

**Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«Томский государственный университет систем управления и радиоэлектроники»
(ТУСУР)**

УТВЕРЖДАЮ

Проректор по учебной работе
_____ Л.А. Боков
«____» _____ 20 г.

Вводится в действие с" ____ " _____ 20 г.

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ**

по дисциплине

«Языки и технологии программирования»

Составлена кафедрой

«Управление инновациями»

Для студентов, обучающихся
по направлению подготовки 220600.68 «Инноватика», магистерская программ
«Мультимедийные многопроцессорные системы на кристалле»

Форма обучения

очная

Томск 2011

Содержание

Введение	3
Практическое занятие № 1. «Первоначальное знакомство со структурой процесса разработки в ЗАО “Элекард-Девайsez”. Анализ структуры одного из мультимедиа компонентов. Сборка мультимедиа компонента»	4
Практическое занятие № 2 «Проектирование мультимедиа компонента (DirectShow-фильтра)»	5
Практическое занятие № 3. «Изучение инструментальных средств: SVN, MS Visual Studio, WinDbg»	5
Практическое занятие № 4. «Реализация структуры спроектированного мультимедиа компонента»	6
Практическое занятие № 5. «Поиск и устранение проблемных мест в реализованном мультимедиа компоненте»	6
Практическое занятие № 6. «Подготовка тестов и тестирование мультимедиа компонента»	7
Библиографический список	9

Введение

Изучение дисциплины «Языки и технологии программирования» (уровень дисциплины региональный (вузовский)) имеет основополагающее значение в специальной подготовке студентов по направлению «Инноватика».

Цель данного пособия состоит в выработке навыков в области программирования на языках высокого уровня, организации и ведения процесса проектирования, разработки, тестирования и сопровождения программного обеспечения.

Для полноценного понимания и усвоения материала необходимо предварительно изучить дисциплины "Программирование".

Для углубленного изучения и освоения материала целесообразно применение различных форм самопроверки знаний студентов: тесты, задачи, упражнения. Они могут быть использованы при проведении практических занятий в университете, выполнении курсовых, контрольных и аудиторных работ, а также при самостоятельном изучении данных дисциплин.

Одним из наиболее интенсивных способов изучения дисциплины является самостоятельное решение практических задач. При этом вырабатываются навыки оценки сложности поставленной задачи, сроков ее реализации и требующихся программно-аппаратных средств.

Предлагаемые задания позволят глубже освоить теоретические и практические вопросы написания качественного легко сопровождаемого программного кода, понять принципы организации процесса разработки и научиться применять методики проектирования.

Практическое занятие № 1. «Первоначальное знакомство со структурой процесса разработки в ЗАО “Элекард-Девайsez”. Анализ структуры одного из мультимедиа компонентов. Сборка мультимедиа компонента»

Цель занятия:

Получить представление о структуре процесса разработки в ЗАО “ЭЛЕКАРД-Девайsez”. Познакомиться со структурой DirectShow-фильтра. Научиться работать с большими объемами исходных текстов. Получить навыки сборки программного обеспечения из исходных текстов.

Задание

Сформировать методологию разработки выданного проекта и реализовать проект.

Порядок выполнения.

1. Получить на группу задание по реализации небольшого учебного проекта.

2. Согласно сформированной группе выбрать наиболее подходящую на Ваш взгляд модель процесса разработки и поделить роли между участниками. Обоснование выбора включить в отчет.

3. Используя в качестве среды разработки Visual C++, в качестве системы контроля версий Tortoise GIT, соблюдая распределение ролей, выполнить в группе выданное задание. Процесс выполнения и результаты включить в отчет.

4. Получить доступ к репозиторию Subversion (будет выдан в виде отдельного файла). Скачать с него последнюю ревизию нужного проекта при помощи TortoiseSVN и TortoiseGIT. Достоинства и недостатки двух способов работы с Subversion описать в отчете.

5. Проанализировать исходный код, структуру проекта. Скомпилировать, запустить проект (проект содержит легко обнаруживаемые ошибки компиляции и времени выполнения). Отчет о найденных ошибках включить в отчет.

6. Еще раз проанализировать исходники, произвести рефакторинг проекта с учетом рекомендаций, описанных в методическом пособии. Проводимые изменения вместе с их обоснованием включить в отчет.

7. Создать MAKEFILE для автоматизированной сборки проекта. Описание ключевых участков файла включить в отчет.

Результатом практического занятия должно явиться практическое освоение процесса разработки в ЗАО «Элекард Девайсез», существующих методологий разработки, инструментальных средств разработки, формирование командного мышления.

Практическое занятие № 2 «Проектирование мультимедиа компонента (DirectShow-фильтра)»

Цель занятия:

Познакомиться с языком UML, структурой DirectShow-фильтра.
Изучить основные принципы проектирования программного обеспечения

Задание

Подготовить выданный проект к разработке.

Порядок выполнения.

1. Познакомиться с языком UML.
2. Выбрать тип DirectShow фильтра для проектирования.
3. С помощью программного пакета MS Visio создать UML структуру.
4. Подготовиться к ответам на теоретическую часть лабораторной работы.

Результатом практического занятия должно явиться практическое освоение основ работы мультимедиа компонентов (DirectShow фильтра), получение навыков по подготовке проекта к разработке.

Практическое занятие № 3. «Изучение инструментальных средств: SVN, MS Visual Studio, WinDbg»

Цель занятия:

Изучить возможности и получить навыки работы с инструментальными средствами: SVN, MS Visual Studio, WinDbg.

Задание

Собрать наиболее позднюю версию проекта из системы контроля версий, с помощью средств отладки обнаружить привнесенную ошибку (утечка памяти, исключение и т.п.).

Порядок выполнения.

1. Получить исходные тексты проекта из репозитория SVN.
2. Загрузить проект в среду MS Visual Studio.
3. Собрать проект.
4. Настроить и запустить встроенный отладчик MS Visual Studio.
5. Запустить WinDbg, подключиться к запущенному процессу с исследуемым модулем.

Результатом практического занятия должно явиться приобретение навыков работы с системой контроля версий, инструментальными средствами сборки и отладки программных продуктов.

Практическое занятие № 4. «Реализация структуры спроектированного мультимедиа компонента»

Цель занятия:

Получить навыки разработки мультимедиа компонентов.

Задание

Реализовать спроектированной на практическом занятии №2 проект.

Порядок выполнения.

1. Запустить MS Visual Studio и создать новый проект типа C++ DLL.
2. Реализовать структуру ранее спроектированного мультимедиа компонента (практическое занятие №2).
3. Собрать реализованный мультимедиа компонент.

Результатом практического занятия должно явиться получение навыков разработки мультимедиа компонентов (DirectShow фильтров), практическое освоение изученных концепция проектирования.

Практическое занятие № 5. «Поиск и устранение проблемных мест в реализованном мультимедиа компоненте»

Цель занятия:

Получить навыки рефакторинга программного обеспечения.

Задание

Устранить проблемные, недоработанные места в реализованном на практическом занятии №4 проекте.

Порядок выполнения.

1. Открыть проект, реализованного в лабораторной работе №4, мультимедиа компонент.
2. Разработать список элементов для поиска проблемных мест в реализации на основании материала описанного в разделе 9.6 (см. “Учебно-методические материалы”).
3. Устранить проблемные места реализованном мультимедиа компоненте.

Результатом практического занятия должно явиться получение навыков рефакторинга, поиска ошибок и устранения проблем.

Практическое занятие № 6. «Подготовка тестов и тестирование мультимедиа компонента»

Цель занятия:

Познакомиться с основными видами и способами тестирования.
Получить практические навыки по проведению тестов.

Задание

Написать тест для одного из модулей выданного проекта.

Порядок выполнения.

1. Изучить исходный код выданного проекта. Выявить функциональное назначение проекта, принцип его работы. Определить главные вычислительные модули (основные вычислительные функции). Провести статистическое тестирование. Описание свойств проекта, обоснование выбора модулей, результаты тестирования включить в отчет.
2. Написать тест для выбранного модуля. Охарактеризовать впечатления от применения методологии TDD: достоинства/недостатки в данном конкретном случае. Сделать выводы о корректности работы модуля. Результаты включить в отчет.

3. Какие типы тестирования по способу выбора входных значений целесообразно применить для выданного проекта. Обоснованное мнение включить в отчет.

4. Выберите и обоснуйте наиболее полезный на Ваш взгляд метод интеграционного тестирования (восходящее тестирование, моноконтное тестирование, нисходящее тестирование). Проведите выбранный метод тестирования.

5. Охарактеризуйте требования к выданному проекту с точки зрения верного функционирования, производительности, стрессоустойчивости, корректности работы на поддерживаемом аппаратном обеспечении, безопасности, надежности, удобства. Определите наиболее важное требование и проведите для него системный тест. Описание и выводы включить в отчет.

Результатом практического занятия должно явиться получение практических навыков по написанию тестов для программных проектов.

Библиографический список

1. Гради Буч, Роберт Максимчук, Майкл Энгл, Бобби Янг, Джим Коннален, Келли Хьюстон. Объектно-ориентированный анализ и проектирование с примерами приложений. – М.: «Вильямс», 2008. – 720 с.: ил.
2. Фаулер М. UML. Основы, 3-е издание. – Пер. с англ. – СПб: Символ–Плюс, 2004. – 192 с.,ил.
3. Иан Грэхем. Объектно-ориентированные методы. Принципы и практика. — 3-е изд. — М.: «Вильямс», 2004. — 880 с.
4. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб: Питер, 2001. — 368 с.: ил.
5. Коплиен Дж. Мультипарадигменное программирование для C++. Библиотека программиста. – СПб.: Питер, 2005. — 235 с: ил.
6. Коплиен Дж. Программирование на C++. Классика CS. Библиотека программиста. – СПб.: Питер, 2005. — 479 с: ил.