

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет
им. Г. И. Носова»
Многопрофильный колледж



**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ
ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ**

**Профессиональный модуль
ПМ.03. Участие в интеграции программных модулей**

**Специальность 09.02.03 Программирование в компьютерных
системах - базовой подготовки**

Магнитогорск, 2017

ОДОБРЕНО:

Предметно-цикловой комиссией
Информатика и вычислительная техника

Председатель И.Г.Зорина

Протокол № 7 от 14 марта 2017

Методической комиссией МпК
Протокол №4 от «23» марта 2017г

Составитель:

преподаватель ФГБОУ ВО МГТУ МпК Ирина Геннадьевна Зорина/
преподаватель ФГБОУ ВО МГТУ МпК Людмила Александровна
Фетисова/
преподаватель ФГБОУ ВО МГТУ МпК Власта Диляуровна Тутарова/

Методические указания по самостоятельной работе разработаны на основе рабочей программы профессионального модуля ПМ.03. Участие в интеграции программных модулей

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К современному специалисту общество предъявляет широкий перечень требований, среди которых немаловажное значение имеет наличие у выпускников определенных способностей и умения самостоятельно добывать знания из различных источников, систематизировать полученную информацию, давать оценку конкретной ситуации. Формирование такого умения происходит в течение всего периода обучения через организацию самостоятельной работы. Процесс самостоятельной работы позволяет ярко проявиться индивидуальным способностям личности. Только через самостоятельную работу студент может стать высококвалифицированным компетентным специалистом, способным к постоянному профессиональному росту.

Задачи самостоятельной работы:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать справочную документацию и специальную литературу;
- развитие познавательных способностей и активности: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

Самостоятельная работа является одним из видов учебных занятий и предполагает активную роль студента в ее планировании, осуществлении и контроле.

Самостоятельная работа является обязательной для каждого студента. Самостоятельная работа может осуществляться индивидуально или группами студентов в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности, уровня умений студентов.

Контроль результатов внеаудиторной самостоятельной работы студентов может осуществляться в пределах времени, отведенного на обязательные учебные занятия и внеаудиторную самостоятельную работу студентов по междисциплинарному курсу, может проходить в письменной, устной или смешанной форме, с представлением изделия или продукта творческой деятельности студента.

В качестве форм и методов контроля внеаудиторной самостоятельной работы студентов могут быть использованы - проверка выполненной работы преподавателем, тестирование, самоотчеты, контрольные работы, защита творческих работ, зачеты, экзамен.

Критериями оценки результатов внеаудиторной самостоятельной работы являются:

- уровень освоения учебного материала;
- умение использовать теоретические знания при выполнении практических задач;
- сформированность общеучебных умений;
- обоснованность и четкость изложения ответа;
- оформление материала в соответствии с требованиями

ВИДЫ ЗАДАНИЙ ДЛЯ САМОСТОЯТЕЛЬНОЙ ВНЕАУДИТОРНОЙ РАБОТЫ

В ходе занятий Вам будут предлагаться типовые задания. Данные методические рекомендации призваны оказать помощь в организации самостоятельной внеаудиторной работы при выполнении домашних заданий.

МДК 03.01 Технология разработки программного обеспечения

Тема 03.01.01 Модели процесса разработки программного обеспечения

Выполнение практических заданий

Цель работы: построение каскадной модели

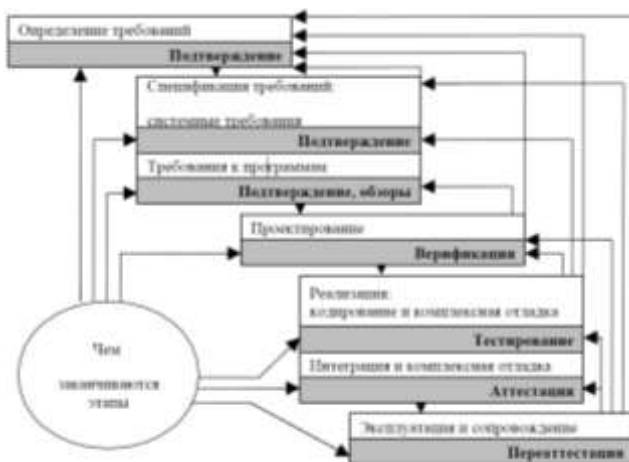
Характерные черты каскадной модели:

- завершение каждого этапа (они почти те же, что и в классической модели) проверкой полученных результатов, с целью устранить как можно большее количество проблем, связанных с разработкой изделия;
- циклическое повторение пройденных этапов (как в классической модели).

Мотивация каскадной модели связана с так называемым управлением качеством программного обеспечения. В связи с ней уточняются понятия этапов, некоторые из них структурируются (спецификация требований и реализация). На рисунке приведена схема каскадной модели, построенная как модификация классической итерационной модели. В каждом блоке, обозначающем этап, указано действие, которым этап завершается (наименования этих действий отмечены серым фоном). Из рисунка видно, что в этой модели тестирование не выделяется в качестве отдельного этапа, а считается лишь порогом, через который нужно перейти, чтобы завершить этап, точно так же как и другие подобные действия.

В соответствии с каскадной моделью завершение этапа определения системных требований включает фиксацию их в виде специальных документов, называемых обзорами того, что требуется от системы (описание функций), а спецификация требований к программам – подтверждением выполнения зафиксированных в обзорах функций в планируемых к реализации программах.

Кроме того, подтверждение предполагается и на первом этапе, т.е. после определения требований. Это отражает тот факт, что полученные требования необходимо согласовывать с заказчиком.



Этап планирования и анализа требований

Целью этого этапа является получение требований и выработка производных от них требований для этапа оценки безопасности.

Входные данные:

- требования к системе, аппаратный интерфейс, архитектура системы;
- план разработки ПО;
- стандарты на требования к ПО.

Первичный результат – данные о требованиях.

Этап разработки

Цель этого этапа является создание архитектуры ПО и требований НУ, а также выработка производных от них требований для этапа оценки безопасности.

Входные данные:

- данные о требованиях к ПО;
- план разработки ПО;
- стандарты проектирования ПО.

Первичный результат – описание разработки, включающее архитектуру ПО и требования

Вариант 1. Построить каскадную модель для программного модуля «Учет успеваемости студентов». Программный модуль предназначен для оперативного учета успеваемости студентов в сессию заместителем директора по СПО и учебной частью. Сведения об успеваемости студентов должны храниться в течение всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.

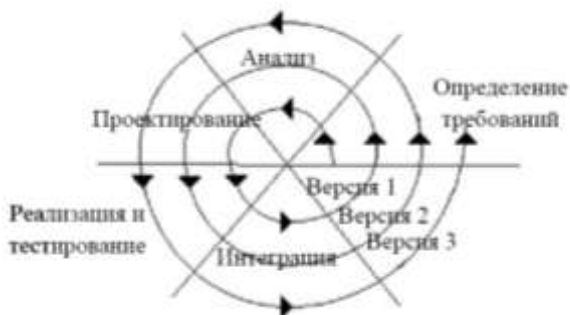
Формы контроля: своевременное представление выполненных заданий

Критерии оценки: - правильность построения модели оформление.

**Тема 03.01.02 Основные принципы процесса разработки
программного обеспечения
Выполнение практических заданий**

Цель работы: построение спиральной модели и проведение сравнительного анализа

Спиральная модель разработки ПО, в тех или иных версиях используемая во множестве конкретных прикладных методик, построена на следующем шаблоне. Прежде всего, в ходе общения с заказчиком определяется набор наиболее важных и критичных возможностей будущей системы. Далее совместными усилиями определяются желаемые сроки для реализации этой базовой функциональности. Формируется план, начинаются работы и отслеживается их выполнение.



Работа выполняется согласно алгоритму, приведённому ниже.

Шесть шагов спиральной модели:

1. В процессе общения с заказчиком формируется общее видение проекта, а также описываются функциональные возможности, которые необходимо реализовать в определенные сроки с нужным качеством.
2. Расставляются приоритеты, задающие порядок реализации основных функциональных возможностей.
3. Согласовываются временные рамки проекта. Часто для этого применяются методики стоимостного прогнозирования. Далее исполнитель решает, сколько функциональных возможностей в соответствии с их приоритетами удастся реализовать в оговоренный срок.
4. На данном этапе определяются архитектура и ядро будущей системы. Ядро должно представлять собой законченный работающий вариант системы с небольшим набором необходимых возможностей. Не исключено, что заказчик видит архитектуру как жесткую конструкцию и

не предусматривает средств ее расширения для обеспечения дополнительной или менее приоритетной функциональности. Поэтому далее определяется способ реализации требований с более низкими приоритетами – это можно делать, например, с помощью встроенного языка сценариев или подключением динамических библиотек, для чего необходимо определить внутренние интерфейсы ядра.

Этот шаг выполняется, как правило, в два и большее число итерационных циклов.

5. Готовится план работ. Он ориентирован на сроки, определенные на третьем этапе, и нацелен на скорейшую реализацию ядра системы. Взаимодействуя с работающим прототипом, заказчик быстрее и точнее вырабатывает и уточняет дальнейшие требования и корректирует приоритеты.

6. Разработка системы в соответствии с планом.

Вариант 1. Построить спиральную модель для программного модуля «Личные дела студентов». Программный модуль предназначен для получения сведений о студентах сотрудниками учебной части и отдела кадров. Сведения должны храниться в течение всего срока обучения студентов и использоваться при составлении справок и отчетов.

Формы контроля: своевременное представление выполненных заданий

Критерии оценки: - правильность построения модели оформление.

Тема 03.01.03

Основные подходы к интегрированию программных модулей Выполнение практических заданий

Цель работы: ознакомиться с правилами написания технического задания.

Разработка технического задания выполняется в следующей последовательности. Прежде всего, устанавливают набор выполняемых функций, а также перечень и характеристики исходных данных. Затем определяют перечень результатов, их характеристики и способы представления.

Далее уточняют среду функционирования программного обеспечения: конкретную комплектацию и параметры технических средств, версию используемой операционной системы и, возможно, версии и параметры другого установленного программного обеспечения, с которым предстоит взаимодействовать будущему программному продукту.

В случаях, когда разрабатываемое программное обеспечение собирает и хранит некоторую информацию или включается в управление каким-

либо техническим процессом, необходимо также четко регламентировать действия программы в случае сбоев оборудования и энергоснабжения.

1. Общие положения

1.1. Техническое задание оформляют в соответствии с ГОСТ 19.106—78 на листах формата А4 и А3 по ГОСТ 2.301—68, как правило, без заполнения полей листа. Номера листов (страниц) проставляют в верхней части листа над текстом.

1.2. Лист утверждения и титульный лист оформляют в соответствии с ГОСТ 19.104—78. Информационную часть (аннотацию и содержание), лист регистрации изменений допускается в документ не включать.

1.3. Для внесения изменений и дополнений в техническое задание на последующих стадиях разработки программы или программного изделия выпускают дополнение к нему. Согласование и утверждение дополнения к техническому заданию проводят в том же порядке, который установлен для технического задания.

1.4. Техническое задание должно содержать следующие разделы:

- введение;
- наименование и область применения;
- основание для разработки;
- назначение разработки;
- технические требования к программе или программному изделию;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приемки;
- приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них. При необходимости допускается в техническое задание включать приложения.

Пример 1. Разработать техническое задание на программный продукт, предназначенный для наглядной демонстрации школьникам графиков функций одного аргумента $y = f(x)$. Разрабатываемая программа должна рассчитывать таблицу значений и строить график функций на заданном отрезке по заданной формуле и менять шаг аргумента и границы отрезка. Кроме этого, программа должна запоминать введенные формулы.

Порядок выполнения работы

1. Разработать техническое задание на программный продукт.
2. Оформить работу в соответствии с ГОСТ 19.106—78. При оформлении использовать MS Office.

Формы контроля: своевременное представление выполненных заданий

Критерии оценки: - правильность составленного технического задания

Тема 03.01.04
Основные методы и средства эффективной разработки
Выполнение практических заданий

Цель работы: Приобретение навыков исследовательской работы в процессе решения задач моделирования процессов управления проектами по созданию продукции на основе CALS –технологий

Задачами исследования являются:

1. Изучение основных положений и структуры стандарта по управлению проектами.
2. Анализ процессов и групп процессов, а также структуры зависимостей между процессами управления проектом и областями знаний (группа процессов управления, группа процессов планирования, группа процессов управления интеграцией проекта, группа процессов по управлению содержанием проекта, группы процессов управления сроками проекта, управления стоимостью проекта, управления качеством проекта).
3. Управления человеческими ресурсами, управления коммуникациями, управления рисками проекта, управления поставками проекта.
4. Разработка структурно-функциональных моделей процессов управления проектами.
5. Разработка информационных моделей процессов управления проектами.
6. Разработка инфологической и даталогической моделей исследуемых процессов управления.
7. Анализ результатов моделирования с целью разработки предложений по обоснованию выбора инструментальных средств автоматизации моделируемых процессов
8. Разработка предложений по интеграции исследуемых процессов управления проектом с АСУП и PDM-системой.

Основные организационно-методические и нормативные материалы Основными базовыми положениями для решения задач лабораторной работы являются:

единое информационное пространство (ЕИП), создаваемое путем интеграции автоматизированных процессов всех участников жизненного цикла продукции;

методология и методы формирования ЕИП в соответствии со стандартом STEP (ISO-10303);

модель жизненного цикла продукции, процессы и организация жизненного цикла АС и ПО, которые регламентируются стандартом ISO/IEC-12207;

стандарт PMBOK ,регламентирующий организацию систем управления проектами.

назначение и содержание следующих моделей :

- Информационная модель. Эта модель представляет собой схему, в которой описаны взаимодействия между ключевыми сущностями проекта.
- Инфологическая модель. Эта модель является промежуточным звеном между описанием процесса и готовой реляционной БД. Для того чтобы качественно построить БД ,необходимо проанализировать сущности, участвующие в процессе ,и их свойства.
- Даталогическая модель. Она представляет собой фактически описание реляционных таблиц с ключевыми полями и связями между ними.

Инструментальные средства

Для построения структурно-функциональной модели объекта исследования в нотации стандарта IDEF0, а также информационных моделей процессов управления проектами следует использовать средства инструментальной поддержки организационного проектирования в среде BP win/ER win.

Задачу структурно-функционального моделирования можно решить при помощи стандарта IDEF0.

IDEF0 – это методология и графическая нотация, предназначенная для формализации и описания бизнес-процессов.

В IDEF0 основными элементами являются:

Работы (Activity).

Работы обозначают поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты (см. раздел 1.2.2. Маклаков С. В. BPwin и ERwin: CASE-средства для разработки информационных систем).

Стрелки (Arrow).

Взаимодействие работ с внешним миром и между собой описывается в виде стрелок. Стрелки представляют собой некую информацию и именуются существительными (например, "Заготовка", "Изделие", "Заказ") (см. раздел 1.2.3. Маклаков С. В. BPwin и ERwin: CASE-средства для разработки информационных систем).

Существует пять видов стрелок:

- 1.Вход (Input) .
- 2.Управление (Control).
3. Выход (Output).
- 4.Механизм (Mechanism).
- 5.Вызов (Call).

В IDEF0 нет временных последовательностей (WorkFlow), только логические отношения между работами.

Процессы в области знаний	Группы процессов управления проектом				
	Группа процессов инициации	Группа процессов планирования	Группа процессов исполнения	Группа процессов мониторинга и управления	Группа завершающих процессов
4. Интеграция управления проектом	Разработка Устава проекта 3.2.1.1 (4.1) Разработка предварительного описания содержания проекта 3.2.1.2 (4.2)	Разработка плана управления проектом 3.2.2.1 (4.3)	Руководство и управление исполнением проекта 3.2.3.1 (4.4)	Мониторинг и управление работами проекта 3.2.4.1 (4.5) Общее управление изменениями 3.2.4.2 (4.6)	Закрытие проекта 3.2.5.1 (4.7)
5. Управление содержанием проекта		Планирование содержания 3.2.2.2 (5.1) Определение содержания 3.2.2.3 (5.2) Создание ИСР 3.2.2.4 (5.3)		Подтверждение содержания 3.2.4.3 (5.4) Управление содержанием 3.2.4.4 (5.5)	
6. Управление сроками проекта		Определение состава операций 3.2.2.5 (6.1) Определение взаимосвязей операций 3.2.2.6 (6.2) Оценка ресурсов операций 3.2.2.7 (6.3) Оценка длительности операций 3.2.2.8 (6.4) Разработка расписания 3.2.2.9 (6.5)		Управление расписанием 3.2.4.5 (6.6)	
7. Управление стоимостью проекта		Стоимостная оценка 3.2.2.10 (7.1) Разработка бюджета расходов 3.2.2.11 (7.2)		Управление стоимостью 3.2.4.6 (7.3)	
8. Управление качеством проекта		Планирование качества 3.2.2.12 (8.1)	Процесс обеспечения качества 3.2.3.2 (8.2)	Процесс контроля качества 3.2.4.7 (8.3)	
9. Управление человеческими ресурсами проекта		Планирование человеческих ресурсов 3.2.2.13 (9.1)	Набор команды проекта 3.2.3.3 (9.2) Развитие команды проекта 3.2.3.4 (9.3)	Управление командой проекта 3.2.4.8 (9.4)	
10. Управление коммуникациями проекта		Планирование коммуникаций 3.2.2.14 (10.1)	Распространение информации 3.2.3.5 (10.2)	Отчетность по исполнению 3.2.4.9 (10.3) Управление участниками проекта 3.2.4.10 (10.4)	
11. Управление рисками проекта		Планирование управления рисками 3.2.2.15 (11.1) Идентификация рисков 3.2.2.16 (11.2) Качественный анализ рисков 3.2.2.17 (11.3) Количественный анализ рисков 3.2.2.18 (11.4) Планирование реагирования на риски 3.2.2.19 (11.5)		Мониторинг и управление рисками 3.2.4.11 (11.6)	
12. Управление поставками проекта		Планирование покупок и приобретения 3.2.2.20 (12.1) Планирование контрактов 3.2.2.21 (12.2)	Запрос информации у продавцов 3.2.3.6 (12.3) Выбор продавцов 3.2.3.7 (12.4)	Администрирование контрактов 3.2.4.12 (12.5)	Закрытие контракта 3.2.5.2 (12.6)

Формы контроля: своевременное представление выполненных заданий

Тема 03.01.05

Основы верификации и аттестации программного обеспечения

Выполнение практических заданий

Цель работы: Приобретение навыков исследовательской работы в процессе решения задач моделирования процессов управления проектами по созданию продукции на основе CASE-технологий

Основные организационно-методические и нормативные материалы
Основными базовыми положениями для решения задач являются: единое информационное пространство (ЕИП), создаваемое путем интеграции автоматизированных процессов всех участников жизненного цикла продукции;

методология и методы формирования ЕИП в соответствии со стандартом STEP (ISO-10303);

модель жизненного цикла продукции, процессы и организация жизненного цикла АС и ПО, которые регламентируются стандартом ISO/IEC-12207;

стандарт PMBOK ,регламентирующий организацию систем управления проектами.

назначение и содержание следующих моделей :

Информационная модель. Эта модель представляет собой схему, в которой описаны взаимодействия между ключевыми сущностями проекта.

Инфологическая модель. Эта модель является промежуточным звеном между описанием процесса и готовой реляционной БД. Для того чтобы качественно построить БД ,необходимо проанализировать сущности, участвующие в процессе ,и их свойства.

Даталогическая модель. Она представляет собой фактически описание реляционных таблиц с ключевыми полями и связями между ними.

Инструментальные средства

Для построения структурно-функциональной модели объекта исследования в нотации стандарта IDEF0, а также информационных моделей процессов управления проектами следует использовать средства инструментальной поддержки организационного проектирования в среде BP win/ER win.

Задачу структурно-функционального моделирования можно решить при помощи стандарта IDEF0.

IDEF0 – это методология и графическая нотация, предназначенная для формализации и описания бизнес-процессов

Процессы в области знаний	Группы процессов управления проектом				
	Группа процессов инициации	Группа процессов планирования	Группа процессов исполнения	Группа процессов мониторинга и управления	Группа завершающих процессов
4. Интеграция управления проектом	Разработка Устава проекта 3.2.1.1 (4.1) Разработка предварительного описания содержания проекта 3.2.1.2 (4.2)	Разработка плана управления проектом 3.2.2.1 (4.3)	Руководство и управление исполнением проекта 3.2.3.1 (4.4)	Мониторинг и управление работой проекта 3.2.4.1 (4.5) Общее управление изменениями 3.2.4.2 (4.6)	Закрытие проекта 3.2.5.1 (4.7)
5. Управление содержанием проекта		Планирование содержания 3.2.2.2 (5.1) Определение содержания 3.2.2.3 (5.2) Создание ИСР 3.2.2.4 (5.3)		Подтверждение содержания 3.2.4.3 (5.4) Управление содержанием 3.2.4.4 (5.5)	
6. Управление сроками проекта		Определение состава операций 3.2.2.5 (6.1) Определение взаимосвязей операций 3.2.2.6 (6.2) Оценка ресурсов операций 3.2.2.7 (6.3) Оценка длительности операций 3.2.2.8 (6.4) Разработка расписания 3.2.2.9 (6.5)		Управление расписанием 3.2.4.5 (6.6)	
7. Управление стоимостью проекта		Стоимостная оценка 3.2.2.10 (7.1) Разработка бюджета расходов 3.2.2.11 (7.2)		Управление стоимостью 3.2.4.6 (7.3)	
8. Управление качеством проекта		Планирование качества 3.2.2.12 (8.1)	Процесс обеспечения качества 3.2.3.2 (8.2)	Процесс контроля качества 3.2.4.7 (8.3)	
9. Управление человеческими ресурсами проекта		Планирование человеческих ресурсов 3.2.2.13 (9.1)	Набор команды проекта 3.2.3.3 (9.2) Развитие команды проекта 3.2.3.4 (9.3)	Управление командой проекта 3.2.4.8 (9.4)	
10. Управление коммуникациями проекта		Планирование коммуникаций 3.2.2.14 (10.1)	Распространение информации 3.2.3.5 (10.2)	Отчетность по исполнению 3.2.4.9 (10.3) Управление участием проекта 3.2.4.10 (10.4)	
11. Управление рисками проекта		Планирование управления рисками 3.2.2.15 (11.1) Идентификация рисков 3.2.2.16 (11.2) Качественный анализ рисков 3.2.2.17 (11.3) Количественный анализ рисков 3.2.2.18 (11.4) Планирование реагирования на риски 3.2.2.19 (11.5)		Мониторинг и управление рисками 3.2.4.11 (11.6)	
12. Управление поставками проекта		Планирование покупок и приобретения 3.2.2.20 (12.1) Планирование контрактов 3.2.2.21 (12.2)	Запрос информации у поставщика 3.2.3.6 (12.3) Выбор поставщиков 3.2.3.7 (12.4)	Администрирование контрактов 3.2.4.12 (12.5)	Закрытие контракта 3.2.5.2 (12.6)

Формы контроля: своевременное представление выполненных заданий

**Тема 03.01.06 Методы организации работы в коллективах
разработчиков программного обеспечения
Выполнение практических заданий**

Цель работы: изучить вопросы проектирования программного обеспечения при структурном подходе к программированию. Стадия «Техническое задание»

Технический проект должен включать данные об объемах и интенсивности потоков обрабатываемой информации, количестве пользователей программной системы, характеристиках оборудования и программного обеспечения, взаимодействующего с проектируемым программным продуктом.

При разработке технического проекта оформляются:

- ведомость технического проекта. Общая информация по проекту;
- пояснительная записка к техническому проекту. Вводная информация, позволяющая ее потребителю быстро освоить данные по конкретному проекту;
- описание систем классификации и кодирования;
- перечень входных данных (документов). Перечень информации, которая используется как входящий поток и служит источником накопления;
- перечень выходных данных (документов). Перечень информации, которая используется для анализа накопленных данных;
- описание используемого программного обеспечения. Перечень программного обеспечения и СУБД, которые планируется использовать для создания информационной системы;
- описание используемых технических средств. Перечень аппаратных средств, на которых планируется работа проектируемого программного продукта;
- проектная оценка надежности системы. Экспертная оценка надежности с выявлением наиболее благополучных участков программной системы и ее узких мест;
- ведомость оборудования и материалов. Перечень оборудования и материалов, которые потребуются в ходе реализации проекта.

Структурная схема

Структурной называют схему, отражающую состав и взаимодействие по управлению частями разрабатываемого программного обеспечения. Структурная схема определяется архитектурой разрабатываемого ПО.

Функциональная схема

Функциональная схема — это схема взаимодействия компонентов программного обеспечения с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств.

Разработка алгоритмов

Метод пошаговой детализации реализует нисходящий подход к программированию и предполагает пошаговую разработку алгоритма.

Структурные карты

Методика структурных карт используется на этапе проектирования ПО для того, чтобы продемонстрировать, каким образом программный продукт выполняет системные требования. Структурные карты Константайна предназначены для описания отношений между модулями. Техника структурных карт Джексона основана на методе структурного программирования Джексона, который выявляет соответствие между структурой потоков данных и структурой программы. Основное внимание в методе сконцентрировано на соответствии входных и выходных потоков данных.

Порядок выполнения работы

1. На основе технического задания разработать уточненные алгоритмы программ, составляющих заданный программный модуль. Использовать метод пошаговой детализации.
2. На основе уточненных и доработанных алгоритмов разработать структурную схему программного продукта.
3. Разработать функциональную схему программного продукта.
4. Оформить результаты, используя MS Office или MS Visio в виде технического проекта.

Пример 1. Разработать техническое задание на программный продукт, предназначенный для наглядной демонстрации школьникам графиков функций одного аргумента $y = f(x)$. Разрабатываемая программа должна рассчитывать таблицу значений и строить график функций на заданном отрезке по заданной формуле и менять шаг аргумента и границы отрезка. Кроме этого, программа должна запоминать введенные формулы.

Формы контроля: своевременное представление выполненных заданий

Критерии оценки: - правильность составленного технического задания

Тема 03.01.07 Стандарты качества программного обеспечения

Выполнение практических заданий

Цель работы: научиться создавать формальные модели и на их основе определять спецификации разрабатываемого программного обеспечения при структурном подходе к программированию. Стадия «Эскизный проект»

Разработка программного обеспечения начинается с анализа требований к нему. В результате анализа получают спецификации разрабатываемого программного обеспечения, строят общую модель его взаимодействия с пользователем или другими программами и конкретизируют его основные функции.

При структурном подходе к программированию на этапе анализа и определения спецификаций разрабатывают три типа моделей: модели функций, модели данных и модели потоков данных. Поскольку разные модели описывают проектируемое программное обеспечение с разных сторон, рекомендуется использовать сразу несколько моделей, разрабатываемых в виде диаграмм, и пояснять их текстовыми описаниями, словарями и т. п.

Структурный анализ предполагает использование следующих видов моделей:

- диаграмм потоков данных (DFD — Data Flow Diagrams), описывающих взаимодействие источников и потребителей информации через процессы, которые должны быть реализованы в системе;
- диаграмм «сущность—связь» (ERD — Entity-Relationship Diagrams), описывающих базы данных разрабатываемой системы;
- диаграмм переходов состояний (STD — State Transition Diagrams), характеризующих поведение системы во времени;
- функциональных диаграмм (методика SADT);
- спецификаций процессов;
- словаря терминов.

Спецификации процессов

Спецификации процессов обычно представляют в виде краткого текстового описания, схем алгоритмов, псевдокодов, Ploш-форм или диаграмм Насси — Шнейдермана.

Словарь терминов

Словарь терминов представляет собой краткое описание основных понятий, используемых при составлении спецификаций. Он должен включать определение основных понятий предметной области, описание структур элементов данных, их типов и форматов, а также всех сокращений и условных обозначений.

Диаграммы переходов состояний

С помощью *диаграмм переходов состояний* можно моделировать последующее функционирование системы на основе ее предыдущего и текущего функционирования. Моделируемая система в любой заданный момент времени находится точно в одном из конечного множества состояний. С течением времени она может изменить свое состояние, при этом переходы между состояниями должны быть точно определены.

Функциональные диаграммы

Функциональные диаграммы отражают взаимосвязи функций разрабатываемого программного обеспечения.

Они создаются на ранних этапах проектирования систем, для того чтобы помочь проектировщику выявить основные функции и составные части

проектируемой системы и, по возможности, обнаружить и устранить существенные ошибки. Для создания функциональных диаграмм предлагается использовать методологию SADT.

Диаграммы потоков данных

Для описания потоков информации в системе применяются *диаграммы потоков данных* (DFD — Data flow diagrams). DFD позволяет описать требуемое поведение системы в виде совокупности процессов, взаимодействующих посредством связывающих их потоков данных. DFD показывает, как каждый из процессов преобразует свои входные потоки данных в выходные потоки данных и как процессы взаимодействуют между собой.

Диаграммы «сущность—связь»

Диаграмма сущность—связь — инструмент разработки моделей данных, обеспечивающий стандартный способ определения данных и отношений между ними. Она включает *сущности* и взаимосвязи, отражающие основные бизнес-правила предметной области. Такая диаграмма не слишком детализирована, в нее включаются основные *сущности* и *связи* между ними, которые удовлетворяют требованиям, предъявляемым к НС.

Порядок выполнения работы

1. На основе технического задания из лабораторной работы № 1 выполнить анализ функциональных и эксплуатационных требований к программному продукту.
2. Определить основные технические решения (выбор языка программирования, структура программного продукта, состав функций ПП, режимы функционирования) и занести результаты в документ, называемый «Эскизным проектом» (см. приложение 4).
3. Определить диаграммы потоков данных для решаемой задачи.
4. Определить диаграммы «сущность—связь», если программный продукт содержит базу данных.
5. Определить функциональные диаграммы.
6. Определить диаграммы переходов состояний.
7. Определить спецификации процессов.
8. Добавить словарь терминов.
9. Оформить результаты, используя MS Office или MS Visio в виде эскизного проекта.

Отчет по работе должен состоять из:

1. Постановки задачи.
2. Документа «Эскизный проект», содержащего:
 - выбор метода решения и языка программирования;
 - спецификации процессов;
 - все полученные диаграммы;
 - словарь терминов.

Пример 1. Разработать техническое задание на программный продукт, предназначенный для наглядной демонстрации школьникам графиков функций одного аргумента $y = f(x)$. Разрабатываемая программа должна рассчитывать таблицу значений и строить график функций на заданном отрезке по заданной формуле и менять шаг аргумента и границы отрезка. Кроме этого, программа должна запоминать введенные формулы.

Формы контроля: своевременное представление выполненных заданий

Критерии оценки: - правильность составленного технического задания

Тема 03.01.08 Методы и средства разработки программной документации

Выполнение практических заданий

Цель работы: Проектирование программного обеспечения при структурном подходе.

Сущность структурного подхода к разработке ИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимоувязаны. При разработке системы "снизу-вверх" от отдельных задач ко всей системе целостность теряется, возникают проблемы при информационной стыковке отдельных компонентов.

Все наиболее распространенные методологии структурного подхода базируются на ряде общих принципов. В качестве двух базовых принципов используются следующие:

- принцип "разделяй и властвуй" - принцип решения сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения;
- принцип иерархического упорядочивания - принцип организации составных частей проблемы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне.

На стадии проектирования ИС модели расширяются, уточняются и дополняются диаграммами, отражающими структуру программного обеспечения: архитектуру ПО, структурные схемы программ и диаграммы экранных форм.

Перечисленные модели в совокупности дают полное описание ИС независимо от того, является ли она существующей или вновь разрабатываемой. Состав диаграмм в каждом конкретном случае зависит от необходимой полноты описания системы.

Формы контроля: своевременное представление выполненного задания

МДК 03.02 Инструментальные средства разработки программного обеспечения

Тема 03.02.01 Концепция и реализация программных процессов

Реферирование

Цель задания: углубление знаний по темам занятий.

Текст задания. Составить реферат по темам:

- *Возможности программы BP Win*
- *Возможности программы MS Visio*

Реферат (от латинского *refere* - докладывать, сообщать) - краткое изложение содержания одного или нескольких источников, раскрывающее определенную тему. Хотя смысловое значение слова «реферат» переплетается со словом «доклад», реферат является более высокой формой творческой работы. Подготовка к реферату требует глубокого знания аспектов изучаемой проблемы и вопроса, умение обстоятельно их анализировать.

Подготовка реферата способствует всестороннему знакомству с литературой по избранной теме, создает возможность комплексного использования приобретенных навыков работы с книгой, развивает самостоятельность мышления, умение на научной основе анализировать и делать выводы. Материал в реферате излагается с позиции автора исходного текста.

Прежде всего надо знать *из чего состоит реферат*.

Компоненты содержания:

- титульный лист,
- план;
- введение (постановка проблемы, объяснение выбора темы, ее значения, актуальности, определение цели и задач реферата, краткая характеристика используемой литературы);
- основная часть (каждая проблема или части одной проблемы рассматриваются в отдельных разделах реферата и являются логическим продолжением друг друга);
- заключение;
- список литературы.

Титульный лист - лицо реферата. На титульном листе должно присутствовать: Сверху полное название учреждения, для которого

пишется реферат. Далее примерно в центре листа название темы реферата. Чуть пониже справа от темы, группа и Ф.И.О.(Фамилия имя отчество) того, кто пишет реферат, с указанием его статуса в учебном учреждении. На следующий строчке кто принимает его, тоже с указанием статуса. Внизу год создания реферата (можно еще и место, например, Магнитогорск, 2013)

План - второй лист реферата. Хорошо сделанный реферат имеет не только главы, но и подразделы, что указывается в содержании, требует наличие номеров страниц на каждую главу и подраздел реферата.

Введение - краткое описание темы и постановка вопросов. Во введении объясняется:

- почему выбрана такая тема, чем она важна (личное отношение к теме (проблеме), чем она актуальна (отношение современного общества к этой теме (проблеме), какую культурную или научную ценность представляет (с точки зрения исследователей, ученых);

- какая литература использована: исследования, научно-популярная литература, учебная, кто авторы... (Клише: «Материалом для написания реферата послужили ...»)

- из чего состоит реферат (введение, количество глав, заключение, приложения. Клише: «Во введении показана идея (цель) реферата. Глава 1 посвящена..., во 2 главе ... В заключении сформулированы основные выводы...»)

Основная часть реферата состоит из нескольких глав / разделов, постепенно раскрывающих тему. Каждый из разделов рассматривает какую-либо из сторон основной темы. Утверждения позиций подкрепляются доказательствами, взятыми из литературы (цитирование, указание цифр, фактов, определения)

Если доказательства заимствованы у автора используемой литературы - это оформляется как ссылка на источник и имеет порядковый номер.

Ссылки оформляются внизу текста под чертой, где указываются порядковый номер ссылки и данные книги или статьи. В конце каждого раздела основной части обязательно формулируется вывод. (Клише: «Таким образом,.. Можно сделать заключение, что... В итоге можно прийти к выводу...»)

В заключении (очень кратко) формулируются общие выводы по основной теме, перспективы развития исследования, собственный взгляд на решение проблемы и на позиции авторов используемой литературы, о воём согласии или несогласии с ними. Вывод реферата – показывает степень проработки темы.

Список литературы - список источников материалов, использованных при создании реферата. Должен содержать не меньше трех источников, составленных в алфавитном порядке.

Формы контроля: - представление реферата, защита реферата

Критерии оценки: логичность структуры содержания, полнота раскрытия проблемы, качество оформления

Доклад, сообщение

Цель заданий: углубление знаний по темам занятий.

Текст задания. 1. Составить доклад по темам реферата:

- *Возможности программы BP Win*

- *Возможности программы MS Visio*

Доклад - публичное сообщение на определенную тему, в процессе подготовки которого используются те или иные навыки исследовательской работы.

Компоненты содержания:

- план работы;
- систематизация сведений;
- выводы и обобщения.

Рекомендации по выполнению:

В докладе выделяются три основные части:

- 1) Вступительная часть, в которой определяется тема, структура и содержание, показывается, как она отражена в трудах ученых.
- 2) Основная часть содержит изложение изучаемой темы / вопроса / проблемы (желательно в проблемном плане).
- 3) Обобщающая – заключение, выводы.

Формы контроля: выступление на занятии

Критерии оценки: актуальность, глубина, научность теоретического материала; четкость выступления, уровень самостоятельности; использование мультимедийной презентации, ее качество; время выступления

Тема 03.02.02 Принципы построения программного обеспечения

Выполнение практических заданий

Цель заданий: углубление ранее изученного материала, выработка умений и навыков по применению изученных языков программирования,

составлению алгоритма типовых заданий, применение полученных знаний на практике.

Задания

Создание диаграммы вариантов использования

Создание диаграммы последовательностей

Создание диаграммы состояния

Вариант 1 Будильник

Требуется разработать модель программного обеспечения встроенного микропроцессора для будильника. На экране будильника постоянно отображается текущее время (часы и минуты, например: 12:00), двоеточие между числом часов и числом минут загорается и гаснет с интервалом в полсекунды. Управление будильником осуществляется следующими кнопками:

- кнопкой режима установки времени,
- кнопкой режима установки времени срабатывания,
- двумя отдельными кнопками для установки часов и минут,
- кнопкой сброса сигнала «СБРОС».

На будильнике имеется переключатель режима работы со следующими положениями: «ВЫКЛ», «ВКЛ», «РАДИО» и «ТАЙМЕР». Для установки текущего времени нужно нажать на кнопку режима установки и, при нажатой кнопке, нажимать на кнопки установки часов и минут. При каждом нажатии на кнопки, устанавливаемое значение увеличивается на одну единицу (один час или одну минуту соответственно). При достижении максимального значения производится сброс. Для установки времени срабатывания будильника нужно нажать на кнопку режима установки времени срабатывания и, держа кнопку нажатой, нажимать на кнопки установки часов и минут. Когда переключатель режима работы находится в положении «ВКЛ», при достижении времени срабатывания происходит подача звукового сигнала в течение одной минуты. Сигнал можно прервать, нажав на кнопку «СБРОС». При этом сигнал должен быть возобновлен через пять минут. При установке переключателя в положение «ВЫКЛ» звуковой сигнал не подается. Когда переключатель находится в положении «РАДИО» работает радиоприемник. При переводе переключателя в положение «ТАЙМЕР» включается радиоприемник на тридцать минут, а затем часы переходят в состояние будильника (аналогично положению «ВКЛ»). При нажатии на кнопку режима установки времени, будильник должен отображать время срабатывания.

Вариант 2 Генеалогическое дерево

Требуется разработать модель системы для поддержки

генеалогических деревьев. Система хранит сведения о персонах (Ф.И.О., пол, дата рождения, дата смерти, биография) и о родственных связях между ними. Связи бывают только трех видов: «мужья-жены», «дети-родители» и «братья- сестры». Система обеспечивает возможность добавления данных о новых персонах и родственных связях, изменение введенных данных и удаление ненужных данных. Система следит за непротиворечивостью вводимых данных. Например, недопустимо, чтобы человек был собственным предком или потомком. Разработанная модель должна содержать схему базы данных для хранения генеалогических деревьев. Пользователи системы могут осуществлять поиск полезной информации по дереву:

- находить для указанного члена семьи его детей;
- находить для указанного члена семьи его родителей;
- находить для указанной персоны братьев и сестер, если таковые есть;
- получать список всех предков персоны;
- получать список всех потомков персоны;
- получать список всех родственников персоны;
- проследживать цепочку родственных связей от одной персоны до другой (например, если Петр является шурином Ивана, то на запрос о родственных связях между Петром и Иваном выдается такой результат: «Петр - брат Ольги, Ольга - жена Ивана»).

Формы контроля: своевременное представление выполненного задания

Критерии оценки: - диаграммы uml.

Тема 03.02.03 Структура и приемы работы с инструментальными средствами, поддерживающими создание программного обеспечения

Творческое задание

Цель заданий: углубление ранее изученного материала, выработка умений и навыков по применению изученных языков программирования, составлению алгоритма типовых заданий, применение полученных знаний на практике.

Задания

Разработать алгоритм и написать программы в MS Visual Studio

Вариант 1: Написать программу, в окне которой на фоне картинки

отображается график изменения курса доллара. Фоновый рисунок, а также данные должны загружаться из файла.

Вариант 2: Написать программу, используя которую можно оценить способность игрока (испытуемого) запоминать числа. Программа должна выводить числа, а испытуемый - вводить эти числа с клавиатуры. Время, в течение которого игрок будет видеть число, ограничьте, например, одной секундой. По окончании теста программа должна вывести результат: количество показанных чисел и количество чисел, которые испытуемый запомнил правильно.

Вариант 3: Написать программу расчета стоимости поездки на автомобиле. Входные данные: цена бензина за один литр, расход бензина (литров на 100 км), расстояние (км). Предусмотреть корректность введенных данных.

Вариант 4: Написать программу о результатах соревнований по плаванию. В соревнованиях по плаванию приняли участие 25 спортсменов. Данные должны быть записаны в порядке ухудшения результата. После заполнения массива выяснилось, что значение первого элемента не соответствует требованию упорядоченности. Изменить массив так, чтобы данные были упорядочены.

Вариант 5: Составить программу следующей игры. Человек в ответ на появляющийся на экране запрос Чет (2) или нечет (1)? прогнозирует появление одного из двух случайных чисел: 1 или 2. После ввода ответа компьютер случайным образом генерирует одно из указанных чисел, которое выводится на экран, и определяется результат прогноза ("Верно" или "Неверно" или т. п.). Варианты программы:

а) "угадывание" проводится один раз;

б) "угадывание" проводится n раз. В результате игры определяется количество верных и неверных ответов;

в) "угадывание" проводится до тех пор, когда в ответ на запрос Продолжить еще раз? будет введено Нет. В результате игры определяется количество верных и неверных ответов.

Формы контроля: своевременное представление выполненных заданий

Критерии оценки: - правильность составления алгоритмов, написания кода программ; объем выполненных заданий, оформление

МДК 03.03 Документирование и сертификация
Тема 03.03.01. Составление программной документации.

Задание

Составить руководство пользователя для предложенной системы согласно требованиям ГОСТ

Цель задания: углубление ранее изученного материала, выработка умений и навыков по составлению программной документации, применение полученных знаний на практике.

Документ **«Руководство пользователя»** относится к пакету эксплуатационной документации. Основная цель руководства пользователя заключается в обеспечении пользователя необходимой информацией для самостоятельной работы с программой или автоматизированной системой.

Таким образом, документ **Руководство пользователя** должен отвечать на следующие вопросы: что это за программа, что она может, что необходимо для обеспечения ее корректного функционирования и что делать в случае отказа системы.

Руководящими стандартами для создания документа **Руководство пользователя** могут являться как РД 50-34.698-90 в п.п. 3.4. «Руководство пользователя», так и ГОСТ 19.505-79 «Руководство оператора. Требования к содержанию и оформлению». Ниже для сравнения приведены структуры документа согласно двум перечисленным стандартам.

РД 50-34.698-90 (п.п. 3.4 Руководство пользователя)	ГОСТ 19.505-79 Руководство оператора
Введение	
Область применения	
Описание возможностей	
Уровень подготовки пользователя	
Перечень эксплуатационной документации, с которыми необходимо ознакомиться пользователю	
Назначение и условия применения	
Виды деятельности и функции, для автоматизации которых предназначена программа	Назначение программы
Условия, при которых обеспечивается применение программы	Условия выполнения программы

Подготовка к работе	Выполнение программы
Состав и содержание дистрибутивного носителя данных	
Порядок загрузки данных и программ	Порядок загрузки, запуска и завершения программы
Проверка работоспособности	
Описание операций	Описание функций
Описание всех выполняемых функций, задач, комплексов задач, процедур	
Описание операций технологического процесса обработки данных, необходимых для выполнения функций, комплексов задач (задач), процедур	
Аварийные ситуации (действия в случае длительных отказов технических средств, обнаружении несанкционированного вмешательства в данные, действия по восстановлению программ или данных, и т.п.)	Сообщения оператору
Рекомендации по освоению	

Формы контроля: своевременное представление выполненного задания

Критерии оценки: - правильность написания руководства пользователя, оформление.

Тема 03.03.02. Документирование программных средств

Задание

Выполнение творческого задания по проектированию интерфейса программы

Цель задания: углубление ранее изученного материала, выработка умений и навыков по проектированию интерфейсов программных продуктов, применение полученных знаний на практике.

Пользовательский интерфейс — это совокупность информационной модели проблемной области, средств и способов взаимодействия пользователя с информационной моделью, а также компонентов, обеспечивающих формирование информационной модели в процессе работы программной системы.

Основные принципы разработки пользовательского интерфейса

Создание качественного интерфейса требует значительно большего, чем просто соблюдение некоторых инструкций. Оно предполагает реализацию принципа «интересы пользователя превыше всего» и соответствующую методологию разработки всего программного продукта. В англоязычной литературе для описания такого подхода используется термин **User-centered Design -UCD** («Разработка, ориентированная на пользователя»). Эта технология, кроме всего прочего, предполагает как можно более раннее Проектирование интерфейса с последующим его развитием в процессе разработки самого программного продукта.

Основное достоинство хорошего интерфейса пользователя заключается в том, что *пользователь всегда чувствует, что он управляет программным обеспечением, а не программное обеспечение управляет им.*

Для создания у пользователя такого ощущения «внутренней свободы» интерфейс должен обладать целым рядом свойств, рассмотренных ниже.

Естественность интерфейса

Естественный интерфейс — такой, который не вынуждает пользователя существенно изменять привычные для него способы решения задачи. Это, в частности, означает, что сообщения и результаты, выдаваемые приложением, не должны требовать дополнительных пояснений. Целесообразно также сохранить систему обозначений и терминологию, используемые в данной предметной области.

Использование знакомых пользователю понятий и образов (метафор) обеспечивает интуитивно понятный интерфейс при выполнении его заданий. Вместе с тем, используя метафоры, не надо ограничивать их машинную реализацию полной аналогией с одноименными объектами реального мира. Например, папка на Рабочем столе Windows может использоваться для хранения целого ряда других объектов. Метафоры являются своего рода «мостиком», связывающим образы реального мира с теми действиями и объектами, которыми приходится манипулировать пользователю при его работе на компьютере; они обеспечивают «узнавание», а не «вспоминание». Пользователи запоминают действие, связанное со знакомым объектом, более легко, чем они запомнили бы имя команды, связанной с этим действием.

Согласованность интерфейса

Согласованность позволяет пользователям переносить имеющиеся знания на новые задания, осваивать новые аспекты быстрее, и благодаря этому фокусировать внимание на решаемой задаче, а не тратить время на уяснение различий в использовании тех или иных элементов управления,

команд и т.д. Обеспечивая преемственность полученных ранее знаний и навыков, согласованность делает интерфейс узнаваемым и предсказуемым.

Согласованность важна для всех аспектов интерфейса, включая имена команд, визуальное представление информации и поведение интерактивных элементов. Для реализации свойства согласованности в создаваемом программном обеспечении, необходимо учитывать его различные аспекты.

Согласованность в пределах продукта

Одна и та же команда должна выполнять одни и те же функции, где бы она ни встретилась, причем одним и тем же образом. Например, если в одном диалоговом окне команда *Копировать* означает немедленное выполнение соответствующих действий, то в другом окне она не должна требовать от пользователя дополнительно указать расположение копируемой информации. Надо использовать одну и ту же команду, чтобы выполнить функции, которые кажутся подобными пользователю.

Согласованность в пределах рабочей среды

Поддерживая согласованность с интерфейсом, предоставляемым операционной системой, приложение может «опираться» на те знания и навыки пользователя, которые он получил ранее при работе с другими приложениями.

Согласованность в использовании метафор

Если поведение некоторого программного объекта выходит за рамки того, что обычно подразумевается под соответствующей ему метафорой, у пользователя могут возникнуть трудности при работе с таким объектом. Например, если для программного объекта *Корзина* определить операцию *Запуск*, то для уяснения ее смысла пользователю, скорее всего, потребуются посторонняя помощь.

Дружественность интерфейса (принцип «прощения» пользователя)

Пользователи обычно изучают особенности работы с новым программным продуктом методом проб и ошибок. Эффективный интерфейс должен принимать во внимание такой подход. На каждом этапе работы он должен разрешать только соответствующий набор действий и предупреждать пользователей о тех ситуациях, где они могут повредить системе или данным; еще лучше, если у пользователя существует возможность отменить или исправить выполненные действия.

Даже при наличии хорошо спроектированного интерфейса пользователи могут делать те или иные ошибки. Эти ошибки могут быть как «физического» типа (случайный выбор неправильной команды или данных) так и «логического» (принятие неправильного решения на выбор

команды или данных). Эффективный интерфейс должен позволять предотвращать ситуации, которые, вероятно закончатся ошибками. Он также должен уметь **адаптироваться** к потенциальным ошибкам пользователя и облегчать ему процесс устранения последствий таких ошибок.

Принцип «обратной связи»

Всегда обеспечивайте обратную связь для действий пользователя. Каждое действие пользователя должно получать визуальное, а иногда и звуковое подтверждение того, что программное обеспечение восприняло введенную команду; при этом вид реакции, по возможности, должен учитывать природу выполненного действия.

Обратная связь эффективна в том случае, если она реализуется своевременно, т.е. как можно ближе к точке последнего взаимодействия пользователя с системой. Когда компьютер обрабатывает поступившее задание, полезно предоставить пользователю информацию относительно состояния процесса, а также возможность прервать этот процесс в случае необходимости. Ничто так не смущает не очень опытного пользователя, как заблокированный экран, который никак не реагирует на его действия. Типичный пользователь способен вытерпеть только несколько секунд ожидания ответной реакции от своего электронного «собеседника».

Простота интерфейса

Интерфейс должен быть простым. При этом имеется в виду не упрощенчество, а обеспечение **легкости** в его изучении и в использовании. Кроме того, он должен предоставлять **доступ** ко всему перечню функциональных возможностей, предусмотренных данным приложением. Реализация доступа к широким функциональным возможностям и обеспечение простоты работы противоречат друг другу. Разработка эффективного интерфейса призвана сбалансировать эти цели.

Один из возможных путей поддержания простоты — представление на экране информации, минимально необходимой для выполнения пользователем очередного шага задания. Многословные командные имена или сообщения, непродуманные или избыточные фразы затрудняют пользователю извлечение существенной информации.

Другой путь к созданию простого, но эффективного интерфейса — размещение и представление элементов на экране с учетом их смыслового значения и логической взаимосвязи. Это позволяет использовать в процессе работы ассоциативное мышление пользователя.

Можно помочь пользователям управлять сложностью отображаемой информации, используя *последовательное раскрытие* (диалоговых окон, разделов меню и т.д.). Последовательное раскрытие предполагает такую организацию информации, при которой в каждый момент времени на экране находится только та ее часть, которая

необходима для выполнения очередного шага. Сокращая объем информации, представленной пользователю, уменьшают объем информации, подлежащей обработке. Примером такой организации является иерархическое (каскадное) меню, каждый уровень которого отображает только те пункты, которые соответствуют одному, выбранному пользователем, пункту более высокого уровня.

Гибкость интерфейса

Гибкость интерфейса — это его способность учитывать уровень подготовки и производительность труда пользователя. Гибкость предполагает возможность изменения структуры диалога и/или входных данных. Концепция гибкого (*адаптивного*) интерфейса в настоящее время является одной из основных областей исследования взаимодействия человека и ЭВМ. Основная проблема состоит не в том, как организовать изменения в диалоге, а в том, какие признаки нужно использовать для определения необходимости внесения изменений и их сути.

Эстетическая привлекательность

Проектирование визуальных компонентов является важнейшей составной частью разработки программного интерфейса. Корректное визуальное представление используемых объектов обеспечивает передачу весьма важной дополнительной информации о поведении и взаимодействии различных объектов. В то же время следует помнить, что каждый визуальный элемент, который появляется на экране, потенциально требует внимания пользователя, которое, как известно, не безгранично. Следует формировать на экране среду, которая не только содействовала бы пониманию пользователем представленной информации, но и позволяла бы сосредоточиться на наиболее важных ее аспектах.

Наибольших успехов в проектировании пользовательского интерфейса, обладающего перечисленными свойствами, к настоящему времени добились разработчики компьютерных игр.

Качество интерфейса сложно оценить количественными характеристиками, однако более или менее объективную его оценку можно получить на основе приведенных ниже частных показателей.

1. Время, необходимое определенному пользователю для достижения заданного уровня знаний и навыков по работе с приложением (например, непрофессиональный пользователь должен освоить команды работы с файлами не более чем за 4 часа).

2. Сохранение полученных рабочих **навыков** по истечении некоторого времени (например, после недельного перерыва пользователь должен выполнить определенную последовательность операций за заданное время).

3. Скорость решения задачи с помощью данного приложения; при этом должно оцениваться не быстроедействие системы и не скорость ввода данных с клавиатуры, а время, необходимое для достижения цели решаемой задачи. Исходя из этого, критерий оценки по данному показателю может быть сформулирован, например, так: пользователь должен обработать за час не менее 20 документов с ошибкой не более 1%.

4. Субъективная удовлетворенность пользователя при работе с системой (которая количественно может быть выражена в процентах или оценкой по n-бальной шкале).

Обобщая изложенное выше, можно кратко сформулировать те основные правила, соблюдение которых позволяет рассчитывать на создание эффективного пользовательского интерфейса.

- Интерфейс пользователя необходимо проектировать и разрабатывать как отдельный компонент создаваемого приложения.

- Необходимо учитывать возможности и особенности аппаратно-программных средств, на базе которых реализуется интерфейс.

- Целесообразно учитывать особенности и традиции той предметной области, к которой относится создаваемое приложение.

- Процесс разработки интерфейса должен носить итерационный характер, его обязательным элементом должно быть согласование полученных результатов с потенциальным пользователем.

- Средства и методы реализации интерфейса должны обеспечивать возможность его адаптации к потребностям и характеристикам пользователя.

Формы контроля: своевременное представление выполненного задания

Критерии оценки: - правильно созданный макет интерфейса программного продукта с учетом его направленности.

Тема 03.03.03. Схемы алгоритмов, данных и систем

Задание

Выполнение творческого задания «Построение схем алгоритмов, данных и систем» индивидуального задания

Цель задания: углубление ранее изученного материала, выработка умений и навыков по построению схем алгоритмов, данных и систем программных продуктов, применение полученных знаний на практике.

Ознакомиться с единой системой программной документации «СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ. Обозначения условные и правила выполнения» (ГОСТ 19.701-90). В соответствии с ГОСТ 19.701-90 построить схемы алгоритмов, данных и систем» индивидуального задания.

Формы контроля: своевременное представление выполненного задания

Критерии оценки: - правильно созданные схемы алгоритмов, данных и систем программного продукта с учетом его направленности.

Тема 03.03.04. Сертификация программного обеспечения

Задание

Подготовить презентацию к семинару на тему: «Сертификация программного обеспечения»

Цель задания: углубление ранее изученного материала.

Семинарское занятие (семинар) - одна из основных форм организации учебного процесса, представляющая собой коллективное обсуждение студентами теоретических вопросов под руководством преподавателя, в процессе которого решаются задачи познавательного и воспитательного характера. Семинарские занятия в рамках конкретных разделов учебных дисциплин предусмотрены, прежде всего, для глубокой проработки теоретического материала. Семинары формируют у студентов навыки свободного ведения дискуссии, формируют первичные навыки научной работы, стимулируют интерес к самостоятельному поиску новых идей и фактов. В ходе занятий студенты должны проявить умения и навыки анализа документов и справочно-информационной литературы

Порядок проведения семинарского занятия предполагает следующую последовательность:

1. выступление с докладом;
2. вопросы к выступающему;
3. развернутая беседа по теме доклада;
4. обсуждение достоинств и недостатков доклада;
5. заключительное слово докладчика;
6. заключение преподавателя.

Семинарские занятия строятся на системе докладов (сообщений), которые готовятся студентами по заранее выбранной ими теме. Примерные темы докладов приведены в каждой рабочей программе дисциплины. В то же время студент может, по согласованию с преподавателем, самостоятельно сформулировать тему для своего выступления. При подготовке доклада (сообщения) к семинарскому занятию целью является проведение сравнительного анализа проблемных ситуаций, а также способы и специфика решения перечисленных проблем, где основная задача для студента – самостоятельный анализ. Перечень требований к выступлению студента на семинаре: связь выступления с предшествующей темой или вопросом; раскрытие

сущности проблемы; методологическое значение для научной, профессиональной и практической деятельности. Важнейшие требования к выступлениям студентов - самостоятельность в подборе фактического материала и аналитическом отношении к нему, умение рассматривать примеры и факты во взаимосвязи и взаимообусловленности, отбирать наиболее существенные из них. Приводимые участником семинара примеры и факты должны быть существенными, по возможности перекликаться с профилем обучения. 6 Выступление студента должно соответствовать требованиям логики. Четкое вычленение излагаемой проблемы, ее точная формулировка, неукоснительная последовательность аргументации именно данной проблемы, без неоправданных отступлений от нее в процессе обоснования, безусловная доказательность, непротиворечивость и полнота аргументации, правильное и содержательное использование понятий и терминов. При оценке докладов используются наиболее существенные критерии: глубина и самостоятельность проведенного студентом анализа проблемных ситуаций отечественной и зарубежной управленческой практики. В то же время, нельзя не отметить и некоторые формальные требования, которые предъявляются к докладам. Доклад должен иметь четкую структуру. Именно с целью выявления этой структуры всякое выступление на семинаре должно начинаться с изложения его плана. Изложение выводов, полученных студентом в результате анализа, должно соответствовать требованиям логики, быть последовательным и доказательным, аргументация – полной и непротиворечивой. Выступление должно исключать неоправданные отступления от темы. Завершением выступления студентов на семинаре являются ответы на вопросы, которые группа готовит по предложенным учебно-методическим комплексом вопросам семинара. Вопросно-ответный прием публичных выступлений входит в критерий оценки выступлений студентов. Важным критерием при оценке выступления студента на семинарском занятии является соблюдение регламента. Продолжительность выступления не может превышать 10-15 минут. Необходимо также помнить о грамотности, стилистической выдержанности речи выступающего, корректности использования специальных терминов. Использование бытовой лексики и слов-паразитов, тавтология, стилистические и грамматические ошибки, а также чтение текста доклада с листа снижают впечатление от выступления и не могут не сказаться на его оценке. Доклад на семинарском занятии может сопровождаться компьютерной презентацией. Содержание презентации должно включать название темы доклада, его план, реферативное изложение его содержания, основные

выводы. Презентация может включать демонстрацию схем, таблиц и др. слайдов, служащих для иллюстрации тех или иных положений доклада.

Формы контроля: своевременное представление выполненного задания

Критерии оценки: - презентация к семинару на тему: «Сертификация программного обеспечения».

Тема 03.03.05. Государственная регистрация программы для ЭВМ и базы данных

Задания

Изучить пакет документов для государственной регистрации базы данных

Цель задания: получить и закрепить навыки и знания в области государственной регистрации базы данных, углубление ранее изученного материала.

Для оформления государственной регистрации базы данных на сайте Федерального института промышленной собственности в разделе Подача заявки/Подача заявки на регистрацию программы для ЭВМ или базы данных в колонке Пользовательская и техническая документация: **приведено Руководство пользователя** с поэтапным указанием действий при регистрации программы для ЭВМ или базы данных (<http://new.fips.ru/podacha-zayavki/podacha-zayavki-na-registratsiyu-programmy-dlya-evm-ili-bazy-dannykh/GuidePrEVM%2020170925.pdf>).

Формы контроля: своевременное представление выполненного задания

Критерии оценки: - пакет документов для государственной регистрации базы данных.