

**Министерство образования и науки Российской Федерации
Государственное образовательное учреждение высшего профессионального образования
«Томский государственный университет систем управления и радиоэлектроники»**

УТВЕРЖДАЮ

Зав.кафедрой ЭС

_____ Н.Е.Родионов
" ____ " _____ 2012 г.

Вводится в действие с " ____ " _____ 20 г.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ РАБОТ**

по дисциплине

Прикладное программирование микропроцессорных систем

Составлена кафедрой

Электронных систем

Для студентов, обучающихся
по направлению подготовки 222000.68 «Инноватика»

Форма обучения

очная

Составитель преподаватель кафедры
Электронных систем

Нестеренко П.Г.

"10" августа 2012 г

Томск 2012 г.

Введение

Программирование микропроцессорных систем имеет некоторые особенности обусловленные более плотным взаимодействием программного обеспечения (ПО) с аппаратной частью комплекса. Это взаимодействие накладывает как чисто программные особенности, так и организационные подходы при создании программного обеспечения.

Полученные знания и навыки могут быть использованы при разработке ПО микропроцессорных систем, в системах и комплексах автоматизированного управления.

Общие требования

Практические работы выполняются студентами очной формы обучения индивидуально под контролем со стороны преподавателя. Все консультации осуществляются преподавателем. Число студентов, одновременно присутствующих на занятии не должно превышать 12 человек. Если в списочном составе группы студентов больше 12, то группа должна быть разделена на подгруппы численностью от 6 до 12 человек в каждой.

Для выполнения практических заданий целесообразно в учебном расписании выделять 4 академических часа подряд, без больших перерывов. Расписание также должно предусматривать раздельное проведение занятий у подгрупп, если группа была разделена.

Перед началом занятий студенты должны изучить инструкцию по охране труда. Преподаватель должен убедиться в знании инструкции, задавая студенту вопросы по ее содержанию, после чего сделать соответствующую запись в журнале охраны труда.

Во время проведения практических занятий в аудитории (лаборатории) студентам запрещается передавать друг другу файлы и другие материалы, являющиеся результатом выполнения заданий.

Студент имеет право:

- Выходить из аудитории (лаборатории) не спрашивая разрешения у преподавателя.
- Самостоятельно распределять аудиторное время, определяя необходимость перерыва или непрерывной работы.
- Просить консультации у преподавателя, если он в текущий момент не распределяет задания, не принимает выполненные работы и не консультирует другого студента.

Преподаватель, давая консультацию студенту, указывает раздел технической документации или методической литературы, в которой имеется ответ на вопрос студента. Если необходимые сведения в документации и литературе отсутствуют, то преподаватель должен дать устные пояснения или продемонстрировать практические действия, приводящие к требуемому результату, с последующей отменой для повторения студентом.

Самостоятельная работа студентов над практическими занятиями осуществляется в той же аудитории (лаборатории), где проводятся практические занятия. Преподаватель должен согласовать со студентами расписание самостоятельной работы - не менее 2 астрономических часов в неделю. В указанное время по учебному расписанию студентов и в аудитории (лаборатории) не должны проводиться другие занятия. Преподаватель должен обеспечить доступ студентов в аудиторию (лабораторию) в указанные часы. Необходимость самостоятельной работы определяет студент.

Консультации, выдача практических заданий и прием результатов выполнения осуществляется только во время аудиторных занятий. Задания выполняются последовательно. Правильное выполнение некоторых заданий возможно только, если студент корректно выполнил предыдущие задания. Поэтому приступать к следующему заданию студент может, только сдав преподавателю результат выполнения предыдущего.

Техническое обеспечение практических работ

Для выполнения практического задания студенту предоставляется индивидуальное рабочее место, в состав которого входят:

- персональный компьютер с операционной системой Windows XP;
- текстовый редактор Word 2003 или подобный.

Размещение и освещенность рабочих мест в учебной аудитории (лаборатории) должно удовлетворять действующим требованиям СанПиН.

Прием результатов выполнения практических заданий

Результаты выполнения практических заданий демонстрируются преподавателю. Во время приема выполненной работы преподаватель вправе:

- Требовать у студента демонстрации выполненного задания в виде таблиц, мнемосхем, рисунком или диаграмм.
- Самостоятельно производить манипуляции с программным обеспечением, не изменяя его конфигурацию.
- Требовать у студента пояснений, относящихся к способам реализации функций задания.

Задание считается выполненным и принимается преподавателем только в том случае, если реализован весь функционал, предусмотренный заданием. Если какие то функции, предусмотренные заданием, не работают, или работают неверно, то результат выполнения подлежит доработке. Студент должен работать внимательно и аккуратно. Подлежат исправлению замеченные преподавателем недочеты:

- грамматические ошибки в надписях, наименованиях элементов;
- небольшие неточности в описаниях, структурах, схемах;

Результаты выполнения заданий сохраняются преподавателем в электронном виде и хранятся в течение двух лет.

До конца семестра студент должен сдать результаты выполнения всех практических заданий, предусмотренным настоящими указаниями. В противном случае студенты к сдаче экзамена (зачета) не допускаются.

Задания для практических занятий

1. Основные этапы жизненного цикла разработки. Нормативное время выполнения – 1 часа. Задание в приложении А.
2. Основные требования ГОСТ на процесс разработки программного обеспечения. Нормативное время выполнения – 1 часа. Задание в приложении Б.
3. Формирование технических требований, технического задания, описания программного продукта, методики тестирования. Нормативное время выполнения – 2 часа. Задание в приложении В.
4. Граф состояний – один из методов логической организации программного обеспечения. Нормативное время выполнения – 2 часа. Задание в приложении Г.
5. Системы контроля версий. Нормативное время выполнения – 2 часа. Задание в приложении Д.
6. Системы отслеживания ошибок (bug-track). Нормативное время выполнения – 2 часа. Задание в приложении Е.
7. Методы проектирования программных средств. Нормативное время выполнения – 2 часа. Задание в приложении Ж.

Библиографический список

1. В.А. Благодатских, В.А. Волнин, К.Ф. Поскалов Стандартизация разработки программных средств.
2. Шальто А.А. Switch-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998. 628 с. УДК 681.3.06:62–507.

Приложение А

Основные этапы жизненного цикла разработки.

Цель работы: Познакомить студентов с основными жизненными циклами программного обеспечения. Получения навыков планирования процессов создания ПО.

Задание считается выполненным, если студент:

1. Графически продемонстрировал основные этапы жизненного цикла ПО.
2. Смог кратко охарактеризовать особенности каждого из этапов жизненного цикла.
3. Привел вспомогательные этапы жизненного цикла и охарактеризовал их.

Вопросы для самоконтроля:

1. Понятие жизненного цикла?
2. Смысл каскадной и спиральной модели жизненного цикла?
3. Положительные и отрицательные стороны каждой из моделей?

Приложение Б

Основные требования ГОСТ на процесс разработки программного обеспечения.

Цель работы: Познакомить студентов с основными стандартами на разработку ПО.

Задание считается выполненным, если студент:

1. Назвал основные организации занимающиеся стандартизацией в области разработки программных средств.
2. Привел основные ГОСТ при разработке программных средств.
3. Графически продемонстрировал модель жизненного цикла ПО в соответствии с требованиями стандарта.

Вопросы для самоконтроля:

1. Определение понятия “стандарт”?
2. Основные виды нормативных документов?
3. Различия стандартов “де-факто” и “де-юре”?
4. Назначение стандартов комплекса ГОСТ 34.xxx ?
5. Назначение стандартов комплекса ГОСТ 19.xxx ?

Приложение В

Формирование технических требований, технического задания, описания программного продукта, методики тестирования.

Цель работы: Получения и закрепления навыков разработки технического задания на программные средства.

Задание считается выполненным, если студент:

1. Разработал прототип (упрощенную версию) технического задания на программный продукт в соответствии с ГОСТ 19.201-78 ЕСПД.
2. Разработал прототип (упрощенную версию) руководства программиста в соответствии с ГОСТ 19.504-79 ЕСПД.

Вопросы для самоконтроля:

1. Краткое содержание стандартов комплекса ГОСТ 19.xxx ?

Приложение Г

Граф состояний – один из методов логической организации программного обеспечения.

Цель работы: Получения навыков разработки ПО в соответствии с моделью графа состояний.

Задание считается выполненным, если студент:

1. Графически продемонстрировал граф переходов на основании словесной формулировки алгоритмов управления.
2. Графическое представление графа переходов соответствует приведенному на рисунке 1.

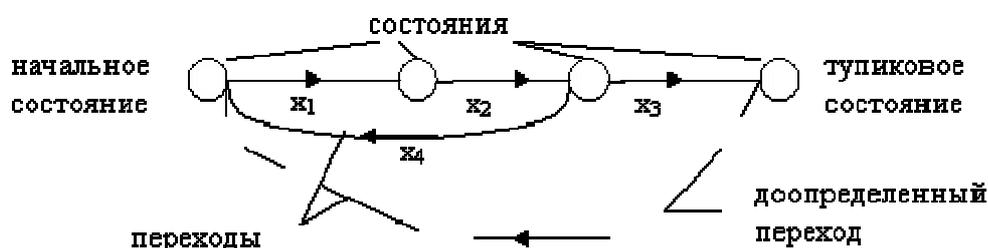


Рисунок 1.

3. В приведенном графе выделено следующее
 - Определенно начальное состояние объекта, из которого начинается процесс управления, и для него вводится начальное состояние на графе переходов, обозначаемое кружком "О".
 - Определяется очередность состояний объекта и необходимое для этого изменение состояний ИМ. Каждый переход из начального состояния в последующие изображается направленной линией, связывающей изображение этих состояний. Стрелка, указывающая направление перехода, обычно изображается в средней части линии. Над стрелкой указывается логическое условие, при выполнении которого осуществляется данный переход, Если из данного состояния возможны несколько переходов, то все они изображаются направленными переходами с соответствующими условиями переходов, при этом все условия должны быть взаимоисключающими, т.е. не должно выполняться более одного условия в данный момент времени.
 - Граф не содержит избыточного числа состояний и переходов, каждое новое состояние вводится только тогда, когда аналогичного состояния на графе не вводилось.
 - Построение графа переходов продолжается до тех пор, пока все последовательности состояний не образуют замкнутые циклы или подграфы.

- В графе нет наличие тупиковых состояний, из которых нет переходов в другие состояния, свидетельствует, как правило, либо об ошибках построения графа переходов, либо о не полноте или ошибочности исходных данных, приведенных в словесной формулировке алгоритма (в этом случае необходимо доопределить и замкнуть граф соответствующим переходом).

Вопросы для самоконтроля:

1. Сформулируйте правила разметки графа состояний для автоматов Мура и Мили.
2. Сформулируйте правила построения графов переходов для автоматов Мура и Мили.
3. Сформулируйте правила построения таблиц переходов-выходов для автоматов Мура и Мили.
4. Сформулируйте правила построения таблицы переходов-выходов автоматов Мура и Мили.

Приложение Д

Системы контроля версий.

Цель работы: Получение и закрепления навыков работы с системами хранения и контроля версий.

Задание считается выполненным, если студент может выполнить следующее:

1. Показать файлы из CVS с определённой меткой.
2. Показать файлы из CVS на определённую дату.
3. Поместить новый проект в CVS.
4. Посмотреть локальные изменения файла по сравнению с последней версией файла из хранилища CVS.
5. Продемонстрировать конфликт и его разрешение в CVS.
6. Установить метку в CVS.
7. Создать ветку в CVS.

Вопросы для самоконтроля:

5. Как идентифицируется хранилище в CVS? В Subversion?
6. Какой информации достаточно, чтобы взять конкретный файл в CVS?
7. Рассказать об операциях checkout, update, commit
8. Что такое метки и ветки?
9. Обзор клиентов cvs?
10. Как поместить новый проект в CVS?
11. Как поместить новый проект в Subversion?
12. Что такое "конфликт"?

Приложение Е

Системы отслеживания ошибок (bug-track).

Цель работы: Получение и закрепления навыков работы с отслеживанием ошибок и управлением качеством продукта.

Задание считается выполненным, если студент может выполнить следующее:

1. Создать отчёт об ошибке в trac.
2. Поставить отметку об устранении ошибки.
3. Поставить отметку о закрытии ошибки.

Вопросы для самоконтроля:

1. Какую проблему решают системы bug-tracking?
2. Какие характеристики ошибки отслеживаются в системах bug-tracking?
3. Классификация пользователей (участников) систем bug-track
4. Функции системы bug-track?

Приложение Ж

Методы проектирования программных средств.

Цель работы: Получения и закрепления навыков взаимодействия исполнителей при проектировании программных средств.

Задание считается выполненным, если студент:

1. Может назвать виды планирования при разработке программного обеспечения.
2. Может привести пример декомпозиции плана.
3. Может выделить фазы создания программного обеспечения.

Вопросы для самоконтроля:

1. Нисходящий анализ процесса управления созданием программного изделия.
2. Установление целей и средства их достижения.
3. Организация планирования разработки программного изделия.
4. Виды планов, связанных с созданием программных изделий.
5. Управление проектом.