Министерство образования и науки РД
Государственное бюджетное профессиональное образовательное учреждение Республики Дагестан
«Кизлярский профессионально-педагогический колледж»

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ ДЛЯ СТУДЕНТОВ СПЕЦИАЛЬНОСТИ 09.02.07 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ

ПМ.02 Сопровождение и обслуживание программного обеспечения компьютерных систем

г. Кизляр

ПМ 04. Сопровождение и обслуживание программного обеспечения *компьютерных систем*

Внедрение и сопровождение информационных систем

Программа профессионального модуля является частью программы подготовки специалистов среднего звена в соответствии с ФГОС по специальности 09.02.07 Информационные системы и программирование в части освоения квалификации: Программист и основных видов профессиональной деятельности (ВПД): «Сопровождение и обслуживание программного обеспечения компьютерных систем».

Цели производственной практики:

Производственная практика направлена на формирование у студентов практических профессиональных умений, приобретение первоначального практического опыта, реализуется в рамках модулей ППССЗ по основным видам профессиональной деятельности для последующего освоения ими общих навыков.

Студент должен закрепить знания такие как:

- основные методы и средства эффективного анализа функционирования программного обеспечения;
- основные виды работ на этапе сопровождения программного обеспечения;
- основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения;
- средства защиты программного обеспечения в компьютерных системах.

Задачи практики:

В результате освоения профессионального модуля обучающийся должен иметь практический опыт:

- в настройке отдельных компонентов программного обеспечения компьютерных систем;
- в выполнение отдельных видов работ на этапе поддержки программного обеспечения компьютерной системы.

Количество времени для данного вида производственной практики – 108 ч.

По данному виду практики проводится дифференцированный зачёт. Студент - практикант, не выполнивший программу практики, не переводится на следующий курс и не допускается к экзамену (квалификационному).

Данный вид практики проходит в организациях профессиональных учреждений. Руководителями практики являются директора организаций.

Задачи производственной (по профилю специальности) практики:

− закрепление полученных теоретических знаний на основе практического участия в деятельности организаций (предприятий) различных форм собственности;

− приобретение опыта профессиональной деятельности и самостоятельной работы,

− сбор, анализ и обобщение материалов для подготовки материалов отчета по практике.

Задачи производственной практики:

− овладение студентами профессиональной деятельностью, развитие профессионального мышления;

− закрепление, углубление, расширение и систематизация знаний, закрепление практических навыков и умений, полученных при изучении дисциплин и профессиональных модулей, определяющих специфику специальности;

− обучение навыкам решения практических задач при подготовке выпускной квалификационной работы;

− проверка профессиональной готовности к самостоятельной трудовой деятельности выпускника.

Государственное бюджетное образовательное учреждение среднего профессионального образования

«Кизлярский профессионально-педагогический колледж»

Утверждаю
зам. директора по УПР
_____Т.Н. Зубкова

Индивидуальное задание
На период производственной практики
ПМ 3. Участие в интеграции программных модулей
Студента группы _____курса _____специальности _____

_____

___
(ФИО студента)

Вопросы, подлежащие изучению и выполнениюза время прохождения практики:

- Техническое задание на разработку ПО
- Внешние и внутренние языки спецификации
- Руководство пользователя
- Руководство программиста
- Автономная отладка и тестирование программного модуля.
- Комплексная отладка и тестирование программного средства.
- Структурные элементы функциональной схемы процесса сертификации программного продукта.
- Структурные элементы процесса сертификации ПП
- Правила сертификации. Сертификат соответствия.
- Порядок заполнения сертификата.

Тема реферата

1. Разбор стандартов и шаблонов представления программ на различных фазах и этапах их разработки
2. Инструменты, методики, нотации построения логических моделей данных и алгоритмов
3. Разбор структуры программных файлов на различных стадиях представления программ – исходник, объектный, исполняемый файл. Управление исключениями - основные синтаксические конструкции, механика исключений, системные исключения, исключения и классы.
4. Государственные и международные стандарты по документированию ПС
5. Отечественные и международные стандарты по документированию метрологии и качеству ПО
6. Организация сбора метрик качества ПП
7. Метрики работы и времени программирования. Метрики ожидаемого числа ошибок в программе
8. Устранение несовершенство программ по метрики Холстеда
9. Виды метрик качества ПП.
10. Интегральные метрики оценки сложности программного продукта качества.
11. Измерительные методы анализа программ.
12. Современные технологии программирования
13. Технология Microsoft.NET
14. Тестирование и отладка программ. Надежность программного обеспечения
15. Экстремальное программирование
16. Суть проектирования . Программирование и тестирование
17. Определение языка моделирования UML. Сущности UML.

18. Внепрограммные сущности. Диаграммы UML.Диаграммы классов, объектов, прецедентов взаимодействий, деятельности, состояний. Архитектура.
19. Технология Microsoft.NET
20. Средства реализации программного кода
21. Документация, создаваемая и используемая в процессе разработки программных средств.
22. Пользовательская документация программных средств.
23. Документация по сопровождению программных средств.
24. Качество программного продукта
25. Принципы модульного программирования.
26. Кодирование информации о товаре
27. Оценка соответствия. Требования к органам, осуществляющим аудит и сертификацию систем качества.
28. Международная электротехническая комиссия

2. Портфолио по практике сдается в последний день практики, руководителю практики от колледжа.

Содержание портфолио:

- Титульный лист
- Цели и задачи практики
- Индивидуальное задание
- Реферат (тема дается руководителем практики)
- Сведения о базе производственной практики
- Индивидуальный календарно-тематический план.
- Дневник практики (по форме)
- Характеристика студента с указанием оценки, заверенная директором учреждения
- Отзыв о практике

Руководитель практики _____

Утверждаю
зам. директора по УПР
_____Т.Н. Зубкова

ВИДЫ РАБОТ
ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ
ПМ 2. Участие в интеграции программных модулей

1. Знакомство с базовым учреждением
2. Настройка и обслуживание аппаратного обеспечения и операционную систему персонального компьютера;
3. Настройка и обслуживание периферийных устройств персонального компьютера, и компьютерную оргтехнику;
4. Осуществлять ввод и обмен данными между персональным компьютером и периферийными устройствами и ресурсами локальных компьютерных сетей;
5. Создание на персональном компьютере текстовых документов, таблиц, презентаций и баз данных;
6. Осуществлять навигацию по ресурсам, поиск, ввод и передачу данных с помощью технологий и сервисов Интернета;
7. Создавать и обрабатывать цифровые изображения и объекты мультимедиа;
8. Обеспечивать меры по обеспечению информационной безопасности.
9. Выполнение реферата по заданной теме

Зав практикой _____Мамедова Е. Г.

Руководитель практики_____Искандырова А. А.

Задания производственной практики.

Начальный этап нисходящей разработки

1. Общие положения

1.1. Этапы разработки программы

Составление внешней спецификации. Цель – уточнение постановки задачи, а именно:

- точная формулировка задачи и всех вводимых ограничений;
- определение состава и структуры входных и выходных данных и формы их ввода и вывода;
- описание *аномалий* - ситуаций, при которых решение не может быть получено (выход данных за границы диапазона, деление на ноль и т.п.);
- описание *тестов* - примеров, на которых будет проверяться работа программы;
- описание *метода решения* - связей входных и выходных величин, общей идеи и схемы решения задачи.

Спецификация представляет собой интерфейс между разработчиком программы и ее пользователем.

Большая часть желательных качеств программы должна быть сконцентрирована в ее спецификации, так, чтобы программа после первой ее реализации требовала улучшений в минимально возможной мере. Исключением является быстродействие, которое можно повышать уже после реализации программы.

•***Проектирование алгоритма***. Цель – создание алгоритма, *понятного человеку.* Центральный и наиболее трудоемкий этап. Требует творчества и не может быть полностью формализован. Созданный на этом этапе текст алгоритма – носитель всей информации о решении, поэтому для записи алгоритма целесообразно использовать наиболее универсальные и одновременно точные средства – «отказ от красивого и умного в пользу незамысловатого и четкого». Такими средствами являются полуформальные языки, используемые для проектирования – *псевдокоды*.

•***Кодирование программы*** состоит в переводе алгоритма с одного языка, достаточно жесткого (псевдокода), на другой, полностью формализованный (язык программирования).

Если первые два этапа выполнены тщательно, то кодирование выполняется без особых трудностей.

•***Отладка и тестирование***. Цель – получение работоспособной программы. Очень важный этап, требующий особого рассмотрения и здесь представленный только самыми основными положениями.

Отладка – процесс изменения программы с целью исправления ошибок.

Тест – совокупность специально подобранных входных и соответствующих им выходных данных, используемая для контроля правильности программы.

Тестирование – исполнение программы на наборе тестов на компьютере.

Отладка включает тестирование.

Принципиальная трудность проектирования тестов состоит в практической невозможности составления всех тестовых наборов данных для всех возможных режимов работы алгоритма. Поэтому задача проектирования тестов сводится к проектированию *ограниченного* их набора,

гарантирующего *с достаточной достоверностью* правильную работу программы *во всех* практически значимых режимах.

Содержание тестов определяется спецификацией задачи и логикой ее решения.

Функциональные тесты составляются на уровне спецификации, до решения. Будущий алгоритм рассматривается как *«черный ящик»* - функция с неизвестной структурой, преобразующая входы в выходы. Суть функциональных тестов: каким бы способом ни решалась задача, при заданных входных значениях должны получиться соответствующие выходные значения.

Структурные тесты составляются для проверки логики решения, или логики работы уже готового алгоритма. Логика определяется последовательностью операций, их условным выполнением или повторением. Совокупность структурных тестов должна обеспечить проверку каждой из таких конструкций.

Чаще всего совокупность тщательно составленных функциональных тестов покрывает множество структурных тестов.

Длительность и сложность данного этапа полностью определяется качеством выполнения предыдущих этапов.

•***Документирование***. Для удобства описания представлено как некоторый этап, но по сути таковым не является.

Элементы документации – все элементы проекта, полученные при его разработке, т.е. описание результатов отдельных этапов. Если процесс разработки программы организован правильно, то документация появляется постепенно и процесс ее создания сопровождает все этапы.

Этапы обладают свойством преемственности, но принципиально важно, чтобы эта преемственность реализовалась сверху вниз, т.е. чтобы предшествующие этапы захватывали последующие, а не наоборот. Возвратов назад по возможности быть не должно.

1.2. Состав документации по отдельным этапам

● ***Используемыеобозначения***

<a>– значение выражения a;

::= – есть по определению;

{ a, b } – альтернативные компоненты; описание ситуации, когда в конкретном случае будет выбран один из вариантов, a или b;

[a] – возможный компонент; в конкретном случае a может отсутствовать.

Остальные обозначения интуитивно понятны и будут поясняться по мере появления.

● Для фиксации результатов выполнения отдельных этапов удобно использовать некоторую схему, одновременно задающую порядок разработки. Ниже предлагается схема, включающая ряд пунктов, снабженных номерами; целесообразно использовать эту нумерацию и не менять ее от разработки к разработке.

● ***Состав документации***

Пункты 1 –6 относятся к этапу составления внешней спецификации.

Пункт 7 – результат проектирования алгоритма; развитие п. 6. Эти пункты пересекаются, причем при правильной разработке все принципиальные моменты решения должны быть отображены в п. 6 и технически реализованы в п. 7.

Пункт 8 – результат кодирования. К взаимосвязи пп. 7 и 8 относится то же, что к пп. 6 и 7.

1. Задача

<четко сформулированное условие задачи>;

<упрощающие предположения и ограничения>

Состав входных данных. Описать так каждое входное данное

2. Входные данные

<тип><имя> – <смысл>;<структура>;<диапазон>;<точность>; [<формат>;]

Входная форма

< желаемое расположение данных на носителе>

Состав выходных данных

2. Выходные данные

<тип><имя> – <смысл>;<структура>;<диапазон>;<точность>; [<формат>;]

Выходная форма

Обр1
Обр2
Обр3
Обр4
Обр5

<Заголовок программы>
<Входные данные>
[<Промежуточные данные>]
<Результаты>
<Реакции на аномалии>

Обр1 и т.д. – образцы вывода. Будем ссылаться на них в алгоритме для сокращения записи.
Номера образцов можно брать произвольно.
Обязательные компоненты выходной формы – образцы 1, 2 и 4 или 5.

4. Аномалии входных данных (недопустимые ситуации)

<описание аномалий>;

Информация, выводимая для каждой из аномалий. Детализация образца 5.

<реакции на аномалии>

Анализ таких ситуаций с точки зрения алгоритмизации является задачей, требующей точно такого же подхода, как и основная задача.

5. Функциональные тесты

№ теста	Входные данные	Ожидаемый результат	Смысл теста
<номер>	<набор значений входных данных>	<соответствующие значения результата>	<смысл ситуации; зависит от задачи>

6. Метод

[<название, если используется уже известный метод>];

<состав используемых при решении промежуточных величин>;

[<связи между данными в математических терминах, в виде формул>];

<словесное описание способа и основных этапов решения>

7. Алгоритм на псевдокоде

алг<имяалгоритма> (<список имен входных и выходных данных>);

арг

<описания входных данных>;
рез

<описания выходных данных>;
нач

<описания промежуточных данных>;
<операции >
кон;
кон<имя алгоритма>;

Курсивом выделены ключевые слова – фиксированные последовательности символов с предопределенным смыслом.

В любое место алгоритма можно вставить комментарий:

{<пояснительный текст>}

Операторы псевдокода см. в примере

Операторы псевдокода приводятся в примере.

8. Программа

9. Структурные тесты

Строго говоря, эти тесты должны разрабатываться сразу после составления алгоритма. Код программы в них привносить ничегоне должен. Этот пункт здесь размещен последним только в методических целях, чтобы в очевидных случаях, когда структурные тесты покрываются функциональными, его можно было опустить.

Для описания структурных тестов может быть использована та же схема, что и для функциональных тестов (см. п.5).

Набор тестов строится так, чтобы проверить правильность выполнения:

- всех ветвлений (условных действий);
- всех циклических действий.

1.3. Нисходящая разработка и нисходящая отладка

Суть *нисходящей разработки* (нисходящего проектирования,проектирования сверху вниз) состоит в пошаговой декомпозиции (разложении) задачи на точно определенные подзадачи.

Формальных критериев декомпозиции не существует, поэтому процесс, особенно для нестандартных задач, является в большой степени творческим и требует навыка и опыта. Парадоксальным на первый взгляд, но наиболее конструктивным критерием выделения подзадач является такой: каждая подзадача должна быть сформулирована кратко, точно и емко («Найти сумму..», «Проверить наличие элементов …» и т.д.).

Процесс разработки является иерархическим.На каждом уровне проектирования полагаем, что каждая выделенная подзадача реализуема, и, не интересуясь деталями этой реализации, рассматриваем ее как единое обобщенное действие, называемое *абстракцией*. Подзадача некоторого уровня раскрывается на следующем, также путем разложения на подзадачи – и т.д., до получения подзадач, алгоритм решения которых можно непосредственно записать на языке программирования.

На любом уровне проектирования одни элементы алгоритма уже доведены до исполнительного уровня и могут быть закодированы, другие же представлены в виде абстракций. *Нисходящая*

отладка представляет собой постепенный процесс отладки такого незавершенного кода программы на каждом уровне.

В основе ее лежит тот факт, что любая подзадача связана с остальными только своим *интерфейсом*, т.е. совокупностью получаемых ею входных данных и передаваемых дальше выходных данных. Можно взять некоторый тест и вместо операций по решению подзадачи вставить операции, формирующие соответствующие тесту выходные значения. Совокупность таких операций называется *заглушкой* (или *имитатором*).

Заглушка должна только получить входные данные подзадачи и напрямую сформировать результаты согласно тесту, поэтому тексты заглушек должны быть максимально простыми. Подставив вместо подзадач заглушки, мы будем иметь, с одной стороны, законченную с точки зрения компьютера программу; с другой стороны, ее результаты работы *на рассматриваемом* тесте не будут отличаться от результатов итоговой программы.

Такой процесс отладки дает на каждом уровне два важных результата:

- возможность отладки уже готовых частей программы;
- возможность отладки взаимодействия подзадач, т.е. логики передачи данных между ними.

При этом то, что уже отлажено, далее меняться не должно, и все усилия можно сосредоточить на раскрытии еще не решенных подзадач.

2. Примеры нисходящей разработки

Основой для выполнения индивидуальных заданий является первый пример.

3. Задание по работе

Выполнить разработку уровня 0 для индивидуальной задачи.

Можно взять реальную задачу с достаточно четко сформулированным условием и приемлемой сложностью. В таком случае возможно объединение в бригады из 2 – 3 человек (в зависимости от сложности задачи).Альтернативный вариант – взять учебную задачу из приведенного ниже списка; номер задачи определяет преподаватель.

4. Список задач

1. Если в матрице A над главной диагональю содержится хотя бы один положительный элемент, а в матрице B – хотя бы один элемент, больший заданного числа, то найти скалярные произведения всех строк матрицы A на все столбцы матрицы B (1-й строки матр. A на 1-й, 2-й, …. столбец матрицы B; 2-й строки матр. A на 1-й, 2-й, …. столбец матрицы B; и т.д.)

2. Определить диапазон значений элементов заданной матрицы. Проверить, попадает ли этот диапазон в заданный интервал; если нет, то сформировать одномерный массив из элементов, не попадающих в интервал.

3. Найти номера строк матрицы, где расположены максимальные элементы столбцов.

Если все максимумы столбцов сосредоточены в верхней половине матрицы, то распечатать эти максимумы.

4. Найти максимальный среди отрицательных и минимальный среди положительных элементов матрицы.

5. Если все элементы массива B меньше числа XB, а массива C – меньше числа XC, то сформировать всевозможные суммы пар $B_k + C_p$, найти среди них максимальную сумму и номера составляющих ее элементов.

6. Если построчные суммы элементов матрицы A упорядочены по возрастанию, то найти минимальный элемент матрицы и его индексы.

7. Найти среднее арифметическое элементов первого упорядоченного по убыванию столбца матрицы.Если матрица не содержит ни одного отрицательного элемента, изменить все ее элементы путем вычитания из них найденного среднего арифметического.

8. Для каждой строки матрицы найти значения отрицательных элементов.

Просуммировать все элементы той строки, где число отрицательных элементов максимально.

9. В заданной матрице поставить на первое место строку с максимальной суммой элементов. Уменьшить элементы столбца, содержащего максимальный элемент матрицы, на заданное число.

10. Если суммы элементов строк матрицы упорядочены по убыванию, то найти максимальный по модулю элемент матрицы и его индексы.

11. Задан целочисленный одномерный массив X. Если все его элементы положительны, то найти максимальное из значений факториалов элементов.

12. Если диагональные элементы квадратной матрицы упорядочены по возрастанию, уменьшить строку с максимальной суммой элементов на заданное число.

13. Дана матрица Аиз m строк и n столбцов. Для строки с максимальным количеством отрицательных элементов подсчитать общее произведение элементов.

14. Дан одномерный массив. Подсчитать число нулей между первым и последним положительными элементами.

15. Даны два одномерных массива аи b. если ни в одном из них нет совпадающих элементов , то проверить, есть ли в массиве ахотя бы один элемент, совпадающий хотя бы с одним элементом массива b.

16. Если в заданной матрице Аиз m строк и n столбцов содержится более двух строк, все элементы которых меньше заданного числа P, то определить номер строки с наибольшим количеством таких элементов.

17. Упорядочить по убыванию все элементы одномерного массива, расположенные между двумя первыми совпадающими элементами.

18. Если строки заданной матрицы Аиз m строк и n столбцов расположены в порядке возрастания числа нулевых элементов в них, то подсчитать число нулевых элементов во всей матрице, иначе определить номер строки с максимальным количеством нулей.

19. Даны две матрицы: Аиз m строк и n столбцов и Виз p строк иq столбцов. В той из них, где элементы строки с заданным номером упорядочены по возрастанию, подсчитать число нулей в каждой строке.

20. Даны два одномерных массива аи b.

Сформировать третий массив с из совпадающих элементов массивов аи b. Если все элементы массива с положительны, то упорядочить его по убыванию.

21. Подсчитать количество отрицательных элементов в каждой строке матрицы Аиз m строк и n столбцов. Если в каких-либо строках есть одинаковое количество отрицательных элементов, то заменить значения этих элементов в первой из таких строк на их модули.

22. Если матрица Аиз m строк и n столбцов содержит ровно k нулевых элементов, то подсчитать сумму элементов строк, лежащих между первой и последней строкой с нулями.

23. В матрице Аиз m строк и n столбцов упорядочить по возрастанию те строки, элементы которых исходно упорядочены по убыванию. Если таких строк нет, то проверить, есть ли строки с нулевыми элементами.

24. Найти в матрице Аиз m строк и n столбцов первую упорядоченную по возрастанию строку. Если такой строки нет, то упорядочить по возрастанию первую строку, все элементы которой положительны.

25. Дана матрица A из m строк и n столбцов. Если хотя бы в одной ее строке все элементы отрицательны, то упорядочить элементы этой строки по убыванию.

26. Даны два одномерных массива: аиз mи bиз n элементов.

Если какой-либо элемент первого массива совпадает с каким-либо элементом второго, то найти первый отрицательный элемент в массиве a.

27. Если в матрице A из m строк и n столбцов все элементы над главной диагональю положительны, то найти минимум из построчных сумм матрицы.

28. упорядочить по возрастанию элементы одномерного массива bиз n элементов, расположенные между первыми двумя его отрицательными элементами.

29. Если все элементы матрицы A из m строк и n столбцов положительны, то проверить, является максимальный элемент в ней единственным.

Министерство образования и науки Республики Дагестан

Государственное бюджетное профессиональное образовательное учреждение

«Кизлярский профессионально-педагогический колледж»

ОТЧЕТ

по производственной практике

ПМ03 Участие в интеграции программных модулей

ФИО
студента_____

Отделение _____группа _____курс
_____

Специальность:
_____
_____

Руководитель практики от колледжа
_____

Производственную практику проходил в
_____
_____

Руководитель практики от
учреждения_____

Сроки практики_____

Оценка _____

Государственное бюджетное образовательное учреждение
среднего профессионального образования
« Кизлярский профессионально – педагогический колледж»

Дневник
прохождения производственной практики
ПМ03Участие в интеграции программных модулей

ФИО студента_____

Отделение _____группа _____курс _____

Специальность:

_____

Руководитель практики от колледжа _____

Производственную практику проходил в _____

_____

_____

Руководитель практики от
учреждения_____

Сроки практики_____

Оценка _____

ОБРАЗЕЦ ДНЕВНИКА

Записи о работах, выполненных во время практики

Дата	Краткое содержание выполненных работ	Оценка и замечание руководителя от учреждения

Сведения о базовом учреждении производственной практики

_____

_____

(наименование учреждения, адрес, телефон)

Директор учреждения

_____

(Ф.И.О.)

Руководитель практики от учреждения

_____

(Ф.И.О.)

Индивидуальный календарно – тематический план.

Дата	Деятельность студента