рассмотрено на заседании кафедры.

Учащийся, выполняющий курсовой проект, должен представить в письменном виде постановку задачи.

Постановка задачи должна содержать:

- содержательную и, по возможности, математическую формулировку решаемой задачи;

- необходимые характеристики разрабатываемой программы;

- требования к взаимодействию с программой;
- описание возможной реакции программы на отклонение в структуре исходных данных, а также наличие ошибок со стороны пользователя в процессе эксплуатации;
- способ тестирования готового программного решения.

Письменно сформулированная тематика проекта будет препятствовать изменениям в процессе разработки программного решения от ее оговоренных требований, сделает возможным обсуждение, оценку или согласованную с руководителем работы корректировку отдельных требований в ходе разработки программы. Кроме того, позволит избежать таких ошибок и противоречий, которые могут быть обнаружены только после разработки большей части программного решения или уже на стадии анализа полученных результатов выполнения программы.

Разработка технического задания

Прежде чем перейти к этапу создания проекта программы необходимо четко сформулировать основные цели разработки, требования к программному продукту, определить сроки и этапы разработки программного продукта, т. е. составить техническое задание.

Техническое задание должно содержать следующие разделы:

- введение, которое содержит краткую характеристику области применения программного средства, а также объекта, в котором предполагается их использовать;
- основание для разработки, содержит наименование документа (листа выдачи задания преподавателем), на основании которого ведется разработка программы;
- назначение разработки;
- требование к программе или программному средству, включает разделы:

а) требования к функциональным характеристикам, перечисляются выполняемые функции и описывается состав, характеристики и форма представления исходных данных и результатов;

б) требования к надежности и какими способами будут указанные результаты достигнуты;

в) требования к составу и параметрам технических средств (персональных ЭВМ, ноутбуков, планшетов, мобильных устройств, т.д.), указываются основные технические характеристики (тип микропроцессора, объем памяти, наличие внешних накопителей).

г) требования к информационной и программной совместимости, задаются методы решения, язык и среда интерактивной разработки, используемая операционная система.

-требования к программной документации (электронного руководства пользователей);

-стадии и этапы разработки, указываются сроки разработки.

Базовый пример разработки технического задания

Разработать техническое задание на программный продукт: “Создание табличного процессора”.

Разрабатываемая программа должна обрабатывать входящие записи, состоящие из чисел, производить базовые арифметические операции, представлять данные в табличном виде, управлять размером шрифтов, сохранять результаты работы в файл и иметь доступ к данным посредством их извлечения из сохраненного файла.

1. Состояние вопроса

Настоящее техническое задание распространяется на разработку табличного процессора, предназначенной выполнять базовые операции по обработке числовых и

строковых данных. Программа может быть использована в качестве учебной для студентов первого курса в рамках дисциплины “Информатика”, раздел офисное программное обеспечение. Студенты должны освоить базовые операции по обработке данных, уметь их вносить в программу и сохранять результаты работы на носитель информации. Существующее программное обеспечение (ПО), которое может решать подобные задачи, является универсальным, например, Microsoft Excel, Libre Calc. ПО имеет сравнительно сложный пользовательский интерфейс, ориентированный на пользователя с значительной практикой работы в указанных программах. При изучении раздела по работе с офисным программным обеспечением на начальном этапе у учащихся данный навык практической работы, как правило, отсутствует. Разрабатываемая программа позволит учащимся проверить свои знания при изучении указанной темы.

2. Основание для разработки

Программа разрабатывается на основе учебного плана кафедры «Компьютерные интеллектуальные технологии проектирования»

3. Назначение

Основным назначением программы является помощь учащимся при изучении раздела «Офисное программное обеспечение» по дисциплине “Информатика”.

4. Требования к программе или программному изделию

4.1. Требования к функциональным характеристикам

4.1.1. Программа должна обеспечивать возможность выполнения следующих функций:

- ввод данных в отдельном диалоговом окне и длительное хранение данных в системе;

- визуальное представление результатов в многодокументной (MDI) архитектуре приложения;

- иметь возможность редактировать введенные значения, заполнять автоматически указанные поля предопределенными данными;

-сохранять результаты вычислений в файле на указанном носителе информации.

4.1.2. Исходные данные:

-внесение значений в программу (числа указываются в заданном диапазоне значений);

-операция обработки данных (например, суммирование всех значений второй строки таблицы);

-чтение данных с носителя информации и автоматическое заполнение таблиц в приложении.

4.2. Требования к надежности

4.2.1. Предусмотреть контроль вводимой информации.

4.2.2. Предусмотреть блокировку некорректных действий пользователя при работе с системой.

4.3. Требования к составу и параметрам технических средств.

4.3.1. Система должна работать на IBM совместимых персональных компьютерах.

4.3.2. Минимальная конфигурация:

-тип процессора Pentium4 и выше, а также процессоры, совместимые с Intel архитектурой;

-объем оперативного запоминающего устройства 512 Мб и более.

4.4. Требования к информационной и программной совместимости

Система должна работать под управлением семейства операционных систем Windows x86 или Windows 64 (Windows XP, 7,8, 10).

5. Требования к программной документации

5.1. Разрабатываемые программные модули должны быть самодокументированы, т. е. тексты программ должны содержать все необходимые комментарии.

5.2. Разрабатываемая программа должна включать справочную информацию об основных терминах соответствующего раздела по работе с электронными таблицами и подсказки учащимся.

5.3. В состав сопровождающей документации должны входить:

5.3.1. Пояснительная записка на 25–30 листах, содержащая описание разработки.

5.3.2. Руководство пользователя.

Теоретическое обоснование разработки

На основании анализа формализованной постановки задачи выбирается основной метод ее решения (если он не оговорен на предыдущем этапе) и составляется общий проект программы. Выбранный подход к решению задачи должен обеспечивать правильный результат для тех условий функционирования программы, которые предусмотрены постановкой задачи, удобство использования программы, простоту подготовки исходных данных и т. п.

Помимо формулировки выбранного метода решения задачи в проекте должны описываться основные части разрабатываемой программы, их функции, взаимосвязь и последовательность выполнения, а также точно определяться входные и выходные данные, как всей программы, так и ее частей. На данном этапе разрабатывается и проект инструкции для пользователей, в котором фиксируется (и, таким образом, может быть заранее оценен и исправлен) предполагаемый режим обращения пользователя (и оператора) с программой.

С целью исключения в разрабатываемой программе ошибок следует разработать на данном этапе и основную стратегию отладки, общий план ее проведения, необходимо решить, каким путем можно будет в дальнейшем убедиться, что выбранный метод решения задачи и разработанная программа обеспечивают получение правильных результатов для указанных в постановке задачи диапазонов изменения величин и условий функционирования программы. Для этого разрабатываются общая методика проверки правильности программы и система необходимых для отладки контрольных примеров.

Алгоритмизация программного решения

Данный этап может пропускаться или объединяться с предыдущим в случае простых программ, но в случае достаточно сложных программ качественное выполнение этапа алгоритмизации может существенно облегчить выполнение следующего за ним этапа программирования, особенно в случаях, когда программист-разработчик недостаточно освоил язык, на котором предстоит программировать.

На данном этапе с помощью блок-схем или других специальных языков отображения алгоритмов (таблица решений, схем и т. д.) представляются общая структура программы и алгоритмы выполнения отдельных частей (блоков) программы.

Для достаточно сложных программ алгоритмизация проводится в несколько шагов с целью постепенной детализации алгоритма. Критерием окончания детализации некоторого блока (блоков) алгоритма является формирование у программиста мысленного представления о том, как данный блок алгоритма может быть реализован средствами данного языка программирования.

Рекомендуется размещать каждую блок-схему на одном стандартном листе бумаги.

После каждого шага детализации проводится проверка полученного алгоритма для выявления возможных ошибок. При этом, возможно, придется изменить и уточнить решения, принятые на этом и предыдущих этапах.

Все эти изменения обязательно должны вноситься в проект программы.

На этапе алгоритмизации следует предусмотреть способы контроля отдельных блоков и методы предстоящей локализации возможных ошибок.

Для этого следует подготовить контрольные примеры, в блок-схемах алгоритмов наметить места, моменты и объемы

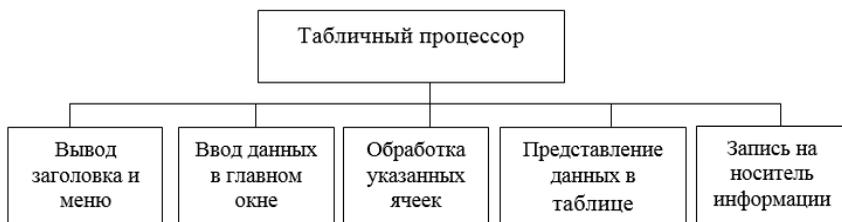
необходимой отладочной печати, которые должны обеспечить возможность быстрой локализации ошибок при отладке.

Разработка структурной и функциональной схем

Структурная схема отражает состав и взаимодействие по управлению частей разрабатываемого программного продукта.

Структурные схемы пакетов программ не информативны, поскольку организация программ в пакеты не предусматривает передачи управления между ними. Поэтому структурные схемы разрабатываются для каждой программы пакета, а список программ пакета определяют, анализируя функции, указанные в техническом задании.

Самый простой вид программного обеспечения – программа, которая в качестве структурных компонентов может включать только подпрограммы и библиотеки ресурсов. Разработку структурной схемы выполняют методом пошаговой детализации.



Структурная схема программы “Табличный процессор”

Метод пошаговой детализации реализует нисходящий подход (разработка «сверху-вниз») и базируется на основных конструкциях структурного программирования (линейная, разветвленная и циклическая). Он предполагает пошаговую разработку алгоритма. Каждый шаг при этом включает разложение функции на подфункции. Так на первом этапе описывают решение поставленной задачи, выделяя общие

подзадачи, на следующем аналогично описывают решение подзадач, формулируя при этом подзадачи следующего уровня. Таким образом, на каждом шаге происходит уточнение функций проектируемого программного обеспечения. Процесс продолжают, пока не доходят до подзадач, алгоритмы решения которых очевидны.

Функциональная схема или схема данных (ГОСТ 19.701 – 90) – схема взаимодействия компонентов программного обеспечения с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств. Для изображения функциональных схем используют специальные обозначения, установленные стандартом (ГОСТ 19.701 – 90).

Все компоненты функциональных схем должны быть описаны. При структурном подходе особенно тщательно необходимо прорабатывать спецификации межпрограммных интерфейсов, так как от качества их описания зависит количество самых проблемных ошибок. К самым проблемным относятся ошибки, обнаруживаемые при комплексном тестировании, так как для их устранения могут потребоваться серьезные изменения уже отлаженных листингов программ.

Программная разработка

Программирование по разработанным блок-схемам алгоритмов представляет собой традиционный этап в создании прикладных программ. Чем выше качество подготовленных блок-схем, тем более легким будет составление программы. Основными трудностями и, следовательно, причины ошибок на этом этапе заключаются, во-первых, в необходимости знания всех требований и ограничений выбранного языка программирования и, во-вторых, в необходимости постоянного внимания ко многим деталям языка, которые приходится учитывать в ходе написания программы. Кроме того, вследствие возможных ошибок, на этапе

алгоритмизации, на этапе программирования наряду с реализацией системы осуществляется логический контроль правильности ранее принятых решений. При обнаружении ошибок такого рода следует внести соответствующие изменения в алгоритм, поскольку к нему придется обращаться на следующих этапах, и тексты алгоритма и программы должны соответствовать друг другу. Изменяя некоторые части алгоритма, всегда следует проверять, не окажут ли внесенные коррективы влияние на выполнение других блоков и частей.

После исправления программы проводится ее проверка для обнаружения и исправления ошибок, внесенных на данном этапе. Следует помнить об обязательной реализации плана и средства отладки, намеченных на предыдущих этапах.

Чем более тщательно выполнены все предыдущие этапы работы, и, в частности, чем более детально разработан план тестирования программного решения, чем больше внимания было уделено проверке составленного алгоритма и программы, а также применению отладочных средств для облегчения предстоящей отладки, тем меньше времени потребуется на проведение самой отладки и получение корректных результатов.

Тестирование программного решения

На этом этапе производится обнаружение с помощью ЭВМ ошибок в программе и их исправление. Здесь можно выделить следующие этапы:

- устранение синтаксических ошибок, обнаруженных транслятором;

- контроль правильности выполнения программы путем пропуска на ЭВМ подготовленного на предыдущих этапах специальных примеров (тестов) и сравнение промежуточных и окончательных результатов работы программы с полученными при подготовке тестов. Таким способом определяются содержательные (семантические) ошибки, которые не проявляются при трансляции программы;

-локализация ошибок, т. е. точное определение места, где в программе допущена ошибка, выявленная на предыдущем этапе;

-исправление ошибок. Исправления вносятся как в программу, так и в алгоритм.

Руководство пользователя

Руководство пользователя должно быть написано на доступном языке и излагаться ясно, используя короткие предложения.

Руководство пользователя должно содержать следующие разделы:

-общие сведения о программном продукте, содержит краткое описание его функций, реализованные методы и возможные области применения;

-порядок развёртывания программного средства из дистрибутива, создание ссылочных ярлыков, добавление виртуальных машин (при необходимости) и т.д.;

-описание запуска, сообщения, которые при этом могут быть получены;

-инструкция по работе;

-информационные сообщения пользователю.

Содержание пояснительной записки

Пояснительная записка объемом в 25-30 листов машинописного текста с иллюстрациями по курсовому проекту должна содержать:

-техническое задание;

-содержательную и формализованную постановку задачи;

-обоснование выбранного метода решения задачи и проекта системы;

-блок-схемы алгоритмов для каждого уровня детализации с необходимыми пояснениями;

-тексты программ;

-контрольный пример, демонстрирующий правильность выполнения программы в различных режимах ее функционирования и с различными вариантами типов исходных данных;

-руководство пользователя;

-список литературы.

Основную часть составляют материалы, с которыми шла работа на этапах проектирования, алгоритмизации и программирования (проект разработки, общая блок-схема, алгоритмы и т. д.). Поэтому все перечисленные материалы должны быть в рабочем состоянии и по содержанию соответствовать друг другу в отлаженной программе. Кроме того, уже на этапах разработки их нужно представить в таком виде, чтобы они могли быть использованы при составлении пояснительной записки без дополнительных переделок.

Оформление курсового проекта

Документирование программного обеспечения осуществляется в соответствии с Единой системой программной документации (ЕСПД).

Курсовой проект состоит из задания, пояснительной записки и разработанной программы.

Содержание пояснительной записки:

Титульный лист (приложение А)

Задание на курсовой проект (приложение Б)

Содержание пояснительной записки (приложение В)

Защита курсового проекта

Осуществляется оценка курсового проекта по следующим разделам:

-оценка готового проекта;

-оценка на этапе программирования;

- оценка функциональных параметров программы (объем используемой памяти, быстродействие и т. д.);
- оценка возможностей разработанной программы;
- оценка удобства эксплуатации и возможностей сопровождения программы;
- оценка надежности программы;
- оценка качества руководства пользователя, инструкции к внедрению и применению, прочая справочная информация.

Кроме того, учитывается соблюдение графика выполнения курсового проекта и соответствие подготавливаемых автором работы документов действующим стандартам. При защите курсового проекта учащийся должен показать значение основных стадий и этапов технологического процесса разработки программ и программных систем, современных технологий программирования, методов отладки и тестирования программ, состав и назначения проектной и эксплуатационной документации на программное изделие.

Приложение Б
Задание для курсового проектирования

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГТУ»)
КАФЕДРА КОМПЬЮТЕРНЫХ ИНТЕЛЛЕКТУАЛЬНЫХ
ТЕХНОЛОГИЙ ПРОЕКТИРОВАНИЯ

ЗАДАНИЕ

На курсовой проект по предмету
«Объектно-ориентированное программирование»

Тема работы «Разработка табличного процессора для решения
вычислительных задач»

Студент Иванов Андрей Юрьевич группа ИВТ-151
Фамилия, имя, отчество номер группы

Технические условия: среда программирования – Qt
Creator+QT SDK 5.6, язык проектирования C++, ЭВМ,
удовлетворяющая по характеристикам требованиям
программного обеспечения

Руководитель _____ А.Н. Юров _____
Подпись, дата Инициалы, фамилия

Задание принял студент _____ А.Ю. Иванов _____
Подпись, дата Инициалы, фамилия

Приложение В

Состав проекта

Пояснительная записка на 20-25 листах.

Содержание пояснительной записки

Введение (должно отображать актуальность разработанной программы)

1 Обоснование разработки

1.1 Описание предметной области

1.2 Обзор элементов языка программирования

1.3 Среда и базовые шаблоны приложения

2 Анализ данных

2.1 Общая постановка задачи

2.2 Цель и задачи решения

2.3 Описание программ-аналогов, если такие присутствуют

2.4 Теоретические сведения и математическое описание материала (при наличии)

3 Проектирование

3.1 Структурная и функциональная схема

3.2 Описание модулей

3.3 Таблица идентификаторов

3.4 Описание наборов данных (в бд)

3.5 Структура записей файлов (если есть)

3.6 Описание процесса создания интерфейса

4. Реализация

4.1 Описание методов реализации графического интерфейса

4.2 Описание расчетных методов

4.3. Диалоговые сообщения и прочие вспомогательные решения

5. Тестирование

5.1 Работа с отладочными средствами в проекте

- 5.2 Разработка набора тестовых заданий для отладки приложения и анализ результатов
- 6. Виды обеспечения
 - 6.1 Требования к ЭВМ
 - 6.2 Руководство пользователя
- Заключение
- Список литературы
- Приложения А, В, С (Листинг программы)

Описание некоторых разделов проекта

Введение

Введение должно отражать основные задачи, стоящие перед отраслью информационных технологии. Перспективы её развития на ближайшие годы и место специалиста-программиста в этих процессах. Во введении необходимо мотивировать необходимость разработки программы, указать и аргументировать возможность пути и способы решения поставленной задачи. Объем введения 1,5-2 страниц.

Обоснование разработки

Этот раздел должен дать полное представление о теоретической части исследуемой задачи. Для заданий, связанных с решением задач из курса «Численные методы» в этом разделе необходимо дать подробное описание решаемой задачи и ее математическую модель со всеми необходимыми формулами и определениями. Для других заданий в данном разделе описываются общие теоретические основы той области, в которую входит поставленная задача (базы данных, устройства ввода- вывода, ранение информации и т.п.). Объем данного раздела 3-5 страниц.

Обзор элементов языка программирования

В данном разделе учащиеся описывают те конструкции языков С/С++, которые планируется использовать при разработке программы. Например: модули, графика, работа с

экраном, работа с файлами, записи, указатели и т.п. Объем данного раздела 3-5 страниц.

Общая постановка задачи

В процессе постановки задачи четко формируют назначение программного обеспечения и определяют основные требования к нему. Каждое требование представляет собой описание необходимого или желаемого свойства программного обеспечения. Различают функциональные требования, определяющие функции, которые должны выполнять разрабатываемое программное обеспечение, и эксплуатационные требования, определяющие особенности его функционирования.

Требования к программному обеспечению, имеющему прототипы, обычно определяют по аналогии, учитывая структуру и характеристики уже существующего программного обеспечения. Для формулирования требований к программному обеспечению, не имеющему аналогов, иногда необходимо провести специальные исследования, называемые предпроектными. В процессе таких исследований определяют разрешимость задачи, возможно, разрабатывают методы ее решения (если они новые) и устанавливает наиболее существенные характеристики разрабатываемого программного обеспечения. Для выполнения предпроектных исследований, как правило, заключают договор на выполнение научно-исследовательских работ. В любом случае этап постановки задачи заканчивается разработкой технического задания, фиксирующего принципиальные требования, и принятые основных проектных решений.

Приложение Г

Пример оформления постановки задачи

Общая постановка задачи

Согласно техническому заданию разрабатываемая программа предназначена для расчета методом _____ и должна выполнять следующие действия: ввод исходных данных (с клавиатуры, из файла или случайным образом), расчет системы уравнений и вывод результатов расчета в графическом режиме. Кроме того, для удобства использования выбор выполняемого действия должно быть организовано в виде меню. Программа должна начинаться с заставки в которой отображается тема курсового проекта и автор программы. В меню необходимо предусмотреть получение справки по работе с программой и корректный выход при завершении работы. Для того чтобы уберечь программу от некорректных действий пользователя при работе с системой, нужно предоставить контроль вводимой информации и блокировку на те самые некорректные данные.

Результаты расчета в текстовом виде записываются в файл, а в графическом – выводятся на экран монитора.

Программа должна работать на IBM совместных персональных компьютерах. Если говорить о типе процессоров, им должен быть Pentium 4 и выше, а объем запоминающего устройства от 100 Мб. Тип видеоадаптера-SVGA с поддержкой OpenGL 4.0.

Единственное требование к информационной программной совместимости, это то, что программа должна работать под управлением операционных систем семейства Window. Четко определены требования к программной документации. Разработанная система должна включать справочную информацию о работе системы и подсказка пользователю т.к. пользователь не должен быть программистом. В состав сопровождающей документации должны входить: пояснительная записка и руководство пользователя.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки 230100 Информатика и вычислительная техника (квалификация (степень) «бакалавр»). Утвержден приказом Министерства образования и науки Российской Федерации от 09 ноября 2009 г. № 553. Москва, 2009.

2. ГОСТ 7.1–2003. Библиографическая запись. Библиографическое описание. – Взамен ГОСТ 7.1–84 : введ. 2004–07–01. – М. : Изд-во стандартов, 2001. – 47 с.

3. ГОСТ Р 7.0.12–2011. Библиографическая запись. Сокращение слов и словосочетаний на русском языке. Общие требования и правила. – Взамен ГОСТ 7.12–77 : введ. 1995–07–01. Взамен ГОСТ 7.12–93. – М. : Стандартинформ, 2012. – 17 с.

4. ГОСТ 7.82–2001. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления. – Введ. 2002-07-01. – Минск : Межгос. Совет по стандартизации, метрологии и сертификации, 2001. – 23 с.

5. ГОСТ 2.105–95. Общие требования к текстовым документам. – Взамен ГОСТ 2.105–79, ГОСТ 2.906–71 : введ. 1996–07–01. – Минск : Межгос. Совет по стандартизации, метрологии и сертификации ; М. : Изд-во стандартов, 1996. – 28 с.

6. ГОСТ Р 6.30–2003. Требования к оформлению документов. – Взамен ГОСТ 6.30–97 : введ. 2003–07–01. – М.: Изд-во стандартов, 2003. – 16 с.

7. Методические указания к лабораторным работам № 1-3 по дисциплине “Объектно-ориентированное программирование” для студентов направления 230100.62 «Информатика и вычислительная техника» очной формы обучения / ФГБОУ ВПО «Воронежский государственный технический университет»; сост.

А.Н. Юров, М.В. Паринов, В.А. Рыжков, А.А. Филимонова. Воронеж, 2014. 23 с.

8. Методические указания к лабораторным работам № 4-8 по дисциплине “Объектно-ориентированное программирование” для студентов направления 230100.62 «Информатика и вычислительная техника» очной формы обучения / ФГБОУ ВПО «Воронежский государственный технический университет»; сост. А.Н. Юров, М.В. Паринов, В.А. Рыжков, А.А. Филимонова. Воронеж, 2014. 36 с.

9. Методические указания к лабораторным работам № 9-13 по дисциплине “Объектно-ориентированное программирование” для студентов направления 230100.62 «Информатика и вычислительная техника» очной формы обучения / ФГБОУ ВПО «Воронежский государственный технический университет»; сост. А.Н. Юров, М.В. Паринов, В.А. Рыжков, А.А. Филимонова. Воронеж, 2014. 41 с.

СОДЕРЖАНИЕ

Введение	1
Подготовка к курсовому проектированию	2
Разработка технического задания	3
Базовый пример разработки технического задания	4
Теоретическое обоснование разработки.....	7
Алгоритмизация программного решения.....	8
Разработка структурной и функциональной схем	9
Программная разработка.....	10
Тестирование программного решения.....	11
Руководство пользователя	12
Содержание пояснительной записки	12
Оформление курсового проекта	13
Защита курсового проекта	13
Приложение А	15
Приложение Б.....	16
Приложение В	17
Приложение Г.....	20
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	21

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению курсового проекта по дисциплине
“Объектно-ориентированное программирование”
для студентов направления 09.03.01 «Информатика
и вычислительная техника» (все профили)
очной формы обучения

Составитель:
Юров Алексей Николаевич

В авторской редакции

Компьютерный набор А.Н. Юрова

Подписано к изданию 01.03.2016.
Уч.-изд. л. 1,4. «С»

ФГБОУ ВО «Воронежский государственный технический
университет»
394026 Воронеж, Московский просп., 14