

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САХАЛИНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
для студентов по выполнению
практических занятий
по профессиональному модулю 03
**УЧАСТИЕ В ИНТЕГРАЦИИ ПРОГРАММНЫХ
МОДУЛЕЙ**

МДК.03.03. ДОКУМЕНТИРОВАНИЕ И СЕРТИФИКАЦИЯ

09.00.00 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
по специальности

09.02.03 Программирование в компьютерных системах
(базовый уровень подготовки)

Квалификация: техник-программист
Форма обучения: очная

Южно-Сахалинск
2014

УТВЕРЖДАЮ

Зам. директора по УР

И.М. Ким

« 25 » сентября 2014 г.

Разработчик(и):

Чехонина С.А., зам. директора по
информационным технологиям

(указать Ф.И.О., должность)

Одобрено на заседании ПЦК

информационных
технологий

Протокол № 1 от «25» сентября 2014 г.

Председатель ПЦК

Савенкова О.В.

Согласовано

[подпись]
подпись

Курочкин
Ф.И.О.

зам. отделением информационных

СОДЕРЖАНИЕ

ПРАКТИЧЕСКАЯ РАБОТА №1 Написание руководства пользователя заданной программы.....	5
Цель работы	5
Краткие сведения из теории	5
Задания к работе	16
Контрольные вопросы	16
ПРАКТИЧЕСКАЯ РАБОТА №2 Написание руководства системного программиста (администратора) заданной программы	17
Цель работы	17
Краткие сведения из теории	17
Задания к работе	21
ПРАКТИЧЕСКАЯ РАБОТА №3 Разработка руководства по сопровождению ПО.....	22
Цель работы	22
Краткие сведения из теории	22
Задание к работе	23
ПРАКТИЧЕСКАЯ РАБОТА №4 Разработка справочной системы программного продукта.....	25
Цель работы	25
Краткие сведения из теории	25
Задания к работе	27
ПРАКТИЧЕСКАЯ РАБОТА №5 Разработка буклетов, рекламных листов по программному продукту	29
Цель работы	29
Краткие сведения из теории	29
Задание к работе	30
ПРАКТИЧЕСКАЯ РАБОТА №6 Тестирование по методу «черного ящика».....	31
Цель работы	31
Краткие сведения из теории	31
Задание к работе	36
ПРАКТИЧЕСКАЯ РАБОТА №7 Тестирование по методу «белого ящика».....	38

	4
Цель работы	38
Краткие сведения из теории	38
Задание к работе	39
Приложение 1	40
Приложение 2	42

ПРАКТИЧЕСКАЯ РАБОТА №1 Написание руководства пользователя заданной программы

Цель работы

- изучение нормативно правовой документации, регламентирующей разработку документации на программные средства.
- приобретение навыков разработки руководства пользователя программного средства (ПС).

Краткие сведения из теории

Основу отечественной нормативной базы в области документирования ПС составляет комплекс стандартов Единой системы программной документации (ЕСПД). Основная и большая часть комплекса ЕСПД была разработана в 70-е и 80-е годы. Стандарты ЕСПД в основном охватывают ту часть документации, которая создается в процессе разработки ПС, и связаны, по большей части, с документированием функциональных характеристик ПС.

Согласно ЕСПД программный документ – это документ, содержащий сведения, необходимые для разработки, изготовления, эксплуатации и сопровождения программного изделия. Номенклатуру программных документов определяет **ГОСТ 19.101-77 «ЕСПД. Виды программ и программных документов»**. В качестве основных видов программ стандартом определяются:

- **компоненты** – программы, рассматриваемые как единое целое, выполняющие законченную функцию и применяемые самостоятельно или в составе комплекса;
- **комплексы** – программы, состоящие из двух или более компонентов, выполняющие взаимосвязанные функции и применяемые самостоятельно или в составе другого комплекса.

Виды программных документов и их краткое содержание представлены в стандарте описаниями, приведенными в таблице 1.

Таблица 1. Виды программных документов

Вид документа	Содержание документа
Спецификация	Состав программы и документация на нее
Ведомость держателей подлинников	Перечень предприятий, на которых хранятся подлинники программных документов
Текст программы	Запись программы с необходимыми комментариями
Описание программы	Сведения о логической структуре и функционировании программы
Программа и методика испытаний	Требования, подлежащие проверке при испытании программы, а также порядок и методы их контроля
Техническое задание	Назначение и область применения программы; технические, технико-экономические и специальные требования, предъявляемые к программе; необходимые стадии и сроки разработки; виды испытаний
Пояснительная записка	Схема алгоритма, общее описание алгоритма и (или) функционирования программы, а также обоснование принятых технических и технико-экономических решений
Эксплуатационные документы	Сведения для обеспечения функционирования и эксплуатации программы

Перечень эксплуатационных документов, рекомендуемых ЕСПД, представлен в табл. 2.

Таблица 2. Виды эксплуатационных документов

Вид документа	Содержание документа
Ведомость эксплуатационных документов	Перечень эксплуатационных документов на программу

Вид документа	Содержание документа
Формуляр	Основные характеристики программы, комплектность и сведения об эксплуатации программы
Описание применения	Сведения о назначении программы, области применения, применяемых методах, классе решаемых задач, ограничениях для применения, минимальной конфигурации технических средств
Руководство системного программиста	Сведения для проверки, обеспечения функционирования и настройки программы на условия конкретного применения
Руководство программиста	Сведения для эксплуатации программы
Руководство оператора (пользователя)	Сведения для обеспечения процедуры общения оператора с вычислительной системой в процессе выполнения программы
Описание языка	Описание синтаксиса и семантики языка
Руководство по техническому обслуживанию	Сведения для применения тестовых и диагностических программ при обслуживании технических средств

Допускается объединение отдельных видов эксплуатационных документов (за исключением ведомости эксплуатационных документов и формуляра), необходимость объединения указывается в техническом задании. Объединенному документу присваивают наименование и обозначение одного из объединяемых документов. В объединенных документах должны быть приведены сведения, которые необходимо включать в каждый объединяемый документ.

ГОСТ 19.701-90 (ИСО 5807-85) "Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения". Стандарт распространяется на условные обозначения (символы) в схемах алгоритмов, программ, данных и систем и устанавливает правила выполнения схем, используемых для отображения различных видов задач обработки данных и средств их решения.

В РФ действует ряд стандартов в части документирования ПС, разработанных на основе прямого применения международных стандартов ИСО.

ГОСТ Р ИСО/МЭК 9294-93 «Информационная технология. Руководство по управлению документированием программного обеспечения». Стандарт устанавливает рекомендации по эффективному управлению документированием ПС для руководителей, отвечающих за их создание. Целью стандарта является оказание помощи в определении стратегии документирования ПС; выборе стандартов по документированию; выборе процедур документирования; определении необходимых ресурсов; составлении планов документирования.

ГОСТ Р ИСО 9127-94 «Системы обработки информации. Документация пользователя и информация на упаковке для потребительских программных пакетов». В контексте настоящего стандарта под потребительским программным пакетом (ПП) понимается «программная продукция, спроектированная и продаваемая для выполнения определенных функций; программа и соответствующая ей документация, упакованные для продажи как единое целое». Под документацией пользователя понимается документация, которая обеспечивает конечного пользователя информацией по установке и эксплуатации ПП. Под информацией на упаковке понимают информацию, воспроизводимую на внешней упаковке ПП. Ее целью является предоставление потенциальным покупателям первичных сведений о ПП.

Содержание документа "Руководство пользователя"

Документ **"Руководство пользователя"**, разрабатывается на основании методических указаний РД 50-34.698-90.

Данный документ формируется ИТ-специалистом, или функциональным специалистом, или техническим писателем в ходе разработки рабочей документации на систему и её части на стадии «Рабочая документация».

Состав руководства пользователя в соответствии со стандартом:

1. Введение.
2. Назначение и условия применения.
3. Подготовка к работе.
4. Описание операций.
5. Аварийные ситуации.
6. Рекомендации по освоению.

1. Введение

В разделе "Введение" указывают:

- область применения;
- краткое описание возможностей;
- уровень подготовки пользователя;
- перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю.

1.1. Область применения

Требования настоящего документа применяются при:

- предварительных комплексных испытаниях;
- опытной эксплуатации;
- приемочных испытаниях;
- промышленной эксплуатации.

1.2. Краткое описание возможностей

Например:

Информационно-аналитическая система Корпоративное Хранилище Данных (ИАС КХД) предназначена для оптимизации технологии принятия тактических и стратегических управленческих решений конечными бизнес-

пользователями на основе информации о всех аспектах финансово-хозяйственной деятельности Компании.

ИАС КХД предоставляет возможность работы с регламентированной и нерегламентированной отчетностью.

1.3. Уровень подготовки пользователя

Например:

Пользователь ИАС КХД должен иметь опыт работы с ОС MS Windows (95/98/NT/2000/XP), навык работы с ПО Internet Explorer, Oracle Discoverer, а также обладать следующими знаниями:

- знать соответствующую предметную область;
- знать основы многомерного анализа;
- понимать многомерную модель соответствующей предметной области;
- знать и иметь навыки работы с аналитическими приложениями.

Квалификация пользователя должна позволять:

- формировать отчеты в Oracle Discoverer Plus;
- осуществлять анализ данных.

1.4. Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю

Например:

Информационно-аналитическая система «Корпоративное хранилище данных». ПАСПОРТ;

Информационно-аналитическая система «Корпоративное хранилище данных». ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ.

2. Назначение и условия применения

В разделе "Назначение и условия применения" указывают:

- виды деятельности, функции, для автоматизации которых предназначено данное средство автоматизации;
- условия, при соблюдении (выполнении, наступлении) которых обеспечивается применение средства автоматизации в соответствии с назначением (например, вид ЭВМ и конфигурация технических средств, операционная среда и общесистемные программные средства, входная информация,

носители данных, база данных, требования к подготовке специалистов и т. п.).

Например:

Oracle Discoverer Plus в составе ИАС КХД предназначен для автоматизации подготовки, настройки отчетных форм по показателям деятельности, а также для углубленного исследования данных на основе корпоративной информации хранилища данных.

Работа с Oracle Discoverer Plus в составе ИАС КХД возможна всегда, когда есть необходимость в получении информации для анализа, контроля, мониторинга и принятия решений на ее основе.

Работа с Oracle Discoverer Plus в составе ИАС КХД доступна всем пользователям с установленными правами доступа.

3. Подготовка к работе

В разделе "Подготовка к работе" указывают:

- состав и содержание дистрибутивного носителя данных;
- порядок загрузки данных и программ;
- порядок проверки работоспособности.

3.1. Состав и содержание дистрибутивного носителя данных

Например:

Для работы с ИАС КХД необходимо следующее программное обеспечение:

Internet Explorer (входит в состав операционной системы Windows);

Oracle JInitiator устанавливается автоматически при первом обращении пользователя к ИАС КХД.

3.2. Порядок загрузки данных и программ

Например:

Перед началом работы с ИАС КХД на рабочем месте пользователя необходимо выполнить следующие действия:

Необходимо зайти на сайт ИАС КХД ias-dwh.ru.

Во время загрузки в появившемся окне "Предупреждение о безопасности", которое будет содержать следующее: 'Хотите установить и выполнить "Oracle JInitiator" ...' Нажимаем на кнопку "Да".

После чего запуститься установка Oracle JInitiator на Ваш компьютер. Выбираем кнопку Next и затем OK.

3.3. Порядок проверки работоспособности

Например:

Для проверки доступности ИАС КХД с рабочего места пользователя необходимо выполнить следующие действия:

Открыть Internet Explorer, для этого необходимо кликнуть по ярлыку «Internet Explorer» на рабочем столе или вызвать из меню «Пуск».

Ввести в адресную строку Internet Explorer адрес: ias-dwh.ru и нажать «Переход».

В форме аутентификации ввести пользовательский логин и пароль. Нажать кнопку «Далее».

Убедиться, что в окне открылось приложение Oracle Discoverer Plus.

В случае если приложение Oracle Discoverer Plus не запускается, то следует обратиться в службу поддержки.

4. Описание операций

В разделе "Описание операций" указывают:

- описание всех выполняемых функций, задач, комплексов задач, процедур;
- описание операций технологического процесса обработки данных, необходимых для выполнения функций, комплексов задач (задач), процедур.

Для каждой операции обработки данных указывают:

- наименование;
- условия, при соблюдении которых возможно выполнение операции;
- подготовительные действия;
- основные действия в требуемой последовательности;
- заключительные действия;

- ресурсы, расходуемые на операцию.

В описании действий допускаются ссылки на файлы подсказок, размещенные на магнитных носителях.

4.1. Выполняемые функции и задачи

Например:

Oracle Discoverer Plus в составе ИАС КХД выполняет функции и задачи, приведенные в таблице ниже:

Функции	Задачи	Описание
Обеспечивает многомерный анализ в табличной и графической формах	Визуализация отчетности	В ходе выполнения данной задачи пользователю системы предоставляется возможность работы с выбранным отчетом из состава преднастроенных.
	Формирование табличных и графических форм отчетности	В ходе выполнения данной задачи пользователю системы предоставляется возможность формирования собственного отчета в табличном или графическом виде на базе преднастроенных компонентов.

4.2. Описание операций технологического процесса обработки данных, необходимых для выполнения задач

Например:

Задача: «Визуализация отчетности»

Операция 1: Регистрация на портале ИАС КХД

Условия, при соблюдении которых возможно выполнение операции:

- Компьютер пользователя подключен к корпоративной сети.
- Портал ИАС КХД доступен.
- ИАС КХД функционирует в штатном режиме.

Подготовительные действия:

- На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

- На иконке «ИАС КХД» рабочего стола произвести двойной щелчок левой кнопкой мышки.
- В открывшемся окне в поле «Логин» ввести имя пользователя, в поле «Пароль» ввести пароль пользователя. Нажать кнопку «Далее».

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

15-30 секунд.

5. Аварийные ситуации

В разделе "Аварийные ситуации" указывают:

1. действия в случае несоблюдения условий выполнения технологического процесса, в том числе при длительных отказах технических средств;
2. действия по восстановлению программ и/или данных при отказе магнитных носителей или обнаружении ошибок в данных;
3. действия в случаях обнаружении несанкционированного вмешательства в данные;
4. действия в других аварийных ситуациях.

Например:

В случае возникновения ошибок при работе ИАС КХД, не описанных ниже в данном разделе, необходимо обращаться к сотруднику подразделения технической поддержки ДИТ (HelpDesk) либо к ответственному Администратору ИАС КХД.

<i>Класс ошибки</i>	<i>Ошибка</i>	<i>Описание ошибки</i>	<i>Требуемые действия пользователя при возникновении ошибки</i>
<i>Сбой в электр</i>	<i>Нет электр</i>	<i>Рабочая станция</i>	<i>Перезагрузить рабочую станцию.</i>

опитан ии рабоче й станции и	итания рабочей станции или произоше л сбой в электрон итании.	выключил ась или перезагру зилась.	Проверить доступность сервера ИАС КХД по порту 80, выполнив следующие команды: - нажать кнопку «Пуск» - выбрать пункт «Выполнить» - в строке ввода набрать команду <code>telnet ias_dwh.ru 80</code> - если открылось окно <i>Telnet</i> , значит соединение возможно. Повторить попытку подключения (входа) в ИАС КХД
Сбой локальн ой сети	Нет сетевого взаимод ействия между рабочей станцией и сервером приложе ний ИАС КХД	Отсутст вует возможн ость начала (продолж ения) работы с ИАС КХД. Нет сетевого подключе ния к серверу ИАС КХД	Перезагрузить рабочую станцию. Проверить доступность сервера ИАС КХД по порту 80, выполнив следующие команды: - нажать кнопку «Пуск» - выбрать пункт «Выполнить» - в строке ввода набрать команду <code>telnet ias_dwh.ru 80</code> - если открылось окно <i>Telnet</i> , значит соединение возможно. После восстановления работы локальной сети повторить попытку подключения (входа) в ИАС КХД.

6. Рекомендации по освоению

В разделе "Рекомендации по освоению" указывают рекомендации по освоению и эксплуатации, включая описание контрольного примера, правила его запуска и выполнения.

Например:

Рекомендуемая литература:

Oracle® Business Intelligence Discoverer Viewer User's Guide

Oracle® Business Intelligence Discoverer Plus User's Guide

Рекомендуемые курсы обучения:

Discoverer 10g: Создание запросов и отчетов

В качестве контрольного примера рекомендуется выполнить операции задачи «Визуализация отчетности», описанные в п. 4.2. настоящего документа.

Задания к работе

1. Подготовить документ (*.doc), содержащий структуру основных разделов руководства пользователя стандартного форматирования: шрифт TimesNewRoman, 12 пт, поля, межстрочный интервал - стандартные, как в техническом задании, имя файла - <ФИО студента. Руководство пользователя>.

2. На основании технического задания на разработку (практическая работа МДК 03.01), заполнить разделы руководства пользователя "Введение", "Назначение и условия применения", "Подготовка к работе".

3. Сохранить документ с именем (Фамилия, инициалы студента. Наименование работы).

4. Прикрепить файл руководства пользователя в разделе Руководство пользователя (практическая работа 1) учебного сервера.

Контрольные вопросы

1. Перечислить состав разделов руководства пользователя.
2. Пояснить состав раздела «Введение».
3. Пояснить состав раздела «Назначение и условия применения2 применения».
4. Пояснить состав раздела «Подготовка к работе»
5. Пояснить состав раздела «Описание операций»
6. Пояснить состав раздела «Аварийные ситуации»
7. Пояснить состав подраздела «Рекомендации по освоению»

ПРАКТИЧЕСКАЯ РАБОТА №2 Написание руководства системного программиста (администратора) заданной программы

Цель работы

- изучение нормативно правовой документации, регламентирующей разработку документации на программные средства.
- приобретение навыков разработки руководства системного программиста ПС.

Краткие сведения из теории

Если программа более-менее проста, пользователь может установить ее себе на компьютер самостоятельно. Поддерживать ее работоспособность, например, выполнить переустановку, если возникнут какие-нибудь проблемы, он тоже, как правило, в состоянии.

Сложными, в том числе, серверными и сетевыми программными продуктами приходится заниматься квалифицированным специалистам, *системным администраторам*. Более того, во многих компаниях сотрудникам запрещено устанавливать на своих рабочих местах программы по своему усмотрению. Даже простые программы там ставит только системный администратор.

В обязанности системного администратора также входит поддержание работоспособности программ, используемых в рамках тех или иных систем. Эта деятельность может заключаться в периодической проверке логов, резервном копировании данных, замерах производительности, устранении различных технических проблем.

Руководство системного администратора — программный документ, предоставляющий специалисту информацию, необходимую для выполнения этой работы.

В ЕСПД специалист, сходный по обязанностям с современным системным администратором, называется *системным программистом*, а адресованный ему документ — руководством системного программиста.

Содержание документа "Руководство системного программиста (администратора)"

Если речь идет о сложных программных или аппаратно-программных комплексах, то системный программист (администратор) - это квалифицированный специалист, которому приходится принимать нетривиальные решения.

Чтобы удовлетворить его потребности в информации, составителю документации необходимо понимать, как мыслит его читатель, что и в какой момент может его заинтересовать, какая подробность изложения материала будет достаточной для него.

Поэтому разрабатывать руководство системного администратора должен либо его коллега, имеющий к тому же навыки технического писателя, либо технический писатель, имеющий хотя бы небольшой опыт работы системным администратором.

В случае небольших «монолитных» программ руководство системного администратора может оказаться документом небольшим по объему и простым по структуре.

Руководство системного администратора на программный или аппаратно-программный комплекс сложнее, поскольку в нем приходится описывать каждый компонент по отдельности и способы их интеграции как друг с другом, так и со сторонним программным обеспечением: серверами баз данных, почтовыми серверами, антивирусами, средствами шифрования и пр.

Итак, в руководстве системного администратора должны быть изложены:

- назначение и область применения программы (или комплекса);
- состав программы, основные принципы ее функционирования;
- комплект поставки (если он не указан в отдельном документе);

- системные требования для программы или ее компонентов;
- предпочтительная очередность установки компонентов;
- процедура установки программы или каждого ее компонента;
- порядок обязательной первоначальной настройки программы;
- способы интеграции установленных копий компонентов между собой;
- интеграция программы со сторонним ПО, например, с сервером БД;
- способы и периодичность контроля правильности работы программы;
- порядок текущего обслуживания работающих копий программы;
- порядок решения всевозможных вспомогательных задач;
- аварийные ситуации и способы их устранения.

Нередко в руководстве системного администратора вдобавок приходится описывать:

- пользовательский интерфейс административной консоли;
- утилиты командной строки и синтаксис их запуска;
- конфигурационные файлы и правила их написания;
- язык для составления управляющих скриптов.

Все зависит от того, какие средства для установки и настройки программы реализовали ее разработчики, какие именно инструменты они дают в руки системному администратору.

Методика и стиль изложения материала

Методика изложения материала в руководстве системного администратора сильно зависит от того, каким образом программой можно управлять.

Если большинство задач решается через административную консоль с графическим интерфейсом, то документ будет

больше похож на руководство пользователя или руководство администратора.

Если системному администратору придется активно составлять конфигурационные файлы и писать скрипты, документ будет ближе к руководству программиста или описанию языка программирования.

Типовая структура руководства системного программиста

Структура руководства системного программиста, приведена в ГОСТ 19.503-79:

- Общие сведения о программе.
- Структура программы.
- Настройка программы.
- Проверка программы.
- Дополнительные возможности.
- Сообщения системному программисту.

Приблизительная современная структура руководства системного администратора.

- Общие сведения о программе (комплексе).
- Архитектура и принципы функционирования.
- Системные требования.
- Установка программы (комплекса).
- Административная консоль и работа с ней.
- Файл конфигурации. Составление и правка.
- Обязательная начальная настройка программы (комплекса).
- Проверка правильности функционирования программы (комплекса).
- Мероприятия по текущему обслуживанию программы (комплекса).
- Оптимизация работы программы (комплекса).
- Аварийные ситуации и способы их устранения.
- Особенности.

Задания к работе

1. Подготовить документ (*.doc), содержащий структуру основных разделов руководства системного программиста стандартного форматирования: шрифт TimesNewRoman, 12 пт, поля, межстрочный интервал - стандартные, как в техническом задании, имя файла - <ФИО студента. Руководство пользователя>.

2. На основании технического задания на разработку (практическая работа МДК 03.01), заполнить разделы руководства пользователя "Общие сведения о программе", "Структура программы", "Настройка программы".

3. Сохранить документ с именем (Фамилия, инициалы студента. Наименование работы).

4. Прикрепить файл руководства в разделе Руководство системного программиста (практическая работа 2) учебного сервера.

ПРАКТИЧЕСКАЯ РАБОТА №3 Разработка руководства по сопровождению ПО

Цель работы

- приобретение навыков разработки руководства по сопровождению ПС

Краткие сведения из теории

Документация по сопровождению ПС (system documentation) описывает ПС с точки зрения его разработки.

Эта документация необходима, если ПС предполагает изучение того, как оно устроена (сконструирована), и модернизацию его программ.

То есть **тексты пишутся для разработчиков , подобных исполнителям** (*исполнители - это те, кто изначально создали ПС*).

В случае необходимости модернизации ПС к этой работе привлекается специальная команда разработчиков-сопроводителей. Этой команде придется иметь дело с такой же документацией, которая определяла деятельность команды первоначальных (основных) разработчиков ПС, - с той лишь разницей, что эта документация для команды разработчиков-сопроводителей будет, как правило, чужой (она создавалась другой командой).

Документацию по сопровождению ПС можно разбить на группы:

- документация, определяющая строение программ и структур данных ПС и технологию их разработки;

- документацию, помогающую вносить изменения в ПС.

Документация первой группы содержит итоговые документы каждого технологического этапа разработки ПС. Она включает следующие документы:

- **Внешнее описание ПС** (Requirements document).

• **Описание архитектуры ПС** (description of the system architecture), включая внешнюю спецификацию каждой ее программы.

- Для каждой программы ПС - **описание ее модульной структуры**, включая внешнюю спецификацию каждого включенного в нее модуля.

- Для каждого **модуля - его спецификация** и описание его строения (design description).

- **Тексты модулей** на выбранном языке программирования (program source code listings).

- **Документы установления достоверности ПС** (validation documents), описывающие, как устанавливалась достоверность каждой программы ПС и как информация об установлении достоверности связывалась с требованиями к ПС.

Документы установления достоверности ПС включают прежде всего документацию по тестированию (схема тестирования и описание комплекта тестов), но могут включать и результаты других видов проверки ПС, например, доказательства свойств программ.

Документация второй группы содержит:

- **Руководство по сопровождению ПС** (system maintenance guide), известные проблемы, связанные с ПС,

- **какие части системы являются аппаратно- и программно-зависимыми,**

- **возможности дальнейшего развития ПС.**

Задание к работе

1. Изучить РД 50-34.698-90, определить состав разделов руководства по сопровождению программы (http://prj-exp.ru/patterns/pattern_user_guide.php).

2. Подготовить документ (*.doc), содержащий структуру основных разделов руководства по попровождению стандартного форматирования: шрифт TimesNewRoman, 12 пт, поля, межстрочный интервал - стандартные, как в техническом задании, имя файла - <ФИО студента. Руководство по сопровождению>.

3. На основании результатов разработки ПС (в рамках курсового проектирования или производственной практики, заполнить разделы руководства.

4. Сохранить документ с именем (Фамилия, инициалы студента. Руководство по сопровождению).

5. Прикрепить файл руководства в разделе Руководство по сопровождению.(практическая работа 5) учебного сервера.

ПРАКТИЧЕСКАЯ РАБОТА №4 Разработка справочной системы программного продукта

Цель работы

- изучение нормативно правовой документации, регламентирующей разработку документации на программные средства.
- изучение структуры справочной системы ПС.
- закрепление навыков разработки справочной системы ПС.

Краткие сведения из теории

Общие сведения о структуре справочной системы

В справочную систему следует включать разделы, ознакомление с которыми предшествует работе с программой: системные требования, установка программы и т. п.

Когда пользователь запрашивает справку, например, нажатием на клавишу **F1**, на экран автоматически выводится раздел, связанный с активным элементом. Таковым может оказаться верхнее диалоговое окно, элемент интерфейса, на котором находится фокус ввода, элемент интерфейса, на который пользователь навел указатель мыши и т. д.

В разных операционных системах, графических оболочках и прикладных платформах способы запроса справки, вообще говоря, могут быть оригинальными. На практике в общем и целом пользователю везде предлагается примерно один и тот же набор возможностей и методов доступа к справке.

Возможно включение в справочные системы мультимедийный контент: звуковые и видео-файлы, а также Flash.

Полноценная **справочная система состоит**, по крайней мере, из двух частей: **общей и контекстной**. Более того, материал, который из-за большого объема невыгодно печатать на бумаге или неудобно искать в линейном документе, нередко включают только в справочную систему.

Общая часть справочной системы

В **общую часть** обычно включают материал всех имеющихся руководств: пользователя или оператора, администратора, системного администратора и других документов, разумеется при наличии таковых. Общей частью читатель пользуется как электронной книгой.

Таким образом получается электронная энциклопедия по программе или программно-аппаратному комплексу. Удобство ее еще и в том, что большинство форматов справки (и, следовательно, большинство просмотровых программ) позволяет снабдить справку удобными средствами навигации и поиска, в том числе, полнотекстового.

Контекстная часть справочной системы

В **контекстную часть** справочной системы включают:

- описание каждого режима и диалогового окна;
- подсказки по элементам главного окна, окон документов и диалоговых окон;
- подсказки по пунктам меню и кнопкам панелей инструментов;
- подсказки по употребляемым терминам.

Разделы контекстной части связывают с определенными элементами интерфейса, что требует координации, (как правило, несложной) действий автора справки и разработчиков программы.

Методика и стиль изложения информации в справочной системе

На практике справочная система часто представляет собой текст руководства пользователя, представленный в определенном электронном формате.

Как правило, справочная система — это гипертекст. Гипертекст не предполагает последовательного чтения. Читатель может выполнить поиск или нажать на клавишу **F1** и в результате получить доступ к произвольному (с точки зрения автора) разделу гипертекста.

Составляя тот или иной раздел, автор не может рассчитывать на то, что читатель уже усвоил материал предшествующих разделов или хотя бы осведомлен об их существовании. Поэтому следует стремиться к тому, чтобы каждый раздел представлял собой самостоятельную статью, при ограниченном объеме в понятную пользователю в отрыве от остального материала.

Таким образом, раздел должен быть чем-то вроде статьи в энциклопедии или обстоятельного ответа на заданный кем-то вопрос. Это может достигаться, в частности, более активным дублированием текста (определений, концепций), чем в линейном документе.

Если полное понимание раздела невозможно без ознакомления с некоторыми другими разделами справки, на них должны быть сделаны ссылки.

Глоссарий – перечень уникальных понятий, используемых в приложении и его интерфейсе. В качестве примера таковых могут выступать названия элементов меню, окон, режимов, текст командных кнопок и т.д.

Недопустимо наличие различных терминов для определения одного и того же понятия. Описание понятий должно отвечать промышленному руководству, соответствующему выбранной платформе (MS Microsoft).

Задания к работе

1. Составить схему меню программы, определить количество режимов работы, количество диалоговых окон.
2. В тетради для практических работ составить список основных терминов предметной области для определения глоссария приложения.
3. Определить разделы, которые должны быть включены в содержание и предметный указатель справочной системы.
4. С использованием текстового редактора подготовить 5-6 статей по каждому разделу справочной системы: изложение алгоритмов выполнения пользователем отдельных функций

(процедурная справка). Размер статьи - 1 -2 экранных страницы.

5. Добавить описания общей концепции приложения, ее функциональности в целом, а также отдельных функций, предоставляемых пользователю для выполнения (обзорная справка).

6. Сформировать файл тем справок в виде файла *.rtf .

7. Сформировать файл справочной системы *.hlp, создав файл Проекта справки и откомпилировав его средствами программы MS Help Workshop (HCRTF).

8. Используя те же средства создания справочной системы, сформировать файл содержания *.cnt

ПРАКТИЧЕСКАЯ РАБОТА №5 Разработка буклетов, рекламных листов по программному продукту

Цель работы

- приобретение навыков формулирования назначения программы и ее основных конкурентных преимуществ,
- приобретение навыков разработки рекламных буклетов для продвижения ПС.

Краткие сведения из теории

ЛИСТОВКИ - ЭТО.

- непериодическое издание на листе бумаги...
- лист бумаги с текстом и иллюстрациями...
- вид агитационной или рекламной полиграфии...
- полиграфическая продукция для раздачи...
- средство распространения информации

ЛИСТОВКИ - это вид полиграфической продукции. И как любая полиграфия, листовки подразделяются по различным параметрам.

Рекламные листовки - самый распространенный вид листовок. Данный вид листовок предназначен для привлечения новых клиентов к своей продукции или услугам. Или же рекламные листовки предлагают уже имеющимся клиентам новый продукт или новую услугу.

Рекламные листовки могут быть как дешевыми (бюджетный вариант листовки), напечатанными на простой писчей или офсетной бумаге в 1-2 цвета, так и более качественными, напечатанными на хорошей мелованной бумаге офсетным или трафаретным способом.

На рекламных листовках могут быть применены различные способы привлечения внимания.

Информационные листовки содержат развернутую информацию о конкретном товаре или конкретной услуге. Как правило, в таких листовках описывается или один вид деятельности предприятия, или один товар / группа товаров.

Информационные листовки оформляются в корпоративном стиле компании-продавца или в корпоративном стиле предлагаемого товара/услуги.

Задание к работе

1. Изучить предложенные преподавателем рекламные буклеты: структурирование текстовой и графической информации, цветовые характеристики.
2. Составить макет буклета в тетради для практических работ: размер бумаги, количество сложений, расположение блоков информации.
3. Используя MS Publisher выбрать наиболее подходящий шаблон, подготовить рекламный буклет на программное изделие.
4. Сохранить файл буклета с именем (Фамилия, инициалы студента. Наименование работы).
5. Прикрепить файл буклета в разделе Рекламный буклет (практическая работа 5) учебного сервера.

ПРАКТИЧЕСКАЯ РАБОТА №6 Тестирование по методу «черного ящика»

Цель работы

- приобретение навыков разработки тестовых заданий по методу "черного ящика".

Краткие сведения из теории

Одним из способов проверки программ является стратегия тестирования, называемая стратегией "черного ящика" или тестированием с управлением по данным. В этом случае программа рассматривается как "черный ящик" и такое тестирование имеет целью выяснения обстоятельств, в которых поведение программы не соответствует спецификации.

Стратегия "черного ящика" включает в себя следующие методы формирования тестовых наборов:

1. *эквивалентное разбиение;*
2. *анализ граничных значений;*
3. *анализ причинно-следственных связей;*
4. *предположение об ошибке.*

1. Эквивалентное разбиение

Основу метода составляют положения:

1. Исходные данные программы необходимо разбить на конечное число классов эквивалентности.
2. Каждый тест должен включать по возможности максимальное количество различных входных условий, что позволяет минимизировать общее число необходимых тестов.

Первое положение используется для разработки набора "интересных" условий, которые должны быть протестированы, а второе - для разработки минимального набора тестов.

Разработка тестов методом эквивалентного разбиения осуществляется в два этапа:

- выделение классов эквивалентности;
- построение тестов.

Выделение классов эквивалентности

Классы эквивалентности выделяются путем выбора каждого входного условия (обычно это предложение или фраза из спецификации) и разбиением его на две или более групп (таблица 1).

Таблица 1 Классы эквивалентности

Входное условие	Правильные классы эквивалентности	Неправильные классы эквивалентности
Если входные условия описывают <i>область</i> значений (например «целое данное может принимать значения от 1 до 999»)	выделяют один правильный класс $1 \leq x \leq 999$	два неправильных $x < 1$ и $x > 999$.
Если входное условие описывает <i>число</i> значений (например, «в автомобиле могут ехать от одного до шести человек»)	определяется один правильный класс эквивалентности	и два неправильных (ни одного и более шести человек).
Если входное условие описывает множество входных значений и есть основания полагать, что каждое значение программист трактует особо (например, «известные способы передвижения на АВТОБУСЕ, ГРУЗОВИКЕ, ТАКСИ, МОТОЦИКЛЕ или ПЕШКОМ»),	определяется один правильный класс эквивалентности для каждого значения	один неправильный класс (например «на ПРИЦЕПЕ»)
Если входное условие описывает ситуацию «должно быть» (например, «первым символом	один правильный класс эквивалентно	один неправильный (первый символ - не

идентификатора должна быть буква»),	сти (первый символ - буква)	буква)
--	-----------------------------------	--------

Если есть основание считать, что различные элементы класса эквивалентности трактуются программой неодинаково, то данный класс разбивается на меньшие классы эквивалентности.

Построение тестов

Этот шаг заключается в использовании классов эквивалентности для построения тестов. Этот процесс включает в себя:

- Назначение каждому классу эквивалентности уникального номера.
- Проектирование новых тестов, каждый из которых покрывает как можно большее число непокрытых классов эквивалентности, до тех пор, пока все правильные классы не будут покрыты (только не общими) тестами.
- Запись тестов, каждый из которых покрывает один и только один из непокрытых неправильных классов эквивалентности, до тех пор, пока все неправильные классы не будут покрыты тестами.

Разработка индивидуальных тестов для неправильных классов эквивалентности обусловлено тем, что определенные проверки с ошибочными входами скрывают или заменяют другие проверки с ошибочными входами. Недостатком метода эквивалентных разбиения в том, что он не исследует комбинации входных условий.

2. Анализ граничных значений

Граничные условия - это ситуации, возникающие на, выше или ниже границ входных классов эквивалентности.

Применение метода анализа граничных условий требует определенной степени творчества и специализации в рассматриваемой проблеме. Тем не менее, существует несколько общих правил этого метода:

1. Построить тесты для границ области и тесты с неправильными входными данными для ситуаций незначительного выхода за границы области, если входное условие описывает область значений (например, для области входных значений от -1.0 до $+1.0$ необходимо написать тесты для ситуаций -1.0 , $+1.0$, -1.001 и $+1.001$).
2. Построить тесты для минимального и максимального значений условий и тесты, большие и меньшие этих двух значений, если входное условие удовлетворяет дискретному ряду значений. Например, если входной файл может содержать от 1 до 255 записей, то проверить 0, 1, 255 и 256 записей.
3. Использовать правило 1 для каждого выходного условия. Причем, важно проверить границы пространства результатов, поскольку не всегда границы входных областей представляют такой же набор условий, как и границы выходных областей. Не всегда также можно получить результат вне выходной области, но, тем не менее, стоит рассмотреть эту возможность.
4. Использовать правило 2 для каждого выходного условия.
5. Если вход или выход программы есть упорядоченное множество (например, последовательный файл, линейный список, таблица), то сосредоточить внимание на первом и последнем элементах этого множества.
6. Попробовать свои силы в поиске других граничных условий.

3. Анализ причинно-следственных связей

Метод анализа причинно-следственных связей помогает системно выбирать высокорезультативные тесты. Он дает полезный побочный эффект, позволяя обнаруживать неполноту и неоднозначность исходных спецификаций.

Для использования метода необходимо понимание булевой логики (логических операторов - и, или, не). Построение тестов осуществляется в несколько этапов.

1. Спецификация разбивается на «рабочие» участки, так как таблицы причинно-следственных связей становятся громоздкими при применении метода к большим спецификациям.
2. В спецификации определяются множество причин и множество следствий. ^ Причина есть отдельное входное условие или класс эквивалентности входных условий. Следствие есть выходное условие или преобразование системы. Каждой причине и следствию приписывается отдельный номер.
3. На основе анализа семантического (смыслового) содержания спецификации строится таблица истинности, в которой последовательно перебираются все возможные комбинации причин и определяются следствия каждой комбинации причин.
4. Каждая строка таблицы истинности преобразуется в тест.

4. Предположение об ошибке

Часто программист с большим опытом выискивает ошибки "без всяких методов". При этом он подсознательно использует метод "предположение об ошибке".

Процедура метода предположения об ошибке в значительной степени основана на интуиции.

Основная идея метода состоит в том, чтобы перечислить в некотором списке возможные ошибки или ситуации, в которых они могут появиться, а затем на основе этого списка составить тесты.

Другими словами, требуется перечислить те специальные случаи, которые могут быть не учтены при проектировании.

Общая стратегия тестирования

Все методологии проектирования тестов могут быть объединены в общую стратегию. Это оправдано тем, что каждый метод обеспечивает создание определенного набора

тестов, но ни один из них сам по себе не может дать полный набор тестов. Приемлемая стратегия состоит в следующем:

1. Если спецификация состоит из комбинации входных условий, то начать рекомендуется с применения метода функциональных диаграмм.
2. В любом случае необходимо использовать анализ граничных значений.
3. Определить правильные и неправильные классы эквивалентности для входных и выходных данных и дополнить, если это необходимо, тесты, построенные на предыдущих шагах.
4. Для получения дополнительных тестов рекомендуется использовать метод предположения об ошибке.

Задание к работе

Часть 1. Разработка тестовых наборов с использованием метода эквивалентного разбиения

1. Ознакомиться с теоретическими сведениями по стратегиям тестирования.
2. В соответствии с вариантом, подготовить исчерпывающие тесты с использованием метода эквивалентного разбиения.
3. Выполнить тестирование предложенной программы.
4. Сделать вывод о результативности тестирования с использованием стратегии "черного ящика".
5. При наличии ошибки в программе, сформулировать гипотезу для локализации ошибки.

Часть 2. Разработка тестовых наборов с использованием метода анализа граничных условий

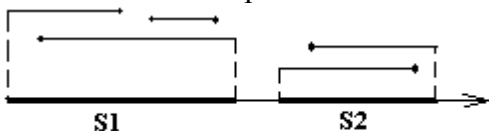
1. Ознакомиться с теоретическими сведениями по стратегиям тестирования.
2. В соответствии с вариантом, подготовить исчерпывающие тесты с использованием метода анализа граничных условий.
3. Выполнить тестирование предложенной программы.
4. Сделать вывод о результативности тестирования с использованием стратегии "черного ящика".
5. При наличии ошибки в программе, сформулировать гипотезу для локализации ошибки.

Часть 3. Разработка тестовых наборов с использованием анализа причинно-следственных связей

1. Ознакомиться с теоретическими сведениями по стратегиям тестирования.
2. В соответствии с вариантом, подготовить исчерпывающие тесты с использованием метода анализа причинно-следственных связей.
3. Выполнить тестирование предложенной программы.
4. Сделать вывод о результативности тестирования с использованием стратегии "черного ящика".
5. При наличии ошибки в программе, сформулировать гипотезу для локализации ошибки.

Варианты заданий к практической работе 6

1. Выполнить тестирование программы решения уравнения $ax^2 + bx - c = 0$, где a , b , c - любые вещественные числа.
2. Выполнить тестирование программы определения суммарной длины тени, которую отбрасывают на ось OX отрезки, параллельные этой оси и заданные координатами x начала и конца отрезка:



$$S = S1 + S2$$

3. Выполнить тестирование программы определения вида треугольника, заданного длинами его сторон: равносторонний, равнобедренный, прямоугольный, разносторонний.
4. Выполнить тестирование программы определения вида четырехугольника, заданного координатами вершин на плоскости: квадрат, прямоугольник, параллелограмм, ромб, равнобедренная трапеция, прямоугольная трапеция, трапеция общего вида, четырехугольник общего вида.

ПРАКТИЧЕСКАЯ РАБОТА №7 Тестирование по методу «белого ящика»

Цель работы

- приобретение навыков разработки тестовых заданий по методу "белого ящика".

Краткие сведения из теории

Тестирование по принципу белого ящика характеризуется степенью, какой тесты выполняют или покрывают логику (исходный текст программы).

1. Метод покрытия операторов

Целью этого метода тестирования является выполнение каждого оператора программы хотя бы один раз.

2. Метод покрытия решений (покрытия переходов)

Более сильный метод тестирования известен как покрытие решений (покрытие переходов). Согласно данному методу каждое направление перехода должно быть реализовано по крайней мере один раз.

Покрытие решений обычно удовлетворяет критерию покрытия операторов. Поскольку каждый оператор лежит на некотором пути, исходящем либо из оператора перехода, либо из точки входа программы, при выполнении каждого направления перехода каждый оператор должен быть выполнен.

3. Метод покрытия условий

Лучшие результаты по сравнению с предыдущими может дать метод покрытия условий. В этом случае записывается число тестов, достаточное для того, чтобы все возможные результаты каждого условия в решении выполнялись по крайней мере один раз.

Критерий покрытия решений/условий требует такого достаточного набора тестов, чтобы все возможные результаты каждого условия в решении выполнялись по крайней мере один раз, все результаты каждого решения выполнялись по крайней мере один раз и, кроме того, каждой точке входа передавалось управление по крайней мере один раз.

4. Метод комбинаторного покрытия условий

Критерием, который решает эти и некоторые другие проблемы, является комбинаторное покрытие условий. Он требует создания такого числа тестов, чтобы все возможные комбинации результатов условия в каждом решении выполнялись по крайней мере один раз.

Набор тестов, удовлетворяющих критерию комбинаторного покрытия условий, удовлетворяет также и критериям покрытия решений, покрытия условий и покрытия решений/условий.

Задание к работе

Часть 1. Разработка тестовых наборов с использованием метода покрытия операторов

1. В соответствии с вариантом составить тесты для проверки программы (приложение 1), используя метод покрытия операторов.

Часть 2. Разработка тестовых наборов с использованием метода покрытия решений

1. В соответствии с вариантом составить тесты для проверки программы (приложение 1), используя метод покрытия решений.

Часть 3. Разработка тестовых наборов с использованием метода комбинаторного покрытия условий

1. В соответствии с вариантом составить тесты для проверки программы (приложение 1), используя метод комбинаторного покрытия условий.

Приложение 1

1. Условие задачи

Написать программу, в которой с клавиатуры вводятся три целых положительных числа, которые являются сторонами треугольника. На экран выводится сообщение, является ли треугольник неравносторонним, равнобедренным, равносторонним. (Треугольник должен быть правильным, т.е. сумма длин любых двух сторон должна быть больше длины третьей стороны.)

Программа должна быть составлена с применением пользовательских процедур и функций.

2. Спецификация программы:

Название задачи: треугольник

Название программы: triangle

Система программирования: Turbo Pascal 7.0

Системные требования: IBM PC/AT 286 и выше

Входные данные:

Три целых числа – стороны треугольника

Выходные данные:

Сообщение о виде треугольника.

3. Постановка задачи:

Нужно составить программу, которая бы сравнивала каждую пару сторон и, в зависимости от результата, выдавала бы на экран сообщение о типе треугольника.

4. Спецификация переменных:

Имя переменной в программе	Назначение переменной в программе	Тип переменной	Диапазон типа
a	Массив, содержащий длины сторон	Array of Integer	32768..32767 каждый
i	Вспомогательная переменная, счетчик	Integer	32768..32767

5. Текст программы:

```
program triangle;
  Uses Crt;
  var a : array [1..3] of integer;
  i : integer;
Procedure valid;
begin
  if (a[1]+a[2]>a[3])
    and (a[1]+a[3]>a[2])
    and (a[2]+a[3]>a[1])
  then
    begin
      if (a[1]=a[2]) and (a[1]=a[3])
      then WriteLn ('Треугольник равносторонний ')
      else
        if (a[1]=a[2]) or (a[1]=a[3]) or (a[2]=a[3])
        then WriteLn ('Треугольник равнобедренный')
        else WriteLn ('Треугольник неравносторонний');
      end
    else WriteLn ('Треугольник неправильный')
    end;
begin
  ClrScr;
  WriteLn ('Ввод длин сторон треугольника:');
  for i := 1 to 3 do
    begin
      WriteLn ('Введите длину ',i,' стороны треугольника :');
      Read (a[i]);
    end;
  Valid;
  ReadKey;
end.
```

Приложение 2

1. Условие задачи

Написать программу, в которой находятся корни квадратного уравнения вида $ax^2 + bx + c = 0$. Случай комплексных корней (дискриминант меньше нуля) не рассматривать. Коэффициенты a , b и c вводятся с клавиатуры. Значения корней выводятся на экран.

Программа должна быть составлена с применением пользовательских процедур и функций.

2. Спецификация программы:

Название задачи: квадратное уравнение

Название программы: `quadrur`

Система программирования: Turbo Pascal 7.0

Системные требования: IBM PC/AT 286 и выше

Входные данные:

Три действительных числа – коэффициенты.

Выходные данные:

Вычисленные корни уравнения.

3. Постановка задачи:

Нужно составить программу, которая бы вычисляла решения квадратного уравнения, если уравнение действительно является квадратными и если имеются действительные корни уравнения.

4. Спецификация переменных:

Имя переменной в программе	Назначение переменной в программе	Тип переменной	Диапазон типа
a	Первый коэффициент уравнения	Real	2.9e-39..1.7e38
b	Второй коэффициент уравнения	Real	2.9e-39..1.7e38
c	Третий	Real	2.9e-39..1.7e38

	коэффициент уравнения		
--	--------------------------	--	--

5. Текст программы:

```

program quadrur;
Uses Crt;
var
a, b, c: real;
procedure Entering;
begin
Write ('Введите значение a: ');
ReadLn (a);
Write ('Введите значение b: ');
ReadLn (b);
Write ('Введите значение c: ');
ReadLn (c);
end; procedure Entering
function Dis:real;
begin
dis := b*b-4*a*c;
end; function Dis
function Valid:boolean;
begin
Valid := true;
if a = 0 then
begin
Writeln ('Уравнение линейное');
Valid := false;
end else if Dis < 0 then
begin
Writeln ('Уравнение не имеет действительных корней');
Valid := false;
end;
end; function Valid
function Results:real;
begin
WriteLn ('x1 = ',(-b-SQRT(dis))/(2*a):3:6);
WriteLn ('x2 = ',(-b+SQRT(dis))/(2*a):3:6);
end; function Results

```

```
begin
ClrScr;
Write ('Решение квадратного уравнения');
WriteLn (' вида  $ax^2+bx+c=0$ ');
WriteLn;
Entering;
if Valid = true then
Results;
ReadKey;
end.
```

8. Набор тестов (тестирование): Метод покрытия операторов

Очевидно, что выполнить все операторы можно, подобрав тестовые значения, при которых будут выполнены все ветви функции Valid, где происходит сравнение с нулем переменной a и функции Dis. Остальные части программы имеют линейную структуру и при выполнении программы будут выполнены в любом случае. Таким образом, подобрав значения переменных, при которых истинны пары условий $a = 0$, $Dis < 0$ $a = 0$, $Dis > 0$

$a \neq 0$, $Dis < 0$

$a \neq 0$, $Dis > 0$, выполним каждый оператор программы.

Первую пару можно не рассматривать, т.к. при $a = 0$ нет смысла переходить к проверке значения дискриминанта, т. е. необходимо провести три теста:

Тест 1 Определяет, будет ли выполнен оператор вывода

на экран с сообщения «Уравнение линейное», когда $a = 0$

Входные данные: $a = 0$, $b = 2$, $c = 3$

Тест 2

Определяет, будет ли выполнен оператор вывода на экран с сообщения

«Уравнение не имеет действительных корней», когда $a \neq 0$, $dis < 0$

Входные данные: $a = 3$, $b = 2$, $c = 4$

Тест 3

Определяет, будет ли выполнена функция Results, когда $a \neq 0$, $dis > 0$

Входные данные: $a = 2$, $b = 5$, $c = 3$

Метод покрытия решений

Для того, чтобы каждое решение приняло значение ИСТИНА либо ЛОЖЬ хотя

бы один раз, достаточно трех тестов, когда $a = 0$; $a \neq 0$, $Dis < 0$; $a \neq 0$, $Dis > 0$:

Тест 4

Определяет, будет ли выполнен оператор вывода на экран с сообщения

«Уравнение линейное», когда $a = 0$

Входные данные: $a = 0, b = 3, c = -2$

Тест 5

Определяет, будет ли выполнен оператор вывода на экран с о общения

«Уравнение не имеет действительных корней», когда $a \neq 0, \text{dis} < 0$

Входные данные: $a = 3, b = 2, c = 1$

Тест 6 Определяет, будет ли выполнена функция Results , когда $a \neq 0, \text{dis} > 0$

Входные данные: $a = 2, b = 4, c = 1$

9. Ошибки:

Тест 1: выявляет ошибку, когда при введенном $a = 0$ вычисления производятся далее.

Тест 2: выявляет ошибку, когда при значении $\text{Dis} < 0$ вычисления производятся далее.

Тест 3: выявляет ошибку, когда при значениях $a \neq 0, \text{Dis} > 0$ вычисления далее не производятся.

Тест 4: выявляет ошибку, когда при введенном $a = 0$ вычисления производятся далее.

Тест 5: выявляет ошибку, когда при значении $\text{Dis} < 0$ вычисления производятся далее.

Тест 6: выявляет ошибку, когда при значениях $a \neq 0, \text{Dis} > 0$ функция Valid не возвращает значение true и вычисления далее не производятся.