

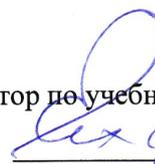
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ  
ИНКЛЮЗИВНОГО ВЫСШЕГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО ЭКОНОМИЧЕСКИЙ  
УНИВЕРСИТЕТ»

---

КАФЕДРА ЦИФРОВЫХ ТЕХНОЛОГИЙ

УТВЕРЖДАЮ

Проректор по учебно-методической работе



Е.С. Сахарчук

«27» 04 2022 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**  
Средства проектирование интерфейсов

образовательная программа направления подготовки 09.03.01 «Информатика и  
вычислительная техника»  
шифр, наименование

**Направленность (профиль)**

Программное обеспечение вычислительной техники и информационных систем

Квалификация (степень) выпускника: бакалавр

Форма обучения очная

Курс 3 семестр 6

Москва 2022

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования направления (специальности) 09.03.01 «Информатика и вычислительная техника», утвержденного приказом Министерства образования и науки Российской Федерации № 929 от «19» сентября 2017 г. Зарегистрировано в Минюсте России «10» октября 2017 г. № 48489

Разработчики рабочей программы:

МГГЭУ, доцент кафедры цифровых технологий

место работы, занимаемая должность

 Перепелкина Ю.В. 14.03 2022 г.  
Подпись Ф.И.О. Дата

Рабочая программа утверждена на заседании кафедры цифровых технологий  
(протокол № 4 от «21» 03 2022 г.)

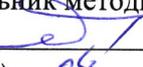
на заседании Учебно-методического совета МГГЭУ  
(протокол № 1 от «27» 04 2022 г.)

СОГЛАСОВАНО:

Начальник учебно-методического управления

 И.Г. Дмитриева  
«27» 04 2022 г.

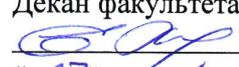
Начальник методического отдела

 Д.Е. Гапеенок  
«27» 04 2022 г.

Заведующий библиотекой

 В.А. Ахтырская  
«27» 04 2022 г.

Декан факультета ПМиИ

 Е.В. Петрунина  
«27» 04 2022 г.

## Содержание

- **ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ**
- **СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)**
- **ОСОБЕННОСТИ ОБУЧЕНИЯ ИНВАЛИДОВ И ЛИЦ С ОВЗ**
- **УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**
- **ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ**
- **ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**
- **УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)**
- **МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)**

## 1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ

### 1.1. Цели и задачи освоения учебной дисциплины (модуля)

**Цель** освоения дисциплины: формирование теоретических знаний и умений представления об основах анализа, выбора и эксплуатации, а также разработки протоколов и интерфейсов информационных систем, формирование практических навыков использования графических библиотек для организации человеко-машинного взаимодействия оконного интерфейса.

#### **Задачи:**

- изучить основы технического проектирования интерфейса;
- создать информационную систему с развитым интерфейсом пользователя;
- научиться использовать графические библиотеки для организации человеко-машинного взаимодействия.

### 1.2. Место дисциплины (модуля) в структуре образовательной программы направления подготовки 09.03.03 «Прикладная информатика» (бакалавриат).

Учебная дисциплина «Средства проектирования интерфейсов» относится к части, формируемой участниками образовательных отношений блока «Дисциплин (модулей)» Б1. Изучение учебной дисциплины базируется на знаниях, умениях и навыках, полученных обучающимися при изучении предшествующих курсов: «Информационные технологии обработки данных», «Объектно-ориентированное проектирование», «Операционные и телекоммуникационные системы». Изучение учебной дисциплины «Средства проектирования интерфейсов» необходимо для освоения дисциплин учебного плана «Архитектура корпоративных информационных систем», «Интеллектуальные системы», «Управление развитием информационных систем».

### 1.3. Требования к результатам освоения учебной дисциплины (модуля)

Процесс освоения учебной дисциплины направлен на формирование у обучающихся следующих компетенций:

Код компетенции	Содержание компетенции	Индикаторы достижения компетенции
ПК-3	Способен разрабатывать графический дизайн интерфейса, проектировать пользовательские	ПК-3.1. Знает основные понятия растровой и векторной графики; виды графических устройств; представление различных графических структур данных; основные алгоритмы формирования изображений ПК-3.2. Умеет использовать графические примитивы в языках программирования;

	интерфейсы по готовому образцу или концепции интерфейса	самостоятельно разрабатывать программы для решения задач обработки графической информации; решать прикладные задачи с помощью систем компьютерной графики. ПК-3.3. Владеет теоретическими основами компьютерной и инженерной графики; знаниями об областях применения; о системах компьютерной и инженерной графики; об основах человеко-машинного взаимодействия; об основных методах компьютерной и инженерной графики; интерактивной графики
--	---	--

## 2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)

### 2.1. Объем учебной дисциплины (модуля).

Объем дисциплины «Средства проектирования интерфейсов» составляет 4 зачетных

Вид учебной работы	Всего, часов	Очная форма
		Курс, часов
	Очная форма	1 курс
<b>Аудиторная работа обучающихся с преподавателем (по видам учебных занятий), всего в том числе:</b>	<b>144</b>	<b>144</b>
<b>Лекции</b>	<b>20</b>	<b>20</b>
В том числе, практическая подготовка (ЛПП)		
<b>Практические занятия</b>	<b>44</b>	<b>44</b>
В том числе, практическая подготовка (ПЗПП)		
<b>Лабораторные занятия</b>		
В том числе, практическая подготовка (ЛРПП)		
<b>Самостоятельная работа обучающихся</b>	<b>44</b>	<b>44</b>
В том числе, практическая подготовка (СРПП)		
<b>Промежуточная аттестация (подготовка и сдача), всего:</b>	<b>36</b>	<b>36</b>
Контрольная работа		
Курсовая работа		
Зачет		
Экзамен	<b>2</b>	<b>2</b>
Итого: Общая трудоемкость учебной дисциплины (в часах, зачетных единицах)	144 часов (4 з.е.)	144 часов (4 з.е.)

единицы/144 часа:

## 2.2. Содержание разделов учебной дисциплины (модуля)

№ п/п	Наименование раздела (темы)	Содержание раздела (тематика занятий)	Формируемые компетенции (индекс)
1	Тема 1. Человеко-машинное взаимодействие..	Введение в предметную область человеко-машинного интерфейса. Понятие интерфейса. Виды интерфейсов. Субъекты и объекты интерфейса. История человеко-машинного интерфейса	ПК-3
2	Тема 2. Межпрограммное взаимодействие	История межпрограммного интерфейса. Обоснование необходимости межпрограммного обмена данными. Форматы данных при межпрограммном обмене. Способы организации межпрограммного интерфейса. Организация обмена данными через общие файлы. Организация полнодуплексного обмена данными через общую (разделяемую память). Организация полудуплексного обмена данными через очередь сообщений. Введение понятий клиента и сервера как ролей программ при межпрограммном обмене данными. Поддержка межпрограммного обмена со стороны операционной системы: именованные блоки памяти, сокеты, мьютексы и семафоры. Проектирование клиентской и серверной части. Понятие протокола обмена данными. Форматы данных. Вопросы безопасности.	ПК-3
3	Тема 3. Оконный интерфейс.	Понятие окна. Понятие графического примитива. Понятие оконного примитива. Поддержка диалоговых примитивов со стороны операционной системы и графических библиотек. Понятие «родного интерфейса». Понятие модели и представления. Примитивы, работающие по схеме модель-представление (например, таблицы). Понятие модельного индекса и привязка данных. Вложенность окон. Автоматического расположение элементов. Элемент управления типа «вкладка». Стековые (накладываемые) элементы управления. Создание собственных элементов управления.	ПК-3

4	Тема 4. Модели данных.	Понятие внешнего источника данных. Базы данных как реляционный внешний источник. Язык SQL как унифицированный язык для организации обмена данными с базой. Подключение к базе данных. Переход от реляционной модели к объектной. Выбор формата представления данных (XML, JSON). Проектирование объектного доступа. Вопросы блокировок доступа при организации многопоточной работы. Классический метод подключения к сетевой БД. Выделение ядра системы. Выбор способа сетевого взаимодействия. Проблема многопоточного доступа к данным. Вопросы безопасности.	ПК-3
---	------------------------	--	------

### 2.3. Разделы дисциплины и виды занятий

#### Очная форма обучения

№ п/п	Наименование раздела (темы)	Аудиторная работа		Внеауд. работа	Объем в часах
		Л	ПЗ/ЛР	СР	Всего
		в том числе, ЛПП	в том числе, ПЗПП/ЛРПП	в том числе, СРПП	в том числе, ПП
<b>2 семестр</b>					
1.	Тема 1. Человеко-машинное взаимодействие..	5	10	10	20
2.	Тема 2. Межпрограммное взаимодействие	5	11	11	21
3.	Тема 3. Оконный интерфейс.	5	11	11	21
4.	Тема 4. Модели данных.	5	12	12	21
	Экзамен	<b>2</b>			
	<i>Итого:</i>	20	44	44	144
	<i>В том числе ПП:</i>				

### 2.4. План самостоятельной работы обучающегося по дисциплине (модулю)

#### Очная форма обучения

№	Название разделов и тем	Виды	Трудоем	Формируемые	Формы
---	-------------------------	------	---------	-------------	-------

		самостоятельной работы	кость	компетенции	контроля
1	Тема 1. Человеко-машинное взаимодействие..	Работа с источниками	10	ПК-3	Устный опрос
2	Тема 2. Межпрограммное взаимодействие	Работа с источниками	11	ПК-3	Устный опрос, практическое задание
3	Тема 3. Оконный интерфейс.	Работа с источниками	11	ПК-3	Устный опрос, практическое задание
4	Тема 4. Модели данных.	Работа с источниками	12	ПК-3	Устный опрос, практическое задание

### 3. ОСОБЕННОСТИ ОБУЧЕНИЯ ИНВАЛИДОВ И ЛИЦ С ОВЗ

Обучение лиц с ограниченными возможностями здоровья осуществляется с учетом индивидуальных психофизических особенностей, а для инвалидов также в соответствии с индивидуальной программой реабилитации инвалида.

Для получения обучающимися, имеющими ограниченные физические возможности, качественного образования должны выполняться следующие важные условия: обучающийся должен иметь возможность беспрепятственно посещать образовательное учреждение и использовать в своём обучении дистанционные образовательные технологии.

Для обучения и контроля обучающихся с нарушениями координации движений предусмотрено проведение тестирования с использованием компьютера.

Во время аудиторных занятий обязательно использование средств обеспечения наглядности учебного материала с помощью мультимедийного проектора. Скорость изложения материала должна учитывать ограниченные физические возможности студентов.

### 4. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

**Учебно-методическое обеспечение дисциплины для организации самостоятельной работы студентов** (содержит перечень основной литературы, дополнительной литературы, программного обеспечения и Интернет-ресурсы).

В распоряжении преподавателей и обучающихся имеется основное необходимое материально-техническое оборудование, Интернет-ресурсы, доступ к полнотекстовым

электронным базам, книжный фонд библиотеки Московского государственного гуманитарно-экономического университета.

## **5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ**

Интерактивные образовательные технологии, используемые в аудиторных занятиях самостоятельной работе обучающихся не предусмотрены.

## **6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

### **6.1. Организация входного, текущего и промежуточного контроля обучения**

Входное тестирование – не предусмотрено

Текущий контроль – опрос, тестирование.

Промежуточная аттестация – Экзамен

### **6.2. Тематика рефератов, проектов, творческих заданий, эссе и т.п.**

### **6.3. Курсовая работа**

Не предусмотрено.

### **6.4. Примеры тестовых заданий**

1. Вид диалогового взаимодействия пользователя с ПК, при котором используются различные команды, набираемые на клавиатуре и отображаемые на экране –

- a) Командный интерфейс
- b) Консольный интерфейс
- c) Интерфейс программной строки
- d) Программный интерфейс

2. Мера того, насколько хорошо диалог соответствует различным уровням подготовки и производительности труда пользователя

- a) Последовательный диалог
- b) Диалог, управляемый пользователем
- c) Гибкость
- d) Поддержка пользователя

3. Мера помощи, которую диалог оказывает пользователю при его работе с системой

- a) Последовательный диалог
- b) Диалог, управляемый пользователем
- c) Гибкость
- d) Поддержка пользователя

4. Критерии хорошего диалога (уберите лишнее)

- a) Естественность.
- b) Последовательность.

- c) Точность.
- d) Поддержка пользователя

5. Системы качества – совокупность:

¢ организационной структуры,  
¢ ответственности,  
¢ процедур,  
¢ процессов,  
¢ ресурсов, обеспечивающая осуществление руководства качеством продукции или процессов

6. На процессы разработки и оценки качества ПС оказывают влияние следующие обобщенные показатели ПС:

- назначение ПС;
- Область применения и
- Тип решаемых функциональных задач;
- Объем и сложность ПС;
- Необходимый состав и требуемые значения характеристик качества ПС и величина допустимого ущерба из-за недостаточного их качества;
- Степень связи решаемых задач с реальным масштабом времени или допустимой длительностью ожидания результатов решения задачи;
- Прогнозируемые значения длительности эксплуатации и перспектива создания множества версий ПС;
- Предполагаемый тираж производства и применения ПС;
- Степень необходимой документированности ПС.

7. ПО как продукт имеет отличия от других промышленных продуктов:

¢ наращивание объемов выпуска какого-то вида программного продукта происходит практически мгновенно и имеет низкую стоимость, так как производство следующей единицы программного продукта связано только с копированием информации на носитель (компакт-диск, флеш-накопитель или жесткий диск);

¢ большие ресурсы затрачиваются на стадии планирования, реализации и тестирования;

¢ сильное влияние человеческого фактора на производство программного продукта, так как производство программного продукта – интеллектуальная и творческая деятельность;

¢ в жизненном цикле программного продукта, как правило, отсутствует этап утилизации;

¢ программный продукт не подвержен физическому старению, а только моральному.

8. Мероприятия, обеспечивающие приемлемый уровень качества ПС  
Административные

1. Проведение обучения персонала, переподготовки.
2. Тщательное документирование всех изменений в структуре программного средства. Для этого используются средства поддержки версионности.

3. Назначение ответственных лиц за каждую доработку программного средства.
4. Уделение внимания текущему контролю качества и заключительному контролю качества.
5. Обеспечение мониторинга качества, например, фиксирование ошибок, поступивших от пользователя программного средства. Использование систематических испытательных методов, где испытания будут разработаны параллельно с разработкой программы.

9. Мероприятия, обеспечивающие приемлемый уровень качества ПС

Административные

- ¢ Введение внутренних стандартов. Такие стандарты обычно содержат соглашения о именовании переменных в программном коде, именовании файлов данных, процедур и функций.
- ¢ Организация отдела тестирования как самостоятельного подразделения.
- ¢ Проведение совместных аттестаций с пользователем.
- ¢ Обращение внимания на уровень и простоту обслуживаемости программного обеспечения.

10. Мероприятия, обеспечивающие приемлемый уровень качества ПС  
Технологические

- Выбор стандарта качества и четкое следование ему на всех этапах. Создание модели проекта с регулярными проверками, которые будут выполняться независимыми командами экспертизы. Такая модель может быть построена, например, на основе стандартов качества (например, ISO 9000).
- Единая среда разработки. Лучшие результаты дают программные продукты разработки, которые поддерживают несколько или все этапы жизненного цикла программного обеспечения. На данный момент такими комплексными решениями являются, например, продукты Oracle Designer, продукты фирмы Rational.
- Использовать формальный язык спецификаций (например, UML, DESIGN IDEF).
- Выбор надежной СУБД (если программное средство работает с массивами информации и использование СУБД оправдано).
- Тщательное тестирование программного обеспечения.

11. Мероприятия, обеспечивающие приемлемый уровень качества ПС  
Технологические

5. Широкое внедрение автоматизации тестирования.
  6. Использование полностью проверенной программной среды окружения и языка программирования, которые минимизируют опасность внесения ошибки.
- Использование статистических методов для сбора информации о качестве ПС.

- Изучение результатов испытаний (тестов) и ошибок для использования в постоянном усовершенствовании программы. Источник в случае возникновения отказа должен быть найден и устранен. Недостаточно найти ошибку в программном обеспечении и исправить ее. Изменения должны быть сделаны в процессе разработки ПО.
- Использование испытательной среды, которая предостережет от передачи пользователю ненадежного программного обеспечения. Создание автоматических средств приемки.

12. Для различных стадий ЖЦ стандартами определены следующие представления о качестве ПС:

- Целевое качество (ЦК) – необходимое и достаточное качество, отражающее реальные потребности заказчика или пользователя; ЦК не может быть полностью определено в начале проектирования ПС, поскольку заказчик не всегда может его четко определить, однако разработчики должны стремиться к достижению ЦК.
- Требуемое качество продукта (ТКП) – значения характеристик, фактически установленные в спецификации требований к качеству; ТКП используется как цель для начального утверждения в спецификации; должны фиксироваться оптимальные и допустимые минимальные требования.

1. Качество проекта (КП) – характеристики, представленные в основных компонентах проекта ПС (архитектуре, структуре программ, проектировании пользовательских интерфейсов); КП отражает концепцию и стратегию проекта.

13. Для различных стадий ЖЦ стандартами определены следующие представления о качестве ПС:

2. Оценочное (или прогнозируемое) качество продукта (ОКП) – оцененное или Международный стандарт ISO 9126 Общее представление о качестве ПС рекомендуется представлять тремя взаимодействующими и взаимозависимыми метриками характеристик качества, отражающими:
  3. внутреннее качество, проявляющееся в процессе разработки и других промежуточных этапов жизненного цикла ПС;
  4. внешнее качество, заданное требованиями заказчика в спецификациях и отражающееся характеристиками конечного продукта;
  5. качество при использовании в процессе нормальной эксплуатации и результативностью достижения потребностей пользователей с учетом затрат ресурсов.

13. Задачи и функции потребителей ПС:

6. для заказчика требуется полное соответствие характеристик программного продукта условиям контракта, технического задания и спецификаций требований;
7. для конечного оперативного пользователя ПС по основному назначению, качество в использовании обуславливают, в основном, характеристики функциональных возможностей, надежности, практичности и эффективности;
8. для персонала сопровождения ПС качество в использовании определяется преимущественно сопровождаемостью;

9. для персонала, выполняющего перенос ПС на иные платформы, а также инсталляцию и адаптацию к среде применения, качество в использовании определяется, прежде всего, мобильностью.
14. Основные факторы, определяющие качество программных средств
  10. назначение, содержание и описание функциональных характеристик, субхарактеристик и атрибутов, определяющих специфические особенности целей, задач, свойств и сферы применения конкретного программного средства – его функциональную пригодность;
  11. конструктивные характеристики качества, способствующие улучшению и совершенствованию назначения, функций и возможностей применения ПС;
  12. метрики, меры и шкалы, выбранных и пригодных для измерения и оценивания конкретных характеристик и атрибутов качества ПС с учетом определенной достоверности;
15. Основные факторы, определяющие качество программных средств
  13. уровни возможной детализации при описании и оценивании определенных характеристик и атрибутов качества ПС;
  14. цели и особенности потребителей результатов оценивания характеристик качества ПС;
  15. внешние и внутренние, негативные факторы, влияющие на достигаемое качество создания и применения ПС;
  16. доступные ресурсы, ограничивающие возможные величины реальных характеристик качества ПС;
  17. конкурентоспособность, выраженная отношением эффективности применения к стоимости приобретения и эксплуатации ПС.
16. Уровни детализации показателей
  18. категорийные-описательные, отражающие набор категорийные-описательные свойств и общие характеристики объекта – его функции, категории ответственности, защищенности и важности, которые могут быть представлены номинальной шкалой категорий-свойств;
  19. количественные, представляемые множеством количественных упорядоченных числовых точек, отражающих непрерывные или дискретные закономерности и описываемые интервальной или относительной шкалой, которые можно объективно измерить и численно сопоставить с требованиями;
  20. качественные – содержащие несколько упорядоченных или отдельных свойств – категорий, которые характеризуются порядковой или точечной шкалой набора категорий (есть – нет, хорошо – плохо), устанавливаются, выбираются и оцениваются в значительной степени субъективно и экспертно.
17. В международном стандарте ISO 9126: 1991, для формализации показателей качества ПС, при отборе минимума стандартизируемых показателей выдвигались и учитывались следующие принципы:
  21. ясность и измеряемость значений;
  22. отсутствие перекрытия между используемыми показателями;
  23. соответствие установившимся понятиям и терминологии;

24. возможность последующего уточнения и детализации. Так же выделены характеристики, которые позволяют оценивать ПС с позиции пользователя, разработчика и управляющего проектом.
18. В интернете кнопка должна быть оформлена как текстовая ссылка, если она перемещает пользователя на другой фрагмент контента, и как кнопка – если она ...
- a.демонстрирует диалог.
  - b.открывает меню;
  - c.запускает действие;
  - d.показывает информацию;
25. В группе радиокнопок как минимум одна радиокнопка должна быть .....
- Выберите один ответ:
- a.проставлена по умолчанию;
  - b.отделена от других.
  - c.заблокирована от модификации;
  - d.доступна пользователю;
26. Сколько существует направлений стандартизации интерфейсов?
- a.5.
  - b.2;
  - c.4;
  - d.3;
27. Режимы, которые физически удерживаются пользователем, называются:
- a.квазирежимами;
  - b.жестами;
  - c.паттернами;
  - d.сценариями.
28. Для решения какой задачи при разработке интерфейса используется модель GOMS (goals, objects, methods and selection rules)?
- a.для разработки главного меню приложения;
  - b.для распределения основных функций приложения по экранным формам;
  - c.для количественной оценки прототипа интерфейса.
  - d.для качественной оценки прототипа интерфейса
29. 1. Программная инженерия:
- + software engineering
  - Инструменты создания программного обеспечения
  - Коллектив инженеров-программистов, разрабатывающих программное обеспечение для компьютеров
  - + Дисциплина, изучающая применение строгого систематического количественного подхода к разработке, эксплуатации и сопровождению программного обеспечения
  - Комплекс программ, предназначенный для решения инженерных задач, связанных с большим количеством расчетов
  - Инженерная индустрия применения прикладного программного обеспечения
  - + Совокупность инженерных методов и средств создания программного

обеспечения

- Прикладное программное обеспечение для решения офисных задач

30. Построение SADT-модели включает в себя выполнение следующих действий:

- Написание программного обеспечения для разрабатываемой системы по требованиям заказчика

+ Сбор информации об объекте, определение его границ

+ Определение цели и точки зрения модели, построение, обобщение и декомпозиция диаграмм

- Представление исследуемой системы в графическом виде

- Представление исследуемого объекта средствами системного моделирования

+ Критическая оценка, рецензирование и комментирование

- Разработка, отладка и тестирование программного обеспечения

- Использование графических пакетов для представления системы в виде модели

31. Моделирование основывается на принципах:

+ Выбор модели оказывает определяющее влияние на подход к решению проблемы и на то, как будет выглядеть это решение

- Декомпозиции системы на отдельные подзадачи

- Инкапсуляции и полиморфизма

- Децентрализации управления системой

+ Каждая модель может быть представлена с различной степенью точности; лучшие модели – те, что ближе к реальности

- Открытой трансформируемой системы

+ Нельзя ограничиваться созданием только одной модели. Наилучший подход при разработке любой нетривиальной системы – использовать совокупность нескольких моделей, почти независимых друг от друга

- Анализа и синтеза проектирования систем

32. В бизнес-процессах выделяют классы процессов:

- Решающие бизнес-процессы

- Регламентирующие бизнес-процессы

+ Основные бизнес-процессы

- Бизнес-процессы поведения системы

- Программируемые бизнес-процессы

- Экономические бизнес-процессы

+ Обеспечивающие бизнес-процессы

+ Бизнес-процессы управления

33. 5. CASE-средства классифицируются по следующим признакам:

+ По применяемым методологиям и моделям систем и БД

- По используемому программному обеспечению

- По этапам жизненного цикла программного обеспечения

+ По степени интегрированности с СУБД

- По уровням детализации и декомпозиции проектируемой системы

+ По доступным платформам

- По используемым языкам программирования

- По степени сложности моделируемой системы

34. 6. К малым интегрированным средствам моделирования относятся:

- ARIS Toolset

- Design/IDEF

+ ERwin

- + BPwin
- Designer/2000
- Paradigm Plus
- + Model Mart
- Rational Rose

35. К средним интегрированным средствам моделирования относятся:

- Rational Rose
- + Design/IDEF
- BPwin
- + Designer/2000
- + ARIS Toolset
- Model Mart
- Paradigm Plus
- ERwin

36. 8. Объектно-ориентированная методология (ООМ) включает в себя составные части:

- + Объектно-ориентированный анализ
- Объектно-ориентированный подкласс
- + Объектно-ориентированное проектирование
- Объектно-ориентированная парадигма
- Объектно-ориентированная экспозиция
- Объектно-ориентированное моделирование
- + Объектно-ориентированное программирование
- Объектно-ориентированная декомпозиция

37. К основным понятиям объектно-ориентированного подхода относятся:

- Обобщение
- + Полиморфизм
- + Инкапсуляция
- Реализация
- Агрегирование
- + Наследование
- Ассоциация
- Композиция

38. Главные принципы объектного подхода:

- + Абстрагирование
- Наследование
- + Ограничение доступа или инкапсуляция
- Безграничный доступ или инкапсуляция
- + Модульность и иерархия
- Агрегирование
- Композиция
- Обобщение и специализация

39. Дополнительные принципы объектного подхода:

- Реализация
- + Типизация
- + Параллелизм
- Внедрение

- Перпендикулярность
  - + Сохраняемость или устойчивость
  - Несохранимость или неустойчивость
  - Динамичность
40. К инструментальным средствам объектно-ориентированного анализа и проектирования относятся:
- + Rational Rose
  - Model Mart
  - + MS Visio
  - + ARIS
  - IDEF1X
  - Erwin
  - BPwin
  - JAM
41. К инструментальным средствам представления функциональных моделей относятся:
- JAM
  - + Model Mart
  - MS Visio
  - ARIS
  - IDEF0
  - + Erwin
  - + BPwin
  - Rational Rose
42. Методологии, поддерживаемые в BPwin:
- IDEF1X
  - + IDEF0
  - IDEF1
  - + IDEF3
  - IDEFX
  - IDEF5
  - + DFD
  - DFD1X
43. Диаграмма IDEF0 может содержать следующие типы диаграмм:
- Диаграмму классов
  - + Контекстную диаграмму, диаграмму декомпозиции
  - Диаграмму компонентов
  - + Диаграмму дерева узлов
  - Диаграмму взаимодействий
  - + Диаграмму только для экспозиции (FEO)
  - Диаграмму последовательности, диаграмму кооперации
  - Диаграмму узлов
44. Уровни логической модели:
- Диаграмма сущность
  - Диаграмма связь
  - Диаграмма пакетов
  - + Диаграмма сущность-связь

- Модель данных, основанная на классах
  - + Модель данных, основанная на ключах
  - Полная операционная модель
  - + Полная атрибутивная модель
45. Внутренние стрелки не входящие в состав диаграммы IDEF0:
- + mechanism- output
  - output-input
  - + mechanism- input
  - output-control
  - output-input feedback
  - output-control feedback
  - output-mechanism
  - + control feedback- mechanism
46. Типы стрелок не входящие в состав диаграммы IDEF0:
- Input
  - + Editor
  - Control
  - + Properties
  - Output
  - Mechanism
  - Call
  - + Dictionary
47. Quick Reports – создание простейших отчетов – позволяет создавать отчеты:
- Group/Totals. Табличный отчет с автоматической группировкой и сортировкой данных
  - Report Header. Печатается единожды в начале отчета
  - + Columnar. Простой табличный отчет
  - Page Header. Печатается в верхней части каждой страницы
  - + Vertical. Простой вертикальный отчет
  - Group Header. Печатается в начале каждой группы
  - + Blank Report. Бланк. Создается пустой бланк отчета, в который не включаются данные
  - Detail. Печатается для каждой строчки набора данных
48. BPwin допускает следующие переходы с одной нотации на другую:
- IDEF3 → DFD
  - DFD → IDEF0
  - + IDEF0 → DFD
  - DFD → DFD
  - IDEF3 → IDEF0
  - + IDEF0 → IDEF3
  - IDEF3 → IDEF3
  - + DFD → IDEF3
49. DFD описывает:
- Функции обработки стрелок (arrow)
  - + Функции обработки информации (работы)
  - Внешние ссылки (external references), объекты, сотрудников или отделы, которые участвуют в обработке информации

- + Документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке информации
- Функции обработки внешних ссылок
- + Внешние ссылки (external references), таблицы для хранения документов (хранилище данных, data store)
- Функции обработки документов
- Документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке внешних стрелок

50. ВРwin позволяет создавать на диаграмме DFD типы граничных стрелок:

- + Обычная граничная стрелка
- Специальная стрелка
- Внутренняя ссылка
- + Межстраничная ссылка и тоннельная стрелка
- + Внешняя ссылка
- Страничная ссылка и тневая стрелка
- Контрольная стрелка
- Стрелка механизм

51. Создать отчет в ВРwin возможно с помощью:

- + Встроенных шаблонов
- Программных модулей, создаваемых разработчиком на языке Visual Basic
- Создать отчет в ВРwin не возможно
- + Report Template Builder
- Отчет создается разработчиком
- Отдельно поставляемых программ
- Встроенных мастер-функций
- + RPTwin

52. В ВРwin 4.0 отчеты могут быть экспортированы в распространенные форматы:

- + Текстовый
- Символьный
- + MS Office
- Графический
- + HTML
- Internet Explorer
- Acrobat
- IBM Rational

53. Поддерживаемые в RPTwin типы операторов:

- + Текстовый оператор конкатенации (&)
- Символ
- Текст
- Дата
- + Арифметические
- Графический оператор конкатенации (&)
- + Логические
- Номер

54. Инструментальное средство ERwin позволяет:

- Редактировать и отлаживать программы
- + Проектировать на физическом и логическом уровне модели данных

- Управлять процессом конструирования ПО
  - Проектировать диаграммы вариантов использования и взаимодействий
  - + Проводить процессы прямого и обратного проектирования баз данных
  - Управлять процессом трансляции и отладки программ
  - + Выравнивать модель и содержимое системного каталога после редактирования
  - Проектировать контекстные диаграммы и диаграммы декомпозиции
55. ERwin позволяет создавать модели следующих типов:
- + Модель, имеющую только логический уровень
  - Модель, имеющую абстрактный уровень
  - Модель, имеющую абстрактный и физический уровни
  - + Модель, имеющую только физический уровень
  - Модель, имеющую абстрактный и логический уровни
  - + Модель, имеющую как логический уровень, так и физический уровень
  - Модель, имеющую концептуальный уровень
  - Модель, имеющую контекстный уровень
56. Для создания моделей ERwin используют международно признанные системы обозначений (нотации):
- IDEF0
  - + IDEF1X
  - IDEF3
  - DFD
  - + IE
  - + DM
  - IDEFDFD
  - IDEF3
57. К основным компонентам диаграммы ERwin относятся:
- + Сущности
  - Переходы
  - + Атрибуты
  - Классы
  - Слияния
  - Разветвления
  - Использования
  - + Связи
58. 30. Точки зрения организации в ARIS:
- Структура внедрения и структура потоков
  - + Организационная структура
  - Управленческая структура
  - Поведенческая структура
  - + Функциональная структура
  - Коммуникационная структура
  - + Структура данных и структура процессов
  - Обобщенная структура
59. Уровни точки зрения в ARIS:
- Описание структуры
  - + Описание требований
  - Описание поведения
  - Описание разработки
  - + Описание спецификации

- + Описание внедрения
- Описание процессов
- Описание классов

60. Методы описания, используемые в ARIS:

- EPT – метод описания потоков
- + EPC - метод описания процессов
- ERM - модель сущность-связь для описания структуры объектов
- + ERM - модель сущность-связь для описания структуры данных
- EPP – метод описания пакетов
- EPC – метод описания компонентов
- + UML - унифицированный язык моделирования
- EPT – метод описания нитей

61. К основным компонентам инструментов ARIS Toolset относятся:

- Internet (интернет)
- WordPad (ввод текстовых данных)
- Media (средство для медиа описания моделей)
- + Explorer (проводник)
- Acrobat (чтение текстовых данных)
- + Designer (средство для графического описания моделей)
- Document (для ввода различных параметров и атрибутов) и выноски
- + Таблица (для ввода различных параметров и атрибутов) и мастер (Wizards)

62. ARIS Business Optimizer позволяет:

- + Определять целевые затраты и рассчитывать стоимость продукта: во что компании обходится предоставление отдельных продуктов
- Принимать решения о времени начала и окончания работы над проектом
- + Принимать решения по аутсорсингу: стоит ли поручить выполнение бизнес-процессов внешнему поставщику услуг
- Определять последовательность работ , выполняемых в ходе работы над проектом
- Определять требования к персоналу компании, которая в дальнейшем будет эксплуатировать программное обеспечение
- Рассчитывать заработную плату сотрудников компании после внедрения программного обеспечения
- Планировать требования к обслуживающему персоналу, сопровождающему программное обеспечение
- + Планировать требования к персоналу: сколько необходимо сотрудников для оптимального выполнения работ

63. «Взгляды» ARIS:

- + Процессы
- Потоки
- + Функции (с целями)
- + Данные и организация
- Процедуры
- Управление и внедрение
- Нити
- Память

64. Уровни анализа ARIS для каждого «взгляда»:
- Поведение
  - + Требования
  - + Спецификации
  - Функции
  - Процедуры
  - Проверка
  - + Внедрение
  - Тестирование
65. MS Visio позволяет создавать схемы, чертежи, диаграммы с помощью:
- + Встроенных шаблонов
  - Панели инструментов
  - + Трафаретов
  - Графических редакторов
  - Дополнительного программного обеспечения
  - Панели рисования
  - + Стандартных модулей
  - Панели автофигур
66. Язык UML – это:
- Язык программирования высокого уровня
  - + Унифицированный язык моделирования
  - Язык для разработки систем искусственного интеллекта
  - + Unified Modeling Language
  - Язык управления базами данных
  - + Язык для визуализации, специфицирования, конструирования и документирования артефактов программных систем
  - Язык создания запросов в базах данных
  - Язык программирования низкого уровня
67. Моделирование в UML позволяет решать задачи:
- Анализа и синтеза систем управления
  - Разработать и отладить программное обеспечение
  - + Визуализировать систему в ее текущем или желательном для нас состоянии
  - Провести тестирование разработанного программного обеспечения
  - + Описать структуру или поведение системы; получить шаблон, позволяющий сконструировать систему
  - Смоделировать разрабатываемую информационную систему
  - + Документировать принимаемые решения, используя полученные модели
  - Рассчитать экономическую эффективность от внедрения программного обеспечения
68. Словарь UML включает строительные блоки:
- Зависимости
  - + Сущности
  - Слияния
  - Разветвления
  - + Связи
  - Группировки
  - + Диаграммы
  - Декомпозиции

69. 41. UML, как язык документирования, помимо исполняемого кода производит и другие продукты, включающие:
- + Требования, архитектуру, проектные решения
  - Спецификацию технических средств
  - + Дизайн, исходный код, проектные планы,
  - Требования к уровню квалификации разработчиков
  - Набор заданий для тестирования программного обеспечения
  - Требования к уровню квалификации персонала сопровождения
  - + Тесты, прототипы, релизы (версии)
  - Требования к выбору языка программирования
70. UML включает синтаксические и семантические правила для:
- Агрегации
  - Тестирования
  - + Имен, областей действия
  - Сборки
  - Сопровождения
  - + Видимости, целостности
  - Вывода из эксплуатации
  - + Исполнения
71. Применение языка UML существенно упрощает последовательное использование механизмов:
- + Спецификации, дополнения
  - + Принятые разделения
  - Выработки требований
  - Создания плана работ
  - + Механизмы расширения
  - Тестирования программного обеспечения
  - Конструирования ПО
  - Сопровождения ПО
72. Механизмы расширения UML включают:
- Исключения
  - + Стереотипы
  - Дополнения
  - Управления
  - + Помеченные значения
  - Слияния
  - + Ограничения
  - Объединения
73. Язык UML предназначен для:
- + Визуализации
  - Тестирования
  - Сопровождения
  - + Специфицирования
  - Снятия с эксплуатации
  - + Конструирования, документирования
  - Анализа требований
  - Обучения персонала

74. В объектно-ориентированном моделировании между классами существуют типы связей:
- Слияние
  - Линейность
  - + Зависимость
  - Разветвление
  - Цикличность
  - + Обобщение
  - + Ассоциация
  - Агрегация
75. В состав графического представления класса в языке UML входят части:
- Отношения
  - + Имя
  - Связи
  - + Атрибуты
  - Описание
  - Сущности
  - + Операции
  - Механизмы
76. Программное обеспечение делится на классы:
- Системное ПО и прикладное ПО
  - + Системное ПО, прикладное ПО и инструментальные средства разработки программ
  - Операционные системы, прикладное ПО, утилиты и драйверы
  - Прикладное ПО и инструментальные средства разработки программ
  - Системное ПО и инструментальные средства разработки программ
  - + Системное ПО, прикладное ПО и системы программирования
  - Операционные оболочки, операционные системы, офисные программы
  - + Системное ПО, прикладное ПО и инструментальное ПО
77. 49. Инструментальные средства разработки программ – это:
- + Средства создания новых программ
  - Сервисные средства разработки ПО
  - Аналитические средства разработки ПО
  - + Программное обеспечение, предназначенное для разработки и отладки новых программ
  - Средства отладки ПО
  - Средства тестирования ПО
  - + Аппаратные и программные инструменты разработки нового ПО
  - Технические инструментальные средства разработки ПО
78. 50. Аппаратные инструментальные средства разработки ПО – это:
- Система для разработки новых программ на конкретном языке программирования
  - Средства создания и редактирования текстов программ
  - + Микропроцессор и подключаемые (внешние) устройства
  - + Устройства вычислительной системы, специально предназначенные для поддержки разработки ПО
  - + Периферийные устройства, микропроцессор вычислительного комплекса, предназначенные для разработки нового ПО
  - Программное обеспечение, написанное на языках программирования низкого уровня
  - Программы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ

- Программы, используемые для корректировки и тестирования других прикладных или системных программ
79. 51. Программные инструментальные средства разработки ПО – это:
- + Программы, позволяющие выполнить все работы, определенные методологией проектирования ПО
  - Системное программное обеспечение, позволяющее сопровождать офисные программные пакеты
  - Средства создания текстовых документов
  - + Программное обеспечение, используемое на всех стадиях разработки нового ПО
  - Программное обеспечение для настройки офисных приложений на условия конкретного применения
  - + Программы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ
  - Устройство компьютера, специально предназначенное для поддержки разработки программных средств
  - Средства создания и редактирования текстовых документов
80. 52. Транслятор – это:
- + Программа, выполняющая перевод программы с одного языка программирования на другой
  - Комплекс программ мультимедийных технологий
  - + Программа, которая выполняет перевод программы с одного языка программирования на машинные коды
  - Программа-переводчик с одного иностранного языка на другой
  - Техническое устройство передачи и преобразования аудио и видеосигналов
  - Техническое устройство для кодирования и декодирования информации
  - Программное обеспечение для обеспечения защиты информации на компьютере
  - + Одно из основных средств автоматизации программирования для преобразования программы, написанный на машинно-независимом языке, в программу на машинном языке конкретной ЭВМ
81. 53. Компилятор – это:
- + Один из видов трансляторов
  - Прикладное программное обеспечение
  - Специальная утилита системного ПО
  - Операционная оболочка
  - + Переводит в коды сразу всю программу и создает независимый исполняемый файл
  - Программное обеспечение, используемое в издательских системах
  - + Программа, которая переводит программу, написанную на языке программирования высокого уровня в программу на машинном языке не участвуя в ее исполнении
  - Переводит в машинные коды 1 строчку программы и сразу ее выполняет
82. 54. Интерпретатор:
- Программа для создания и редактирования электронных таблиц
  - + Программа, анализирующая команды или операторы исходной программы и немедленно выполняющая их
  - Переводит в коды сразу всю программу и создает независимый исполняемый файл
  - + Переводит в машинные коды 1 строчку программы и сразу ее выполняет
  - Программа для создания и редактирования текстовых документов
  - + Один из видов трансляторов
  - Программа создания и управления базами данных
  - Программа создания файлов мультимедиа
83. 55. компоновщик – это:
- Программа для компоновки и оформления тестовых документов
  - + Редактор связей
  - Комплекс программ, для создания и ведения баз данных
  - + Программа, которая из одного или нескольких объектных модулей с

- привлечением библиотечных программ и стандартных подпрограмм формирует загрузочный модуль
  - Программное обеспечение для создания презентаций
  - + Программа сборки загрузочного модуля из полученных в результате отдельной компиляции объектных модулей с автоматическим поиском и присоединением библиотечных подпрограмм и процедур
  - Программа для поиска синтаксических и семантических ошибок в программе
  - Программа
84. 56. Отладчик:
- + Программа, облегчающая программисту выполнение отладки разрабатываемых им программ
  - Программа для создания системы защиты файла
  - Программа создания системы защиты от вирусных атак
  - + Программа, помогающая анализировать поведение отлаживаемой программы, обеспечивая ее трассировку
  - Операционная оболочка для создания и управления файловыми структурами
  - Системное программное обеспечение для настройки операционной системы
  - Программа создания и редактирования графических файлов
  - + Программа, позволяющая выполнять остановы в заданных точках, просмотреть текущие значения переменных и изменять их значения
85. 57. К этапам развития технологии разработки программного обеспечения относятся:
- + «Процедурное» программирование
  - Программирование на алгоритмических языках высокого уровня
  - + Структурный подход к программированию
  - Программирование на языках низкого уровня
  - + Компонентный подход и CASE-технологии
  - Машинно-ориентированное программирование
  - Машинно-независимое программирование
  - Подход к разработке ПО, основанный на стратегии поиска
86. 58. «Стихийное» программирование:
- Разработка программного обеспечения без предварительного составления плана-графики работ
  - + Первый этап в истории развития технологии разработки программного обеспечения, когда программирование фактически было искусством
  - + Период в истории разработки программного обеспечения, когда программа создавалась одним программистом, способным отслеживать последовательность выполняемых операций и местонахождения данных в программе
  - Разработка программ с использованием различных языков программирования низкого и высокого уровня
  - Разработка программ с элементами случайного выбора алгоритмов решения задачи
  - + Характеризуется тем, что типичная программа этого периода состояла из основной программы, области глобальных данных и набора подпрограмм (в основном библиотечных), выполняющих обработку всех данных или их части
  - Разработка программного обеспечения для решения задач теории вероятностей и математической статистики
  - Разработка программного обеспечения для решения задач, построенных на алгоритмах случайного поиска
87. 59. Структурный подход к программированию – это:
- + Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения
  - Создание программного обеспечения на основе структурной схемы решаемой задачи
  - Подход, требующий разработки структурной схемы алгоритма и программы решения задачи
  - + Подход, в основе которого лежит декомпозиция (разбиение на части) сложных

- систем с целью последующей реализации в виде отдельных небольших (до 40-50 операторов) подпрограмм
- Подход к решению задачи, требующий создание структурной схемы этапов работ по разработке программного обеспечения
  - Процесс создания программного обеспечения на основе структурной схемы исследуемого объекта или процесса
  - Технология разработки программного обеспечения на базе структурной схемы развития языков программирования
  - + Подход, требующий представления задачи в виде иерархии подзадач простейшей структуры
88. 60. Объектный подход к программированию – это:
- Технология создания сложного программного обеспечения, основанная на представлении задачи исследования как объекта
  - Технология создания сложного программного обеспечения, предназначенного для автоматизации технологических объектов
  - + Технология создания сложного программного обеспечения, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного типа (класса), а классы образуют иерархию с наследованием свойств
  - Технология создания сложного программного обеспечения, основанная на представлении программы как единого объекта
  - + Технология создания сложного программного обеспечения, позволяющая вести практически независимую разработку отдельных частей (объектов) программы
  - Технология создания сложного программного обеспечения, основанная на объектном представлении кода программы
  - + Технология создания сложного программного обеспечения, в основе которой лежат новые способы организации программ, основанные на механизмах наследования, полиморфизма, композиции, наполнения
  - Технология создания сложного программного обеспечения, основанная на объектно-ориентированном программировании
89. 61. Компонентный подход:
- + Предполагает построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения
  - + Предполагает взаимодействие между компонентами через стандартизованные двоичные интерфейсы и позволяет использовать исполняемые файлы в любом языке программирования, поддерживающем соответствующую технологию
  - Позволяет рассматривать объект исследования, как структуру, состоящую из отдельных компонент
  - способ написания исходного кода программного обеспечения
  - + Позволяет собрать объекты-компоненты в динамически вызываемые библиотеки или исполняемые файлы, и распространять в двоичном виде (без исходных текстов)
  - Способ отладки и тестирования программного обеспечения
  - Способ внедрения и опытной эксплуатации программного обеспечения.
  - Метод выработки требований к разработке программного обеспечения
90. 62. Управление требованиями:
- Задача выявления изначальных проблем заказчика и создание системы, удовлетворяющей этим требованиям
  - + Процесс систематического выявления, организации и документирования требований к сложной системе
  - Выявление требований заказчика и управление ими
  - + Задача, состоящая в том, чтобы понимать проблемы заказчиков в их предметной области и на их языке и создавать системы, удовлетворяющие их потребности
  - Процесс создания программного обеспечения и адаптация его под требования заказчика
  - Разработка требований к программному обеспечению и создание ПО на основе этих требований

- + Процесс, в ходе которого вырабатывается и обеспечивается соглашение между заказчиком и выполняющей проект группой по поводу меняющихся требований к системе
  - Разработка программного обеспечения и выработка требований к изменению работы системы заказчика
91. 63. К методам выявления требований относятся:
- Беседы с первыми руководителями предприятия, для которого разрабатывается программное обеспечение
  - Анализ научной и технической литературы, посвященной вопросам разработки программного обеспечения
  - Личные встречи и беседы со всеми сотрудниками предприятия
  - Анализ технической документации и на основе нее разработка требований к системе
  - На начальном этапе требования не выявляются, а формируются по мере разработки программного обеспечения
  - + Интервьюирование и анкетирование, мозговой штурм и отбор идей
  - + Совещания, посвященные требованиям, создание прототипов
  - + Раскадровки, прецеденты, обыгрывание ролей
92. 64. Требования к разрабатываемой системе должны включать:
- Разработку программного обеспечения и выработку требований к изменению работы системы заказчика
  - + Совокупность условий, при которых предполагается эксплуатировать будущую систему (аппаратные и программные ресурсы, предоставляемые системе; внешние условия ее функционирования; состав людей и работ, имеющих к ней отношение)
  - Построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения
  - + Описание выполняемых системой функций
  - Технологию создания сложного программного обеспечения, основанную а объектном представлении кода программы
  - + Ограничения в процессе разработки (директивные сроки завершения отдельных этапов, имеющиеся ресурсы, организационные процедуры и мероприятия, обеспечивающие защиту информации)
  - Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения
  - Технологию разработки программного обеспечения на базе структурной схемы развития языков программирования
93. 65. Типы средств, иллюстрирующие цели моделирования системы:
- + Функции, которые система должна выполнять
  - + Отношения между данными
  - + Зависящее от времени поведение системы (аспекты реального времени)
  - Способы отладки и тестирования программного обеспечения
  - Создание программного обеспечения на основе структурной схемы исследуемого объекта или процесса
  - Выявление требований заказчика и управление ими
  - Технология разработки программного обеспечения на базе структурной схемы развития языков программирования
  - Построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения
94. 66. Преимущества объектно-ориентированного подхода:
- Быстрота написания программного кода
  - Статичность конфигурации системы
  - + Возможность многократного использования
  - Низкая стоимость проекта
  - + Восприимчивость к изменениям
  - Отсутствие необходимости документирования

- Простота реализуемых моделей
  - + Реалистичное моделирование
95. 67. Требования – это:
- Документ, регулирующий отношения между заказчиком информационной системы и проектировщиком
  - + Некоторое свойство программного обеспечения, необходимое пользователю для решения проблемы при достижении поставленной цели
  - Оформленное заказчиком в виде документа задание на проектирование программного обеспечения
  - + Возможность, которую должна обеспечивать система
  - Характеристика проектируемого программного обеспечения с точки зрения разработчика
  - + Некоторое свойство программного обеспечения, которым должна обладать система или ее компонент, чтобы удовлетворить требования формальной документации
  - Оформленное разработчиком в виде документа задание на проектирование программного обеспечения
  - Характеристика проектируемого программного обеспечения с точки зрения заказчика
96. 68. Типичная схема процесса анализа С-требований включает в себя:
- + Идентификацию заказчика и проведение интервью с представителями заказчика
  - Разработку программного обеспечения в соответствии с требованиями заказчика
  - Изложение заказчику требований к системе на основе разработанного программного обеспечения
  - + Написание С-Требований в форме стандартного документа
  - Верификацию разработанного программного обеспечения в соответствии с требованиями заказчика
  - Составление плана мероприятий по анализу С-требований
  - + Проверку С-Требований и согласование их с заказчиком
  - Адаптацию разработанного программного обеспечения в соответствии с требованиями заказчика
97. 69. В классификацию требований к программной системе входят:
- Требования заказчика
  - Требования, накладываемые условиями эксплуатации
  - + Функциональные требования
  - Требования, накладываемые аппаратными средствами
  - + Нефункциональные требования
  - + Требования предметной области
  - Экономические требования
  - Требования разработчиков
98. 70. Процесс определения и анализа требований включает в себя:
- Анализ работы систем с аналогичной предметной областью
  - + Анализ предметной области, сбор и классификацию требований
  - Проведение совместных совещаний с представителями заказчика
  - + Разрешение противоречий и определение приоритетов
  - Адаптацию требований к разрабатываемому программному обеспечению
  - Декомпозицию общей задачи на подзадачи
  - + Проверку, специфицирование и документирование требований
  - Верификацию требований в соответствии с разработанным программным обеспечением
99. 71. Опорные точки зрения конечных пользователей системы программного обеспечения можно трактовать как:
- + Источник информации о системных данных
  - Структуру требований
  - Источник событий
  - Структуру событий
  - + Структуру представлений

- Получателей требований
  - Источник сценариев
  - + Получателей системных сервисов
100. 72. При аттестации требований выполняются следующие типы проверок документации требований:
- Проверка на совместимость
  - Проверка на управляемость
  - + Проверка правильности требований
  - + Проверка на непротиворечивость
  - Проверка на соответствие
  - Проверка на обратимость
  - + Проверка на полноту и на выполнимость
  - Проверка на заменяемость
101. 73. К методам аттестации требований относится:
- Тестирование
  - + Обзор требований
  - Верификация
  - Сравнительный анализ
  - + Прототипирование
  - Генерация случайных данных
  - + Генерация тестовых сценариев
  - Декомпозиция
102. 74. Уровни организационного управления при планировании разработки системы:
- + Стратегический
  - + Tактический
  - + Оперативный
  - Основной
  - вспомогательный
  - Дополнительный
  - Системный
  - Аналитический
103. 75. Для различных представлений проектируемой системы используют типы моделей:
- Статическая модель
  - Динамическая модель
  - + Модель классов
  - Модель декомпозиции
  - Модель размещения
  - + Модель состояний
  - + Модель взаимодействия
  - Модель агрегации
104. 76. Классификация бизнес-процессов включает следующие классы процессов:
- вспомогательные бизнес-процессы
  - + Основные бизнес-процессы
  - Дополнительные бизнес-процессы
  - + Обеспечивающие бизнес-процессы
  - Обслуживающие бизнес-процессы
  - Бизнес-процессы согласования
  - + Бизнес-процессы управления
  - Руководящие бизнес-процессы
105. 77. Типы D-требований:
- + Функциональные требования
  - Интерфейсные требования
  - + Нефункциональные требования
  - Программные требования

- + Обратные требования
  - Ограниченные требования
  - Производительные требования
  - Надежность
106. 78. Возможные способы организации D-требований:
- По атрибутам, по компонентам
  - По взаимоотношениям сущности
  - По пакетам и по иерархии компонентов
  - + По свойствам, по классам
  - + По вариантам использования
  - По узлам и по использованным процессам
  - + По состояниям и по иерархии функции
  - По прецедентам, по кооперациям
107. 79. К моделированию относится:
- + Система обозначений
  - Система атрибутов
  - + Синтаксис языка моделирования
  - Система свойств
  - Совокупность поведения объектов
  - + Совокупность графических объектов
  - Семантика языка моделирования
  - Совокупность текстовых объектов
108. 80. Классификация имитационных моделей:
- Статистическая
  - Адаптивная
  - + Статическая или динамическая
  - Структурная
  - + Сетерминированная или стохастическая
  - + Непрерывная или дискретная
  - Объединенная
  - Декомпозиционная
109. 81. Принципы разработки эффективного пользовательского интерфейса:
- Сложность, графика
  - + Структура, простота
  - Связь, обработка
  - + Видимость, обратная связь
  - Невидимость, сложность
  - + Толерантность, повторное использование
  - Первое использование, итерация
  - Интеграция, повторение
110. 82. Принципы разработки программного обеспечения:
- Коллективный процесс разработки
  - + Индивидуальный процесс разработки
  - Параллельный процесс разработки
  - + Командный процесс разработки
  - Промежуточный процесс разработки
  - + Модель зрелости возможностей
  - Модель законченности возможностей
  - Модель готовности процессов
111. 83. Типы интерфейсных требований:
- + Пользовательские требования
  - + Аппаратные требования
  - Административные требования
  - Требования к производительности
  - + Программные и коммуникационные требования
  - Требования к надежности

- Требования к устойчивости
  - Атрибуты программной системы и другие требования
112. 84. Технология проектирования определяется как совокупность составляющих:
- Поэтапная процедура
  - + Пошаговая процедура
  - Модели и правила
  - + Критерий и правила
  - Тестирование
  - + Нотаций
  - Прецеденты
  - Классы
113. 85. Разработка и сопровождение ИС в конкретной организации и конкретном проекте должна поддерживаться стандартами:
- Стандарт организации
  - Стандарт конкретного проекта
  - + Стандарт проектирования
  - Стандарт оценки
  - + Стандарт оформления проектной документации
  - Стандарт аудита
  - Стандарт оформления разработки
  - + Стандарт пользовательского интерфейса
114. Результатами проектирования архитектуры являются:
- Модель административного интерфейса
  - Модель процессов
  - Модель потоков
  - Модель классов
  - Модель данных
  - Модель пользовательского интерфейса
  - Модель компонентов
  - Модель узлов
115. Какие работы включает процесс разработки программного обеспечения:
- Документирование, управление конфигурацией
  - Управление, создание инфраструктуры
  - Структура из процессов, работ, задач
  - Обеспечение качества, верификация
  - Анализ требований, проектирование
  - Программирование, сборка, тестирование
  - Ввод в действие, приемка
  - Совместный анализ, аудит
116. Какие технологии разработки программ используются в современном программировании:
- Визуальные
  - Событийные
  - Структурные
  - Объектно-ориентированные
  - Модульные
  - Текстуальные
  - Графические
  - Машинно-ориентированное
117. Объектно-ориентированное проектирование использует инструментальные средства:

- Model mart
- Rational Rose
- Bpwin
- ARIS
- Idef1X
- Erwin
- MS Visio
- Jam

118. Проектирование функциональных моделей поддерживается инструментальными средствами:

- Jam
- Model Mart
- MS visio
- ERwin
- Idef0
- Aris
- Rational rose
- BPwin

119. IEEE – это:

- Коммерческая организация ученых и исследователей
- Просто принятое обозначение, расшифровки не имеет
- Обозначение всемирной компьютерной сети
- Всемирная некоммерческая техническая профессиональная ассоциация ученых и исследователей
- Такая аббревиатура нигде не используется
- Institute Of Electrical and Electronic Engineers, Inc
- Американская организация ученых-экономистов
- Институт инженеров радиотехники и электротехники

120. Ядро знаний SWEBOOK – это:

- ГОСТ на разработку программного обеспечения
- + Нормативный документ, разработанный IEEE
- ГОСТ на разработку информационных систем
- Документ, устанавливающий правовые отношения между заказчиком и разработчиком программного обеспечения
- основополагающий научно-технический документ, который отображает мнение специалистов в области программной инженерии
- Документ, устанавливающий методику тестирования и испытания программного обеспечения
- Документ, который согласуется с современными регламентированными процессами жизненного цикла ПО стандарта ISO/IEC 12207
- ГОСТ на разработку и комплектацию сопровождающей документации

121. Каждая область ядра знаний SWEBOOK представляется:

- Структурной схемой
- + Общей схемой описания
- Диаграммой UML
- Описанием и комментариями
- + Определением понятийного аппарата, методов и средств инженерной деятельности
- Определением языка программирования
- + Определением инструментов поддержки инженерной деятельности
- Иерархической диаграммой

122. К основным областям знаний SWEBOOK относятся:
- + Инженерия требований, проектирование ПО
  - Анализ деятельности системы
  - Управление проектами
  - + Конструирование ПО
  - Управление персоналом
  - + Тестирование ПО, сопровождение ПО
  - Управление конфигурацией
  - Инженерия качества программных средств
123. 95. К организационным областям знаний SWEBOOK относятся:
- Инженерия требований
  - + Управление конфигурацией, управление проектами
  - Конструирование ПО
  - + Процесс инженерии программных средств, методы и средства программной инженерии
  - Проектирование ПО
  - Сопровождение ПО
  - Тестирование ПО
  - + Инженерия качества программных средств
124. В рамках Rational Unified Process (RUP) набор действий по разработке программ включает этапы:
- Создание структурных схем
  - Определения входных, выходных данных
  - Согласование стоимости проекта
  - Согласования требований с заказчиком
  - Создания бизнес-моделей
  - + Определение требований
  - + Проектирование, программирование
  - + Тестирование, внедрение
125. Этапы разработки консалтинговых проектов включают в себя:
- + Анализ первичных требований и планирование работ
  - Снятие программного продукта с эксплуатации
  - Декомпозицию задачи на подзадачи
  - Разработку спецификации и документации
  - + Проведение обследования деятельности предприятия
  - Тестирование и сопровождение программного обеспечения
  - + Построение моделей деятельности предприятия (модели AS – IS – “как есть” и модели TO – BE – “как должно быть”)
  - Разработку программного обеспечения
127. Концепции, лежащие в основе модульного программирования:
- Объем реализации и время исполнения (реакции)
  - Мера автоматизма в работе реализации и инструментах разработки
  - Визуальность и тестируемость разработки
  - + Функциональная декомпозиция, пространственная и временная группировка информации (модульность)
  - + Упрощение связей
  - + Комментируемость функций и данных
  - Надежность, устойчивость
  - Безопасность
128. Инструмент разработки программ выбирается на основе:

- Визуальности, набора реализуемых технологий
- Мощности множества элементов разработки
- Системного подхода к анализу, проектированию и реализации ПО
- Функциональной декомпозиции, пространственной и временной группировка информации (модульность)
- Упрощения связей, комментируемости функций и данных
- + Объема реализации и времени исполнения (реакции), надежности, устойчивости, безопасности
- + Меры автоматизма в работе реализации и инструментах разработки
- + Визуальности и тестируемости разработки

## 6.5. Вопросы к зачету

- 1 Введение в предметную область человеко-машинного интерфейса
- 2 Виды человеко-машинного интерфейса
- 3 Проектирование человеко-машинного интерфейса
- 4 Понятие опыта взаимодействия (user experience, UX).
- 5 Дизайн UI и UX. Отличия и сходства.
- 6 Характеристики качества интерфейса (эргономические показатели). Достоинства и недостатки их использования при проектировании интерфейсов.
- 7 Определение понятия юзабилити (usability).
- 8 Подходы к проектированию интерфейсов.
- 9 «Золотые» правила Шнейдермана (Eight Golden Rules of Interface Design).
- 10 Эвристические правила Я. Нильсена (Usability Heuristics for User Interface Design).
- 11 Этапы проектирования и дизайна интерфейсов.
- 12 Пользовательские истории взаимодействия (user stories). Назначение и основные понятия.
- 13 Описание целевой аудитории. Основные понятия, цели и назначение.
- 14 Структурные схемы страниц (wireframes). Назначение и основные понятия.
- 15 Основные методологии разработки интерфейсов.
- 16 Проектирование, ориентированное на пользователей (Human Centred Design). Достоинства и недостатки.
- 17 Проектирование, ориентированное на задачи пользователей (Task Centered Design). Достоинства и недостатки.
- 18 Дизайн, ориентированный на мотивы пользователей (Goal Centered Design). Достоинства и недостатки.
- 19 Построение модели пользователя
- 20 Виды межпрограммного интерфейса
- 21 Проектирование межпрограммного интерфейса
- 22 Клиент-серверный межпрограммный интерфейс
- 23 Обмен данными через именованные блоки памяти
- 24 Введение в предметную область классического оконного интерфейса
- 25 Основные оконные примитивы
- 26 Составные оконные примитивы
- 27 Модель-представление
- 28 Сложные элементы управления
- 29 Комбинирование оконных элементов
- 30 Введение в предметную область интерфейса программы с внешними источниками данных
- 31 Объектно ориентированный подход к БД
- 32 Клиент-серверный доступ к реляционным источникам

- 33 Юзабилити-консалтинг. Основные понятия и этапы  
34 Тестирование пользовательского интерфейса (UI testing, A/B testing).

#### 6.6. Вопросы к экзамену

Не предусмотрено.

### 7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)

#### 7.1. Основная литература

1. Коваленко, В. В. Проектирование информационных систем : учебное пособие / В.В. Коваленко. — 2-е изд., перераб. и доп. — Москва : ИНФРА-М, 2021. — 357 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/987869. - ISBN 978-5-00091-637-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/987869> (дата обращения: 10.02.2022). — Режим доступа: по подписке.
2. Заботина, Н. Н. Проектирование информационных систем : учебное пособие / Н. Н. Заботина. — Москва : ИНФРА-М, 2020. — 331 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-004509-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1036508> (дата обращения: 10.02.2022). — Режим доступа: по подписке.
3. Жуков, Р. А. Язык программирования Python: практикум : учебное пособие / Р.А. Жуков. — Москва : ИНФРА-М, 2022. — 216 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/textbook\_5cb5ca35aaa7f5.89424805. - ISBN 978-5-16-016971-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1689648> (дата обращения: 10.02.2022). — Режим доступа: по подписке.

#### 7.2. Дополнительная литература

1. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2022. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/491029> (дата обращения: 10.02.2022).
2. Проектирование информационных систем : учебник и практикум для вузов / под общей редакцией Д. В. Чистова. — Москва : Издательство Юрайт, 2022. — 258 с. — (Высшее образование). — ISBN 978-5-534-00492-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/489307> (дата обращения: 10.02.2022).

3. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2022. — 147 с. — (Высшее образование). — ISBN 978-5-534-09172-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/491629> (дата обращения: 10.02.2022).

#### 7.2. Программное обеспечение

1. Сетевой компьютерный класс, оснащенный современной техникой
2. Офисный программный пакет (например, Microsoft Office 2007 или более поздних версий).
3. Web-браузер Edge, Mozilla Firefox или Google Chrome
4. ПО для вывода на экран для проектора
5. Редактор Paint от MS Windows 7-10.

#### Электронные ресурсы

1. Электронная библиотека «Знаниум»: <https://znanium.com/>
2. Электронная библиотека «Юрайт»: <https://urait.ru/>
3. Научная электронная библиотека «Elibrary.ru»: <https://www.elibrary.ru/defaultx.asp>

#### 7.3. Методические указания и материалы по видам занятий

### 8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)

№п/п	Наименование оборудованных учебных кабинетов, лабораторий	Перечень оборудования и технических средств обучения
1.	Лекционная аудитория	Персональный компьютер, мультимедийный проектор
2.	Компьютерный класс	Персональные компьютеры (IBM PC-совместимые) под управлением ОС Microsoft Windows, компьютерная сеть, доступ в сеть Интернет

