Министерство образования Самарской области

**АССОЦИАЦИЯ ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ "ИНСТИТУТ НАПРАВО.РУ"**

УТВЕРЖДЕНО
Приказ директора
от 07.08.2024 № 012/2/с

Воробьев К.А.

# ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПМ.01 Разработка программных модулей

Специальность 09.02.10
Разработка компьютерных игр, дополненной и виртуальной реальности

Квалификация: разработчик компьютерных игр, дополненной и виртуальной реальности
Очная форма обучения
На базе основного общего образования
Срок получения образования: 3 года 10 месяцев

Тольятти, 2024г.

## 1. Паспорт фонда оценочных средств

Фонд оценочных средств предназначен для текущего контроля и промежуточной аттестации по компоненту «ПМ.01 Разработка программных модулей» основной образовательной программы ИНСТИТУТ НАПРАВО.РУ по специальности 09.02.10 «Разработка компьютерных игр, дополненной и виртуальной реальности». Квалификация — разработчик компьютерных игр, дополненной и виртуальной реальности.

Общий объём компонента — 340 академических часов. Форма промежуточной аттестации: Экзамен — семестр 5; Дифференцированный зачёт — семестр 5.

Нормативное основание: федеральный государственный образовательный стандарт, утверждённый приказом Министерства просвещения Российской Федерации от 25 июня 2024 г. № 441; учебный план и соответствующая рабочая программа.

## 2. Проверяемые результаты и формы контроля

| Проверяемая компетенция и результат | Оценочное средство | Подтверждение результата |
|---|---|---|
| ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. | Практическое задание 1 | Рабочий проект, исходные материалы, протокол проверки и защита |
| ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием. | Практическое задание 2 | Рабочий проект, исходные материалы, протокол проверки и защита |
| ПК 1.3. Оформлять программный код в соответствии с установленными требованиями. | Практическое задание 3 | Рабочий проект, исходные материалы, протокол проверки и защита |
| ПК 1.4. Использовать систему контроля версий программного кода для коллективной разработки программного кода. | Практическое задание 4 | Рабочий проект, исходные материалы, протокол проверки и защита |
| ПК 1.5. Выполнять отладку программных модулей с использованием специализированных программных средств. | Практическое задание 5 | Рабочий проект, исходные материалы, протокол проверки и защита |
| ПК 1.6. Осуществлять рефакторинг и оптимизацию программного кода. | Практическое задание 6 | Рабочий проект, исходные материалы, протокол проверки и защита |
| ПК 1.7. Разрабатывать процедуры интеграции программных модулей и платформы/фреймворка. | Практическое задание 7 | Рабочий проект, исходные материалы, протокол проверки и защита |
| ПК 1.8. Выполнять интеграцию программных модулей и платформы/фреймворка. | Практическое задание 8 | Рабочий проект, исходные материалы, протокол проверки и защита |
| ПК 1.9. Осуществлять инсталляцию, настройку и обслуживание программного обеспечения. | Практическое задание 9 | Рабочий проект, исходные материалы, протокол проверки и защита |

## 3. Организация оценивания и критерии

Текущий контроль проводится по завершении тем и практических работ. Обучающийся получает условие задания, исходные данные, требования к результату и критерии оценки. Результат фиксируется в ведомости или журнале с указанием вида работы и даты.

При промежуточной аттестации проверяются результаты всех завершённых разделов соответствующего семестра. Состав варианта обеспечивает сопоставимость сложности.

При наличии нескольких семестров аттестации задания выбираются только из освоенного к этому сроку содержания рабочей программы.

Практическая работа оценивается по правильности и полноте результата, обоснованности способа решения, самостоятельности выполнения, соблюдению требований безопасности и качеству оформления. По заданиям с собственным ключом применяется приведённый ключ; при отсутствии отдельной шкалы используется следующая.

| Оценка | Условия |
|---|---|
| Отлично | 90–100% максимального балла; результат полный, проверен и обоснован. |
| Хорошо | 75–89%; результат правильный, имеются отдельные неточности, не искажающие основное решение. |
| Удовлетворительно | 60–74%; выполнены основные требования; отдельные ошибки исправляются при уточняющих вопросах. |
| Неудовлетворительно | Менее 60%; основные требования не выполнены, результат не подтверждён или содержит существенные ошибки. |

Для зачёта требуется выполнить обязательные работы и подтвердить достижение проверяемых результатов. Для дифференцированного зачёта выставляется оценка по установленной шкале. Нарушение безопасности, создающее угрозу здоровью, является основанием остановить выполнение работы; невыполненные действия оцениваются как неподтверждённые.

Для обучающихся с инвалидностью и ограниченными возможностями здоровья обеспечиваются доступные форматы материалов, необходимые технические средства и специальные условия проведения оценивания с сохранением проверяемых результатов.

## 4. Контрольно-оценочные материалы

## Задание 1. Алгоритм обработки инвентаря

Проверяемый результат: умение формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

Задание. Спроектируйте добавление предмета в инвентарь с ограничением вместимости и объединением одинаковых предметов. Определите входы, выходы и исключительные ситуации.

Представляемые материалы. Блок-схема или псевдокод; трассировка для пустого, заполненного и частично заполненного инвентаря.

Показатели правильного выполнения. Предмет не теряется и не дублируется; отказ при переполнении предсказуем; все ветви алгоритма проверены.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

| Показатель | Максимальный балл |
|---|---|
| Соответствие результата условию и полнота реализации | 4 |
| Корректность проверки и воспроизводимость результата | 3 |
| Качество исходных материалов и документации | 2 |
| Обоснование решений при защите | 1 |

| Показатель | Максимальный балл |
|---|---|
| Всего | 10 |

Ответ для проверяющего. Предмет не теряется и не дублируется; отказ при переполнении предсказуем; все ветви алгоритма проверены. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

## Задание 2. Программный модуль инвентаря

Проверяемый результат: умение разрабатывать программные модули в соответствии с техническим заданием.

Задание. Реализуйте спроектированный алгоритм в отдельном модуле. Отделите модель данных от отображения. Добавьте проверки входных данных.

Представляемые материалы. Исходный код, инструкция запуска, автоматические тесты.

Показатели правильного выполнения. Корректные операции добавления и удаления; неверные данные не приводят к аварийному завершению; результат соответствует техническому заданию.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

| Показатель | Максимальный балл |
|---|---|
| Соответствие результата условию и полнота реализации | 4 |
| Корректность проверки и воспроизводимость результата | 3 |
| Качество исходных материалов и документации | 2 |
| Обоснование решений при защите | 1 |
| Всего | 10 |

Ответ для проверяющего. Корректные операции добавления и удаления; неверные данные не приводят к аварийному завершению; результат соответствует техническому заданию. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

## Задание 3. Оформление и документирование кода

Проверяемый результат: умение оформлять программный код в соответствии с установленными требованиями.

Задание. Примените единый стиль именования и форматирования. Документируйте публичный интерфейс и нетривиальные ограничения.

Представляемые материалы. Оформленный репозиторий, описание интерфейса и пример использования.

Показатели правильного выполнения. Имена отражают назначение; нет неиспользуемого кода; документация позволяет вызвать модуль без чтения его реализации.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

| Показатель | Максимальный балл |
|---|---|
| Соответствие результата условию и полнота реализации | 4 |
| Корректность проверки и воспроизводимость результата | 3 |
| Качество исходных материалов и документации | 2 |
| Обоснование решений при защите | 1 |
| Всего | 10 |

Ответ для проверяющего. Имена отражают назначение; нет неиспользуемого кода; документация позволяет вызвать модуль без чтения его реализации. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

## Задание 4. Совместная разработка

Проверяемый результат: умение использовать систему контроля версий программного кода для коллективной разработки программного кода.

Задание. Создайте ветку для новой функции, внесите изменения и объедините их через запрос на включение. Воспроизведите и разрешите конфликт.

Представляемые материалы. История изменений, две ветки, описание разрешённого конфликта.

Показатели правильного выполнения. Изменения прослеживаются; итоговая ветка собирается; конфликт разрешён без утраты требуемой функциональности.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

| Показатель | Максимальный балл |
|---|---|
| Соответствие результата условию и полнота реализации | 4 |
| Корректность проверки и воспроизводимость результата | 3 |
| Качество исходных материалов и документации | 2 |
| Обоснование решений при защите | 1 |
| Всего | 10 |

Ответ для проверяющего. Изменения прослеживаются; итоговая ветка собирается; конфликт разрешён без утраты требуемой функциональности. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

## Задание 5. Отладка дефектного модуля

Проверяемый результат: умение выполнять отладку программных модулей с использованием специализированных программных средств.

Задание. Найдите причины повторного начисления очков и обращения к отсутствующему объекту. Используйте точки останова и наблюдение переменных.

Представляемые материалы. Протокол воспроизведения, исправление, регрессионные тесты.

Показатели правильного выполнения. Указана первопричина каждого дефекта; исправление устраняет её; ранее работавшие сценарии сохраняются.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

| **Показатель** | **Максимальный балл** |
|---|---|
| Соответствие результата условию и полнота реализации | 4 |
| Корректность проверки и воспроизводимость результата | 3 |
| Качество исходных материалов и документации | 2 |
| Обоснование решений при защите | 1 |
| Всего | 10 |

Ответ для проверяющего. Указана первопричина каждого дефекта; исправление устраняет её; ранее работавшие сценарии сохраняются. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

## Задание 6. Рефакторинг и оптимизация

Проверяемый результат: умение осуществлять рефакторинг и оптимизацию программного кода.

Задание. Устраните дублирование в обработчиках игровых событий. Измерьте время выполнения до и после изменения на одинаковом наборе данных.

Представляемые материалы. Сравнение версий, результаты измерений, тесты эквивалентности.

Показатели правильного выполнения. Поведение сохранено; сравнение воспроизводимо; улучшение подтверждено измерениями, а не субъективной оценкой.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

| **Показатель** | **Максимальный балл** |
|---|---|
| Соответствие результата условию и полнота реализации | 4 |
| Корректность проверки и воспроизводимость результата | 3 |
| Качество исходных материалов и документации | 2 |
| Обоснование решений при защите | 1 |
| Всего | 10 |

Ответ для проверяющего. Поведение сохранено; сравнение воспроизводимо; улучшение подтверждено измерениями, а не субъективной оценкой. Оценка подтверждается

исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

## Задание 7. Проектирование интеграции

Проверяемый результат: умение разрабатывать процедуры интеграции программных модулей и платформы/фреймворка.

Задание. Опишите контракт обмена данными между модулем инвентаря и игровым движком: события, структуры данных, жизненный цикл и ошибки.

Представляемые материалы. Спецификация интерфейса, схема взаимодействия, план интеграционных тестов.

Показатели правильного выполнения. Для каждого события определены отправитель и получатель; учтены повторные вызовы, отсутствие данных и освобождение ресурсов.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

| **Показатель** | **Максимальный балл** |
|---|---|
| Соответствие результата условию и полнота реализации | 4 |
| Корректность проверки и воспроизводимость результата | 3 |
| Качество исходных материалов и документации | 2 |
| Обоснование решений при защите | 1 |
| Всего | 10 |

Ответ для проверяющего. Для каждого события определены отправитель и получатель; учтены повторные вызовы, отсутствие данных и освобождение ресурсов. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

## Задание 8. Интеграция с движком

Проверяемый результат: умение выполнять интеграцию программных модулей и платформы/фреймворка.

Задание. Подключите модуль к сцене. Реализуйте получение предмета, отображение остатка и сохранение состояния между запусками.

Представляемые материалы. Рабочая сцена, конфигурация, протокол интеграционных проверок.

Показатели правильного выполнения. Повторная загрузка восстанавливает состояние; события обрабатываются однократно; модуль не зависит от конкретного элемента интерфейса.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

| Показатель | Максимальный балл |
|---|---|
| Соответствие результата условию и полнота реализации | 4 |
| Корректность проверки и воспроизводимость результата | 3 |
| Качество исходных материалов и документации | 2 |
| Обоснование решений при защите | 1 |
| Всего | 10 |

Ответ для проверяющего. Повторная загрузка восстанавливает состояние; события обрабатываются однократно; модуль не зависит от конкретного элемента интерфейса. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

## Задание 9. Установка и сопровождение

Проверяемый результат: умение осуществлять инсталляцию, настройку и обслуживание программного обеспечения.

Задание. Подготовьте развёртывание проекта на чистом рабочем месте: зависимости, настройки и восстановление после неудачного обновления.

Представляемые материалы. Пакет установки, перечень зависимостей, журнал проверки и инструкция отката.

Показатели правильного выполнения. Установка воспроизводима по инструкции; версии зависимостей указаны; откат возвращает работоспособное состояние.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

| Показатель | Максимальный балл |
|---|---|
| Соответствие результата условию и полнота реализации | 4 |
| Корректность проверки и воспроизводимость результата | 3 |
| Качество исходных материалов и документации | 2 |
| Обоснование решений при защите | 1 |
| Всего | 10 |

Ответ для проверяющего. Установка воспроизводима по инструкции; версии зависимостей указаны; откат возвращает работоспособное состояние. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

## Дополнительные задания и формы

МДК.01.01 Разработка алгоритмов и программного кода Вопросы теоретической части Раздел 1. Общие принципы разработки программных продуктов Задание 1. Перечень контрольных вопросов по разделу

1. Технологии разработки ПО.

2. Методы разработки ПО.

3. Классификация программных продуктов.

4. Классификация инструментария технологии программирования.

5. CASE-технология создания информационных систем.

6. Классификация пакетов прикладных программ.

7. Понятия программного модуля, программного продукта, программного средства.

8. Жизненный цикл разработки ПО.

9. Процессы жизненного цикла.

10. Модели разработки ПО: спиральная, каскадная. Шкала оценки контрольных вопросов: Оценка 5 «отлично» выставляется студенту, если он правильно ответил на поставленный вопрос. Оценка 4 «хорошо» выставляется студенту, если он грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос. Оценка 3 «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки. Оценка 2 «неудовлетворительно» выставляется студенту, который не знает значительной части заданного вопроса. Тестовые задания к разделу

1. Что из перечисленного относится к специфическим особенностям ПО как продукта:

1. низкие затраты при дублировании

2. универсальность

3. простота эксплуатации

4. наличие поддержки (сопровождения) со стороны разработчика

2. Этап, занимающий наибольшее время, в жизненном цикле программы:

1. сопровождение

2. проектирование

3. тестирование

4. программирование

3. Этап, занимающий наибольшее время, при разработке программы:

1. тестирование

2. сопровождение

3. проектирование

4. программирование

4. Первый этап в жизненном цикле программы:

1. анализ требований

2. формулирование требований

3. проектирование

4. автономное тестирование

5. Один из необязательных этапов жизненного цикла программы:

1. проектирование

2. тестирование

3. программирование

4. оптимизация

6. Самый большой этап в жизненном цикле программы:

1. эксплуатация

2. изучение предметной области

3. тестирование

4. корректировка ошибок

7. Какой этап выполняется раньше:

1. тестирование

2. отладка

3. эксплуатация

4. оптимизация

8. Какой из этапов выполняется раньше остальных:

1. отладка

2. оптимизация

3. программирование

4. тестирование

9. Что выполняется раньше:

1. компиляция

2. отладка

3. компоновка

4. тестирование

10. В стадии разработки программы не входит:

1. постановка задачи

2. составление спецификаций

3. автоматизация программирования

4. эскизный проект

11. Самый важный критерий качества программы:

1. надежность

2. работоспособность

3. быстродействие

4. простота эксплуатации

12. Один из способов оценки качества ПО:

1. сравнение с аналогами

2. наличие документации

3. оптимизация программы

4. структурирование алгоритма

13. Существует ли связь между эффективностью и оптимизацией программы:

1. да

2. нет

3. в случаях коллективной разработки ПО

4. в случаях индивидуальной разработки ПО

14. Наиболее важным критерием качества при разработке ПО является:

1. быстродействие

2. удобство в эксплуатации

3. надежность

4. эффективность

15. Одним из способов оценки надежности ПО является:

1. сравнение с аналогами

2. трассировка

3. оптимизация

4. тестирование Шкала оценки тестовых вопросов: «2» – от 0 до 40% «3» – от 41% до 60% «4» – от 61% до 80% «5» – от 81% до 100% Выполнение студентом тестового задания демонстрирует владение им следующих компетенций: ОК 1 Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес. ОК 2 Организовывать собственную деятельность, определять методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. ОК 5 Использовать информационно-коммуникационные технологии для совершенствования профессиональной деятельности. ОК 6 Работать в коллективе и команде, обеспечивать ее сплочение, эффективно общаться с коллегами, руководством, потребителями. ОК 7 Ставить цели, мотивировать деятельность подчиненных, организовывать и контролировать их работу с принятием на себя ответственности за результат выполнения заданий. ОК 8 Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации. Практические работы Перечень тем практических работ:

1. «Разработка и анализ требований к программной системе»

2. «Проектирование программной системы»

3. «Техническое задание» Шкала оценки практических работ: оценка «5» ставится, если:

- учащийся самостоятельно выполнил все этапы решения задач на ЭВМ

- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы; оценка «4» ставится, если:

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с ЭВМ в рамках поставленной задачи

- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок

- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи. оценка «3» ставится, если:

- работа выполнена не полностью, допущено более трех ошибок, но учащийся владеет основными навыками работы на ЭВМ, требуемыми для решения поставленной задачи. оценка «2» ставится, если:

- допущены существенные ошибки, показавшие, что учащийся не владеет обязательными знаниями, умениями и навыками работы на ЭВМ или значительная часть работы выполнена не самостоятельно. Рефераты Перечень тем рефератов:

1. Взаимосвязь между процессами жизненного цикла программного обеспечения.

2. Кризис программирования и способы выхода из него.

**3. Структура процесса и организации, занимающейся разработкой программного**

обеспечения.

4. Роль метрики в процессе разработки программного обеспечения.

5. Парадигма Бейзили.

6. Набор основных метрических показателей.

**7. Структура разделения работ по созданию программного обеспечения.**

8. Оценка объемов и сложности программного обеспечения.

9. Оценка возможных рисков при выполнении программных проектов.

10. Собираемые метрики, используемые стандарты, методы и шаблоны.

11. Методы получения информации при проектировании программного продукта.

12. Техническое проектирование программного продукта.

13. Классификация и назначение интерфейсов.

14. Типы пользовательского интерфейса. Назначение и характеристика.

15. Язык программирования Turbo Pascal. Характеристики.

16. Язык программирования Алгол. Характеристики.

17. Язык программирования Delphi. Характеристики.

18. Языки программирования Basic и Visual Basic. Характеристики.

19. Язык программирования Fortran. Характеристики.

20. Язык программирования Assembler. Характеристики. Шкала оценки рефератов: Критерии и показатели, используемые при оценивании учебного реферата Критерии Показатели

1.Новизна реферированного текста Макс. - 20 баллов

- актуальность проблемы и темы

- новизна и самостоятельность в постановке проблемы, в формулировании нового аспекта выбранной для анализа проблемы

- наличие авторской позиции, самостоятельность суждений.

2. Степень раскрытия сущности проблемы Макс. - 30 баллов

- соответствие плана теме реферата

- соответствие содержания теме и плану реферата

- полнота и глубина раскрытия основных понятий проблемы

- обоснованность способов и методов работы с материалом

- умение работать с литературой, систематизировать и структурировать материал

- умение обобщать, сопоставлять различные точки зрения по рассматриваемому вопросу, аргументировать основные положения и выводы.

3. Обоснованность выбора источников Макс. - 20 баллов

- круг, полнота использования литературных источников по проблеме

- привлечение новейших работ по проблеме (журнальные публикации, материалы сборников научных трудов и т.д.).

4. Соблюдение требований к оформлению Макс. - 15 баллов

- правильное оформление ссылок на используемую литературу

- грамотность и культура изложения

- владение терминологией и понятийным аппаратом проблемы

- соблюдение требований к объему реферата

- культура оформления: выделение абзацев.

5. Грамотность Макс. - 15 баллов

- отсутствие орфографических и синтаксических ошибок, стилистических погрешностей

- отсутствие опечаток, сокращений слов, кроме общепринятых

- литературный стиль. Оценивание реферата Реферат оценивается по 100 балльной шкале, балы переводятся в оценки успеваемости следующим образом:

• 86 – 100 баллов – «отлично»

• 70 – 75 баллов – «хорошо»

• 51 – 69 баллов – «удовлетворительно

• мене 51 балла – «неудовлетворительно». Баллы учитываются в процессе текущей оценки знаний программного материала. Раздел 2. Разработка программного обеспечения Задание 1. Перечень контрольных вопросов по разделу

1. Понятия изобретения, полезной модели, промышленного образца.

**2. Структура ПО.**

3. Проектирование и дизайн интерфейсов.

4. Модульное программирование.

**5. Структурное программирование.**

6. Объектно-ориентированное программирование.

7. Инструменты разработки программных средств.

8. Организация работ при коллективной разработке программных продуктов. Тестовое задание Список тестовых вопросов

1.

1. В каких единицах можно измерить надежность разработанного программного продукта:

2. отказов/час

3. км/час

4. Кбайт/сек

5. операций/сек

2. В каких единицах можно измерить быстродействие ПО:

1. отказов/час

2. км/час

3. Кбайт/сек

4. операций/сек

3. Что относится к этапу программирования:

1. написание кода программы

2. разработка интерфейса

3. работоспособность

4. анализ требований.

4. Укажите правильную последовательность этапов программирования:

1. компилирование, компоновка, отладка

2. компоновка, отладка, компилирование

3. отладка, компилирование, компоновка

4. компилирование, отладка, компоновка.

5. К инструментальным средствам программирования относятся:

1. компиляторы, интерпретаторы

2. СУБД (системы управления базами данных)

3. BIOS (базовая система ввода-вывода)

4. ОС (операционные системы). 6 Доступ, при котором записи файла читаются в физической последовательности, называется:

1. прямым

2. простым

3. последовательным

4. основным

7. Какого метода программирования не существует:

1. логического

2. структурного

3. модульного

4. объектно-ориентированного

8. Какой этап выполняется раньше остальных:

1. разработка алгоритма

2. выбор языка программирования

3. написание исходного кода

4. компиляция

8. Наличие комментариев позволяет:

1. быстрее писать программы

2. быстрее выполнять программы

3. быстрее найти ошибки в программе

4. быстрее произвести описание структуры программы

8. Что определяет выбор языка программирования:

1. область приложения

2. знание языка

3. наличие дополнительных библиотек

4. особенности структуры

8. Для каких задач характерно использование большого количества исходных данных, выполнение операций поиска, группировки:

1. для экономических задач

2. для системных задач

3. для инженерных задач

4. для математический

8. На каком этапе производится выбор языка программирования:

1. проектирование

2. программирование

3. отладка

4. тестирование.

8. Когда приступают к тестированию программы:

1. когда программа уже закончена

2. после постановки задачи

3. на этапе программирования

4. на этапе проектирования

8. Одним из методов автоматизации программирования является:

1. структурное программирование

2. модульное программирование

3. визуальное программирование

4. объектно-ориентированное программирование.

8. Критерием оптимизации программы является:

1. быстродействие или размер программы

2. быстродействие и размер программы

3. надежность или эффективность

4. надежность и эффективность Практические работы Перечень тем практических работ:

4. Анализ выбранного стиля программирования

5. Разработка проекта программного обеспечения

6. Разработка структурного алгоритма

7. Разработка программного продукта с использованием объектно-ориентированного программирования

8. Разработка справочной системы

9. Тестирование методом «белого ящика»

10. Тестирование методом «черного ящика»

11. Способы анализа граничных решений

12. Способы диаграмм причин-следствий

13. Нисходящее тестирование интеграций

14. Восходящее тестирование интеграций

15. Анализ предметной области

16. Автоматизированное тестирование

17. Отладка программ

18. Оптимизация программ

19. Работа в составе бригады Рефераты Перечень тем рефератов:

1. Показатели эффективности программного продукта.

2. Ручная и автоматизированная отладка.

3. Синтаксическая и семантическая отладка.

4. Разрушающая и неразрушающая отладка.

5. Проектирование программных модулей.

6. Кодирование программных модулей.

7. Модульное интеграционное тестирование. Тестирование пользовательского интерфейса.

8. Системное интеграционное тестирование.

9. Классы эквивалентности и граничные условия.

10. Тестирование переходов между состояниями.

11. Нагрузочные испытания.

12. Тестирование функциональной эквивалентности.

13. Регрессионное тестирование.

14. Методы обеспечения надежности на различных этапах жизненного цикла

разработки программного обеспечения.

15. Прогнозирование ошибок.

16. Предотвращение ошибок.

17. Обеспечение отказоустойчивости.

18. Построение диаграммы распределения работников по этапам.

19. Оценка качества программного продукта.

20. Программные средства защиты программных продуктов.

21. Правовые методы защиты программных продуктов.

22. Лицензирование программных продуктов.

23. Экономические аспекты создания и использования программных средств. Итоговой формой контроля сформированности компетенций у студентов по дисциплине является экзамен. ФОС промежуточной аттестации состоит из вопросов и задач к экзамену по дисциплине. Оценивание студента на экзамене Оценка 5 «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал различной литературы, правильно обосновывает принятое нестандартное решение, владеет разносторонними навыками и приемами выполнения практических задач по формированию общепрофессиональных компетенций. Оценка 4 «хорошо» выставляется студенту, если он грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения, а так же имеет достаточно полное представление о значимости знаний по дисциплине. Оценка 3 «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает сложности при выполнении практических работ и затрудняется связать теорию вопроса с практикой. Оценка 2 «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, неуверенно отвечает, допускает серьезные ошибки, не имеет представлений по методике выполнения практической работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по данной дисциплине. Вопросы к билетам

1. Характеристика программы и программного обеспечения

2. Характеристика задач и приложений

3. Технологические и функциональные задачи

4. Процесс создания программ

5. Постановка задачи

6. Алгоритмизация

7. Характеристика программного продукта и его специфика

8. Классификация программных продуктов

9. Каскадная модель жизненного цикла

10. Поэтапная модель жизненного цикла с промежуточным контролем

11. Спиральная модель жизненного цикла

12. Основные процессы жизненного цикла

13. Вспомогательные процессы жизненного цикла

14. Организационные процессы жизненного цикла

15. Качество программного продукта

16. Классы прочности модуля

17. Критерии качества программных продуктов

18. Характеристики качества программного средства

**19. Общие характеристики качества программных систем**

20. Методы управления качеством, используемые в современных технологиях программирования

21. Аттестация программных систем

22. Функциональные требования к программной системе

23. Нефункциональные требования к программной системе

24. Методы первичного сбора требований

25. Анализ требований

26. Что такое техническое задание? Для чего оно нужно?

27. Внутренняя организация программного обеспечения

28. Методологии разработки программного обеспечения

29. Стадии и этапы проектирования

30. Автоматизированное проектирование алгоритмов и программ

31. Неавтоматизированное проектирование алгоритмов и программ

32. Метод структурного проектирования: цель, принцип, преимущество

33. Принципы системного проектирования

34. Нисходящее проектирование программ

35. Принципы модульного проектирования

36. Объектно-ориентированное программирование

37. Объектно-ориентированные языки программирования, их характеристики

38. Проектирование пользовательского интерфейса

39. Определение «кодирование», его цель.

40. Модульное программирование

**41. Структурное программирование**

42. Разработка справочной системы программного обеспечения, основные характеристики

43. Понятие и сущность тестов

44. Виды тестов и сфера их применения

45. Программная ошибка

**46. Структурное тестирование**

47. Тестирование программ методом «белого ящика»

48. Тестирование программ методом «черного ящика»

49. Восходящее тестирование программ

50. Нисходящее тестирование программ

51. Методы функционального тестирования

52. Метод эквивалентного разбиения

53. Метод анализа граничных значений

54. Метод тестирования таблицы решений

55. Комплексное тестирование программ

56. Отладка программ

57. Сопровождение программ

58. Принципы коллективной разработки программ

59. Методы коллективной разработки программ

60. Организация коллективной разработки программистов Типовые задания практической части МДК.01.01 Разработать техническое задание по теме Вашего дипломного проекта. Примеры:

1. Модуль управления персонажем.

2. Система инвентаря игрового персонажа.

3. Система сохранения и загрузки игрового состояния.

4. Модуль диалогов персонажей.

5. Система игровых заданий.

6. Модуль настроек игры.

7. Механика подбора и использования предметов.

8. Система очков и достижений.

9. Модуль управления камерой.

10. Система обработки пользовательского ввода.

11. Модуль поведения неигрового персонажа.

12. Система переходов между игровыми сценами.

13. Модуль звукового сопровождения игровых событий.

14. Система локализации текстов игры.

15. Модуль журнала игровых событий.

16. Механика взаимодействия с объектами виртуальной сцены.

17. Модуль пользовательского интерфейса прототипа.

18. Модуль проверки условий победы и поражения.

19. Система тестирования игровой механики.

20. Модуль управления состояниями игрового приложения.

МДК.01.02 Оптимизация программного кода Вопросы теоретической части Раздел 1. Общая характеристика инструментальных средств разработки программ Задание 1. Перечень контрольных вопросов по разделу

1. Инструментальные среды программирования.

2. Понятие компьютерной технологии разработки.

3. Визуальное проектирование.

4. Delphi. Основные характеристики продукта.

**5. Структура среды программирования.**

6. Стандартные компоненты.

7. Понятие и структура проекта.

8. Базовые конструкции среды Delphi.

9. Средства тестирования и отладки.

10. Основные свойства ООП в Delphi.

11. Методы в Delphi.

12. Основы работы с базами данных в Delphi. Тестовые задания к разделу

1. Программное средство, предназначенное для поддержки разработки других программ, называется -… А) аппаратным инструментом Б) программным инструментом В) программной средой Г) инструментарий технологии программирования

2. Анализаторы обеспечивают…

1) конструирование тех или иных программных продуктов и документов на различных этапах жизненного цикла

2) автоматически приводить документы к другой форме представления или переводить документ одного вида к документу другого вида

3) статическую обработку документов, осуществляя различные виды их контроля, выявление определенных их свойств и накопление статистических данных, либо динамический анализ программ

4) выполнять на компьютере описание процессов или отдельных их частей, представленных в виде, отличном от машинного кода

3. Преобразователи обеспечивают…

1) конструирование тех или иных программных продуктов и документов на различных этапах жизненного цикла

2) автоматически приводить документы к другой форме представления или переводить документ одного вида к документу другого вида

3) статическую обработку документов, осуществляя различные виды их контроля, выявление определенных их свойств и накопление статистических данных, либо динамический анализ программ

4) выполнять на компьютере описание процессов или отдельных их частей, представленных в виде, отличном от машинного кода

4. Сколько классов инструментальных средств выделяют в инструментальной среде разработки и сопровождения программ?

1) 2

2) 4

3) 3

4) 5

5. Среда программирования предназначена для…

1) конструирования тех или иных программных продуктов и документов на различных этапах жизненного цикла

2) автоматического перевода документов к другой форме представления или перевода документа одного вида к документу другого вида

3) поддержки ранних этапов разработки программ и автоматической генерации программ по спецификациям

4) поддержки процессов программирования (кодирования), тестирования и отладки программ

6. Инструментальные среды программирования бывают

1) языково-ориентированные среды и среды общего назначения

2) объектно-ориентированные и языково-ориентированные среды

3) среды общего назначения и прикладные среды

4) среды общего назначения, прикладные среды, логические и математические среды

7. Для поддержки разработки программного продукта на каком-либо одном языке программирования используют…

1) среду программирования общего назначения

2) языково-ориентированную среду программирования

3) интерпретирующую среду программирования

4) прикладную среду программирования

8. Синтаксически-управляемая инструментальная среда программирования базируется на знании

1) семантики языка программирования

2) синтаксиса языка программирования

3) синтаксиса и семантики языка программирования

4) основных управляющих структур языка программирования

9. На рисунке представлена классификация

1) инструментальной системы технологии программирования

2) инструментальной среды разработки и сопровождения программ

3) рабочего места компьютерной технологии

4) языков программирования

10. Инструментальная система технологии программирования – это…

1) программное средство, предназначенное для поддержки разработки других программ

2) устройство компьютера, специально предназначенное для поддержки разработки программного средства

3) интегрированная совокупность программных и аппаратных инструментов, поддерживающая все процессы разработки и сопровождения больших программных продуктов

4) логически связанная совокупность программных и аппаратных инструментов, поддерживающих разработку ПП

11. Устройство компьютера, специально предназначенное для поддержки разработки программного средства, называется -…

1) аппаратным инструментом

2) программным инструментом

3) программной средой

4) инструментарий технологии программирования

12. Редакторы обеспечивают…

1) конструирование тех или иных программных продуктов и документов на различных этапах жизненного цикла

2) автоматически приводить документы к другой форме представления или переводить документ одного вида к документу другого вида

3) статическую обработку документов, осуществляя различные виды их контроля, выявление определенных их свойств и накопление статистических данных, либо динамический анализ программ

4) выполнять на компьютере описание процессов или отдельных их частей, представленных в виде, отличном от машинного кода

13. Инструменты, поддерживающие процесс выполнения программ, обеспечивают…

1) конструирование тех или иных программных продуктов и документов на различных этапах жизненного цикла

2) автоматический привод документов к другой форме представления или перевод документа одного вида к документу другого вида

3) возможность выполнять на компьютере описание процессов или отдельных их частей, представленных в виде, отличном от машинного кода

4) статическую обработку документов, осуществляя различные виды их контроля, выявление определенных их свойств и накопление статистических данных, либо динамический анализ программ

14. Инструментальная система технологии программирования предназначена для…

1) поддержки всех процессов разработки и сопровождения в течение всего жизненного цикла ПС и ориентирована на коллективную разработку больших программных систем с длительным жизненным циклом

2) автоматического перевода документов к другой форме представления или перевода документа одного вида к документу другого вида

3) поддержки ранних этапов разработки программ и автоматической генерации программ по спецификациям

4) поддержки процессов программирования (кодирования), тестирования и отладки программ

15. Рабочее место компьютерной технологии предназначено для…

1) конструирования тех или иных программных продуктов и документов на различных этапах жизненного цикла

2) автоматического перевода документов к другой форме представления или перевода документа одного вида к документу другого вида

3) поддержки ранних этапов разработки программ и автоматической генерации программ по спецификациям

4) поддержки процессов программирования (кодирования), тестирования и отладки программ

16. Инструментальные среды программирования содержат

1) редактор, анализатор и компилятор

2) редактор, интерпретатор и компилятор

3) интерпретатор, компилятор, преобразователь

4) редактор и интерпретатор

17. Для поддержки разработки программного продукта на разных языках программирования (например, текстовый редактор, редактор связей или интерпретатор языка целевого компьютера) используют…

1) среду программирования общего назначения

2) языково-ориентированную среду программирования

3) интерпретирующую среду программирования

4) прикладную среду программирования

18. На рисунке представлена классификация

1) инструментальной системы технологии программирования

2) инструментальной среды программирования

3) рабочего места компьютерной технологии

4) языков программирования

19. При использовании компьютерных технологий для разработки ПП жизненный цикл ПП представлен следующей цепочкой:

1) прототипирование – кодогенерация – комплексная отладка и тестирование – аттестация, применение, сопровождение

2) прототипирование – разработка спецификаций – автоматизированный контроль спецификаций – кодогенерация – комплексная отладка и тестирование – аттестация, применение, сопровождение

3) разработка спецификаций – автоматизированный контроль спецификаций – кодогенерация – комплексная отладка и тестирование – аттестация, применение, сопровождение

4) прототипирование – разработка спецификаций – кодогенерация – аттестация, применение, сопровождение

20. Основными чертами инструментальной системы технологии программирования являются…

1) массовость, дискретность, результативность, определенность, понятность

2) комплексность, ориентированность на коллективную разработку, технологическая определенность, интегрированность

3) актуальность, непротиворечивость, полнота

4) комплексность, актуальность, интегрированность, массовость, понятность Шкала оценки тестовых вопросов: «2» – от 0 до 40% «3» – от 41% до 60% «4» – от 61% до 80% «5» – от 81% до 100% Практические работы Перечень тем практических работ:

1. Разработка программных модулей

2. Проектирование пользовательского интерфейса

3. Разработка пользовательского интерфейса Шкала оценки практических работ: оценка «5» ставится, если:

- учащийся самостоятельно выполнил все этапы решения задач на ЭВМ

- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы; оценка «4» ставится, если:

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с ЭВМ в рамках поставленной задачи

- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок

- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи. оценка «3» ставится, если:

- работа выполнена не полностью, допущено более трех ошибок, но учащийся владеет основными навыками работы на ЭВМ, требуемыми для решения поставленной задачи. оценка «2» ставится, если:

- допущены существенные ошибки, показавшие, что учащийся не владеет обязательными знаниями, умениями и навыками работы на ЭВМ или значительная часть работы выполнена не самостоятельно. Перечень тем рефератов:

1. Инструментальные средства поддержки процесса управления требованиями.

2. Инструментальные средства поддержки процесса разработки проекта.

3. Инструментальные средства реализации кода.

4. Инструментальные средства тестирования.

5. Инструментальные средства поддержки процесса управления конфигурациями.

6. Семейство стандартов моделирования IDEF. Структурные карты. Диаграммы переходов состояний.

7. Инструментальные средства поддержки методологий функционального моделирования.

8. Определение, характеристики и основные элементы процессного подхода.

9. Процессный подход к моделированию деятельности.

**10. Общие сведения, функциональное назначение методологии ARIS.**

11. Архитектура ARIS - пять типов представлений, отражающих основные аспекты деятельности организации.

12. Базовая модель ARIS - этапы описания бизнес-процессов.

13. Виды моделей методологии ARIS - основные принципы построения, структура свойства, составляющие элементы.

14. Использование методологии ARIS в различных областях деятельности. Раздел 2. Применение CASE-средств Задание 1. Перечень контрольных вопросов по разделу

1. Принципы работы с объектами.

2. Использование библиотек, процедур и функций.

3. Состав интегрированной среды разработки VBA.

4. Типы данных в VBA.

5. Константы VBA.

6. Операции языка VBA.

7. Управляющие структуры для организации циклов в VBA.

8. Основные принципы работы в среде VBA.

9. Отличительные особенности сред программирования VBA и Delphi.

10. Инструментальные средства разработки ПО.

11. Классификация инструментальных средств участвующих в разработке приложения. Тестовое задание Список тестовых вопросов

1. Современные крупные проекты информационных систем характеризуются следующими особенностями:

1) сложность описания, требующая тщательного моделирования и анализа данных и процессов

2) наличие совокупности тесно взаимодействующих компонентов

3) наличие прямых аналогов, ограничивающее возможность использования каких-либо типовых проектных решений

4) невозможность интеграции существующих и вновь разрабатываемых приложений

2. Под CASE-средства понимаются программные средства, поддерживающие…

1) процессы создания и сопровождения ИС, включая анализ и формулировку требований, проектирование прикладного ПО

2) процессы тиражирования программного продукта

3) процессы создания и эксплуатации программного продукта

4) процессы компилирования и интерпретации программных продуктов

3. Репозитарий Case – средства – это…

1) специализированная база данных проекта, предназначенная для отображения состояния проектируемой системы в каждый момент времени

2) компонент, обеспечивающий создание и редактирование в интерактивном режиме элементов диаграмм и связей между ними

3) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования

4) компонент, позволяющий получать информацию о проектах в виде отчетов

5) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта

6) набор системных утилит по обслуживанию репозитария

4. Графический редактор Case – средства – это…

1) компонент, обеспечивающий создание и редактирование в интерактивном режиме элементов диаграмм и связей между ними

2) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования

3) компонент, позволяющий получать информацию о проектах в виде отчетов

4) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта

5. Верификатор Case – средства – это…

1) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования

2) компонент, позволяющий получать информацию о проектах в виде отчетов

3) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта

4) набор системных утилит по обслуживанию репозитария

6. Документатор проекта Case – средства – это…

1) компонент, позволяющий получать информацию о проектах в виде отчетов

2) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта

3) набор системных утилит по обслуживанию репозитария

4) компонент, обеспечивающий создание и редактирование в интерактивном режиме элементов диаграмм и связей между ними

7. Сервис Case – средства – это…

1) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования

2) компонент, позволяющий получать информацию о проектах в виде отчетов

3) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта

4) набор системных утилит по обслуживанию репозитария

8. Администратор проекта Case – средства – это…

1) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования

2) компонент, позволяющий получать информацию о проектах в виде отчетов

3) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта

4) набор системных утилит по обслуживанию репозитария

9. Какие методологии проектирования используют Case – средства?

1) структурного и модульного проектирования

2) структурного и объектно-ориентированного проектирования

3) объектно-ориентированного и нисходящего проектирования

4) нисходящего и восходящего проектирования

**10. Структурное проектирование системы основано на…**

1) объектно-ориентированной декомпозиции

2) алгоритмической декомпозиции

3) модульной декомпозиции

4) функциональной декомпозиции

11. Объектно-ориентированное проектирование системы основано на…

1) объектно-ориентированной декомпозиции

2) алгоритмической декомпозиции

3) модульной декомпозиции

4) функциональной декомпозиции

12. Case – средства представляют собой…

1) набор инструментальных средств для проектирования программного продукта

2) набор программных средств для сопровождения программного продукта

3) набор программных и инструментальных средств, поддерживающие все процессы жизненного цикла программного продукта

4) набор аппаратных средств, поддерживающих все процессы жизненного цикла программного продукта

13. Компания-разработчик приобрела новое Case – средство. Сразу ли компания получит ожидаемый результат от применения новой технологии?

1) да

2) нет

14. Сколько классов Case – средств выделяют?

1) 5

2) 3

3) 7

4) 2

15. Case – средства анализа и проектирования, предназначенные для

1) моделирования данных и генерации схем баз данных

2) построения и анализа моделей деятельности организаций (предметной области) или моделей проектируемой системы

3) обеспечения комплексной поддержки требований к создаваемой системе

4) поддержки всего жизненного цикла программного продукта

16. Case – средства управления требованиями предназначены для

1) моделирования данных и генерации схем баз данных

2) построения и анализа моделей деятельности организаций (предметной области) или моделей проектируемой системы

3) обеспечения комплексной поддержки требований к создаваемой системе

4) поддержки всего жизненного цикла программного продукта

17. Case – средства проектирования баз данных предназначены для

1) моделирования данных и генерации схем баз данных

2) построения и анализа моделей деятельности организаций (предметной области) или моделей проектируемой системы

3) обеспечения комплексной поддержки требований к создаваемой системе

4) поддержки всего жизненного цикла программного продукта

18. На каких стандартах базируется технология освоения и внедрения Case – средств?

**1) ГОСТ 2107-99**

2) IEEE (IEEE Std 1348-1995 и IEEE Std 1209-1992)

**3) AES**

**4) ISO**

19. Из каких этапов состоит процесс освоения и внедрения Case – средств?

1) определение потребностей в CASE-средствах, оценка и выбор CASE-средств, практическое внедрение CASE-средств

2) определение потребностей в CASE-средствах, оценка и выбор CASE-средств, выполнение пилотного проекта, практическое внедрение CASE-средств

3) определение потребностей в CASE-средствах, проектирования CASE-средств, практическое применение CASE-средств

4) проектирование CASE-средств, оценка и внедрение CASE-средств, практическое применение CASE-средств

20. Критериями для выбора CASE-средств могут являться

1) открытая архитектура, поддержка полного жизненного цикла ИС с обеспечением эволюционности ее развития, обеспечение целостности проекта, независимость от программно-аппаратной платформы и СУБД

2) модифицируемость, простота, эффективность, учет человеческого фактора, многоплатформенность

3) закрытая архитектура, поддержка полного жизненного цикла ИС с обеспечением эволюционности ее развития, простота, эффективность

4) максимальная зависимость от программных и аппаратных средств системы и характеристик самой системы, жесткая привязка к конкретным информационным процессам, прочность внутренней связи отдельных компонентов системы

21. Комплексность компьютерной поддержки разработки ПП с использованием инструментальной системы технологии программирования означает

1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана

2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов

3) что все инструменты объединены единым пользовательским интерфейсом

4) что инструменты действуют в соответствии с фиксированной информационной схемой системы, определяющей зависимость различных используемых в системе фрагментов данных друг от друга

22. Ориентированность инструментальной системы технологии программирования на коллективную разработку означает

1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана

2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов

3) что все инструменты объединены единым пользовательским интерфейсом

4) что инструменты действуют в соответствии с фиксированной информационной схемой системы, определяющей зависимость различных используемых в системе фрагментов данных друг от друга

23. Технологическая определенность инструментальной системы технологии программирования означает

1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана

2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов

3) что ее комплексность ограничивается рамками какой-либо конкретной технологии программирования

4) что инструменты действуют в соответствии с фиксированной информационной схемой системы, определяющей зависимость различных используемых в системе фрагментов данных друг от друга

24. Интегрированность инструментальной системы технологии программирования по данным означает

1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана

2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов

3) что ее комплексность ограничивается рамками какой-либо конкретной технологии программирования

4) что инструменты действуют в соответствии с фиксированной информационной схемой системы, определяющей зависимость различных используемых в системе фрагментов данных друг от друга

45. Интегрированность инструментальной системы технологии программирования по пользовательскому интерфейсу означает

1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана

2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов

3) что ее комплексность ограничивается рамками какой-либо конкретной технологии программирования

4) что все инструменты объединены единым пользовательским интерфейсом Практические работы Перечень тем практических работ:

4. Анализ предметной области

5. Разработка UML диаграмм

6. Работа с инструментальными средствами, поддерживающими методологию

объектно-ориентированного моделирования.

7. Работа с CASE – средствами проектирования программного обеспечения

8. Работа с CASE – средствами кодирования программного обеспечения

9. Работа с CASE – средствами тестирования программного обеспечения

10. Инструментальные средства поддержки процесса управления требованиями.

11. Инструментальные средства поддержки процесса разработки проекта.

12. Инструментальные средства реализации кода.

13. Инструментальные средства тестирования.

14. Инструментальные средства поддержки процесса управления конфигурациями.

**15. Структурные карты. Вопросы к экзамену**

1. Необходимые инструментальные средства разработки программ

2. Часто используемые инструментальные средства разработки программ

3. Специализированные инструментальные средства разработки программ

4. Интегрированные среды разработки

5. Средства разработки программного обеспечения

6. Определение «разработка программ»

7. Три этапа разработки программ

8. Средства проектирования приложений

9. Средства реализации программного кода

10. Средства тестирования программ

11. Классы инструментальных средств разработки программ

12. Четыре категории инструментальных программ, применяемые при проектировании экспертных систем

13. Оболочки экспертных систем

14. Языки программирования высокого уровня

15. Среда программирования, поддерживающая несколько парадигм

16. Дополнительные модули

17. Языки инженерии знаний

18. Язык логического программирования Prolog

19. Язык функционального программирования Lisp

20. Средства автоматизации разработки экспертных систем

**21. Общее программное обеспечение**

22. Специальное программное обеспечение

23. Инструментальная система технологии программирования

24. Четыре класса компьютерной поддержки инструментальных систем технологий программирования

25. Комплексность

26. Ориентированность на коллективную разработку

27. Технологическая определенность

28. Интегрированность

29. Компоненты инструментальных систем технологий программирования

30. База данных разработки

31. Инструментарий

32. Интерфейсы

**33. Общая архитектура инструментальных систем технологий программирования**

34. Инструментальная система поддержки проекта

35. Языково-зависимая инструментальная система

36. Пользовательский интерфейс

37. Схема организации взаимодействия компьютера и пользователя

38. Процедурно-ориентированный подход к разработке интерфейсов

39. Объектно-ориентированный подход к разработке интерфейсов

40. Типы интерфейсов

41. Интерфейс-меню

42. Интерфейсы со свободной навигацией

43. Критерии оценки интерфейса пользователем

44. Факторы появления Case-технологий

45. Что такое Case-технология?

46. Компоненты интегрированного Case-средства

47. Классификация по категориям Case-средств

48. Классификация по типам Case-средств

49. Вспомогательные типы Case-средств

**50. Структурный подход к разработке ИС**

51. Объектно-ориентированный подход к разработке ИС

52. Факторы, усложняющие определение возможного эффекта от использования Case-средств

53. Качества организации для успешного внедрения Case-средств

54. Проблемы использования Case-средств

55. Факторы появления Case-технологий

**56. Структурный системный анализ**

57. Диаграммы «сущность-связь»

58. Диаграммы классов

59. Язык графического описания UML

60. Диаграмма компонентов

61. Диаграмма композитной структуры

62. Диаграмма развёртывания

63. Диаграмма объектов

64. Диаграмма пакетов

65. Диаграмма деятельности

66. Преимущества UML

**67. IDEF**

68. Диаграммы переходов состояний

69. Методология функционального моделирования ИС

70. Состав функциональной модели

71. Иерархия диаграмм

72. Типы связей между функциями

73. Характеристика современных Case-средств

74. Методология ARIS

75. Программный продукт ARIS Express

76. Основные элементы, используемые в нотации ARIS

77. Архитектура ARIS

78. Имитационное моделирование

79. Применение имитационного моделирования

80. Виды имитационного моделирования

81. Дискретно-событийное моделирование

82. Системная динамика

83. Области применения имитационного моделирования

84. Основные этапы компьютерного моделирования

85. Построение концептуальной модели системы

86. Постановка задачи машинного моделирования

87. Анализ задачи моделирования

88. Определение требований к исходной информации

89. Выдвижение гипотез и принятие предположений

90. Определение параметров и переменных

91. Установление основного содержания модели

92. Обоснование критериев оценки эффективности системы

93. Определение процедур аппроксимации

94. Описание концептуальной модели

95. Проверка достоверности модели

96. Составление технической документации МДК.01.03 Интеграция программных модулей и платформ/фреймворков Типовое контрольное задание (тест)

1) Модель это:

1) новый объект (реальный, информационный или воображаемый), отличный от исходного, который обладает существенными для целей моделирования свойствами. И в рамках этих

целей полностью заменяет исходный объект.

2) некоторое упрощенное подобие реального объекта.

3) физический или информационный аналог объекта, функционирование которого по определенным параметрам подобно функционированию реального объекта

2) Аспектами моделирования могут выступать: a. Внешний вид объекта. b. Назначение объекта. c. Структура объекта. d. Поведение объекта.

3) Выберите вариант, где перечислены виды модели по способу представления: a. Материальные, воображаемые, информационные. b. Статические, динамические. c. Модели внешнего вида, структуры, поведения.

4) Какие виды моделей не относятся к информационным. a. Дескриптивные. b. Наглядные. c. Коммуникативные. d. Детерминированные.

5) Укажите варианты, соответствующий действительности: a. Модель зависит от целей моделирования. b. Модель не зависит от целей моделирования. c. Одному реальному объекту, может соответствовать несколько различных моделей. d. Одна модель может соответствовать нескольким реальным объектам.

6) Укажите свойства, присущие учебным компьютерным моделям. a. Наглядность. b. Динамичность. c. Интерактивность. d. Простота в управлении.

8) Какие этапы в себя включает системный подход: a. дедукция. b. индукция. c. декомпозиция. d. синтез.

9) дайте определение «Система» a. – это множество прямо или косвенно взаимосвязанных элементов. b. – это некоторая совокупность элементов, существующих раздельно. c. – это множество объектов, которые не оказывают влияния друг на друга.

10) Объект, о котором ничего не известно (I =0) называется: a. «Белым ящиком». b. «Серым ящиком». c. «Черным ящиком».

11) В основном тезисе формализации говорится следующее: a. суть объекта не меняется от того, как мы его назовем. b. суть объекта меняется в зависимости от его названия. c. суть объекта зависит от его названия

12) Знак – это: a. элемент конечного множества, отличных друг от друга элементов.

b. совокупность элементов, используемых для обозначения объекта.

13) Язык характеризуется: a. Набором используемых знаков. b. Правилами образования из этих знаков различных языковых конструкций. c. Набором синтаксических, семантических правил использования языковых конструкций.

14) Формальным представлением текстовой информации является. a. Содержание книги b. Бланк c. Рассказ

16) Перечислите виды графов

17) Граф является наиболее удобной формой моделирования: a. структуры объекта. b. поведения объекта. c. внешнего вида объекта.

18) Приведите примеры моделей в форме древовидной структуры

19) Любая модель строится для: a. получения нового объекта b. получения сведений о реальном объекте c. исследования реального объекта.

20) Адекватность модели предполагает: a. воспроизведение моделью всех характеристик реального объекта существенных для модели. b. Воспроизведение моделью характеристик,

существенных для целей моделирования. c. Воспроизведение моделью с необходимой точностью всех характеристик, существенных для целей моделирования.

21) Какие виды подобий различают по адекватности природы объектов: a. Физическое b. Математическое c. Натурное d. Символьное

22) Моделирование это: a. метод (или процесс) создания модели по образу и подобию оригинала b. процесс создания нового объекта, который называется моделью. c. метод (или процесс) изучения свойств объектов - оригиналов посредством исследования соответствующих свойств их моделей Вопросы к экзамену

1) Понятие «модель»

2)Натурные и абстрактные модели.

3)Виды моделирования в естественных и технических науках.

4)Абстрактные модели и их классификация.

5)Компьютерная модель.

6)Вербальные модели.

7)Информационные модели.

8)Объекты и их связи.

9)Основные структуры в информационном моделировании.

10)Примеры информационных моделей.

11)Математические модели.

12)Различные подходы к классификации математических моделей.

13)Геометрическое моделирование и компьютерная графика.

14)Модели с сосредоточенными и распределенными параметрами.

15)Дескриптивные, оптимизационные, многокритериальные, игровые модели.

16)Системный подход в научных исследованиях.

17)Численный эксперимент. Его взаимосвязи с натурным экспериментом и теорией.

18)Достоверность численной модели.

19)Анализ и интерпретация модели.

20)Имитационные модели и системы.

21)Область и условия применения.

22)Этапы построения имитационной модели.

23)Модель популяции.

24)Моделирование систем массового обслуживания.

25)Примеры математических моделей в химии, биологии, экологии, экономике.

26)Учебные компьютерные модели.

27)Программные средства для моделирования предметно-коммуникативных сред (предметной области).

28)Специфика использования компьютерного моделирования в педагогических программных средствах.

29)Метод статистических испытаний