

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Игнатенко Виталий Иванович

Должность: Проректор по образовательной деятельности и молодежной политике

Дата подписания: 25.06.2024 10:29:52

Уникальный программный идентификатор:

a49ae343af5448d45d7e3e1e499659da8109ba78

**Министерство науки и высшего образования РФ**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Заочный государственный университет им. Н.М. Федоровского»**

**ЗГУ**

## **ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

**по дисциплине**

**«Технологии коллективной разработки программных систем»**

Факультет электроэнергетики, экономики и управления (ФЭЭ и У)

Направление подготовки: 09.04.03 «Прикладная информатика

Направленность (профиль): Информационные системы и технологии в бизнесе

Уровень образования: магистратура

Кафедра информационных систем и технологий

*наименование кафедры*

Разработчик ФОС:

доцент, к.э.н.

(должность, степень, ученое звание)

(подпись)

И.С. Беляев

(ФИО)

Оценочные материалы по дисциплине рассмотрены и одобрены на заседании кафедры ИСиТ

И.о.зав. кафедрой \_\_\_\_\_ И.С. Беляев

Фонд оценочных средств по дисциплине «Технологии коллективной разработки программных систем» для текущей/промежуточной аттестации разработан в соответствии с Федеральным государственным образовательным стандартом высшего образования по направлению подготовки 09.04.03 «Прикладная информатика» на основе Рабочей программы дисциплины «Технологии коллективной разработки программных систем», утвержденной решением ученого совета № 12-3 от 28.04.2023, Положения о формировании Фонда оценочных средств по дисциплине (ФОС), Положения о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся ЗГУ, Положения о государственной итоговой аттестации (ГИА) выпускников по образовательным программам высшего образования в ЗГУ.

**1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами образовательной программы**

Таблица 1 – Компетенции и индикаторы их достижения

Код и наименование компетенции	Индикаторы достижения и планируемые результаты обучения по дисциплине
<b>Универсальные</b>	
«УК»	-
<b>Общепрофессиональные</b>	
«ОПК»	-
<b>Профессиональные</b>	
ПК-4: Способен управлять инфраструктурой коллективной среды разработки	ПК-4.1: Выбирает и применяет практики и методологии управления разработкой ПО

Таблица 2. – Паспорт фонда оценочных средств

Контролируемые разделы (темы) дисциплины	Формируемая компетенция	Наименование оценочного средства	Форма оценивания
Тема 1. Введение. Проектирование и коллективная разработка программных продуктов. Риски	ПК-4.1	Тестовые задания	Письменно
Тема 2. Гибкая методология разработки программного обеспечения. Web-сервисы для коллективной разработки	ПК-4.1	Тестовые задания	Письменно
Тема 3. Интегрированные среды разработки программных продуктов. Организация коллективной разработки	ПК-4.1	Тестовые задания	Письменно
Тема 4. Методы тестирования программных компонент. Оптимизация производительности	ПК-4.1	Тестовые задания	Письменно
Тема 5. Коллективная разработка прикладных решений на платформе 1С: Предприятие	ПК-4.1	Контрольные вопросы	Устный опрос

## 2. Перечень контрольно-оценочных средств (КОС)

Для определения качества освоения обучающимися учебного материала по дисциплине используются следующие контрольно-оценочные средства текущего контроля успеваемости, промежуточной аттестации обучающихся:

Таблица 3. Перечень контрольно-оценочных средств

	Наименование оценочного средства	Сроки выполнения	Шкала оценивания	Критерии оценивания
1.	<i>Текущий контроль качества</i>			
1	Контрольные вопросы	2 семестр	Достигнут/ не достигнут пороговый уровень освоения компетенции	Зачтено/ не зачтено
	<i>Промежуточная аттестация</i>			
1	Тестовые задания	2 семестр	Освоил/ не освоил компетенцию*	Зачтено/ не зачтено
2	Экзаменационные билеты	2 семестр	от 2 до 5 баллов	Критерии оценивания приведены ниже
	<p><b>Критерии оценивания тестовых заданий «зачтено», «не зачтено»:</b></p> <ul style="list-style-type: none"> <li>- «Зачтено» выставляется обучающемуся, если он показал достаточно прочные знания основных положений учебной дисциплины, умение самостоятельно решать конкретные практические задачи, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты.</li> <li>- «Не зачтено» выставляется обучающемуся, если при ответе выявились существенные пробелы в знаниях основных положений учебной дисциплины, неумение с помощью преподавателя получить правильное решение конкретной практической задачи из числа предусмотренных рабочей программой.</li> </ul>			
	<p><b>Критерии оценивания для экзамена по 4-балльной шкале:</b></p> <ul style="list-style-type: none"> <li>- оценки «отлично» заслуживает обучающийся, обнаруживший всесторонние, глубокие знания учебного материала и умение свободно выполнять задания, предусмотренные программой; изучивший основную литературу и знакомый с дополнительной литературой, рекомендованной программой обучения; безупречно отвечавший не только на вопросы билета, но и на</li> </ul>			

<p>дополнительные вопросы; проявивший творческие способности в использовании учебного материала;</p> <ul style="list-style-type: none"> <li>- оценки «хорошо» заслуживает обучающийся, обнаруживший полные знания учебного материала, успешно выполнивший предусмотренные программой задания, изучивший основную литературу, отвечавший на все вопросы билета;</li> <li>- оценки «удовлетворительно» заслуживает обучающийся, обнаруживший знания в объёме, необходимом для дальнейшей учёбы и работы по профессии, справившийся с выполнением заданий, знакомый с основной литературой, допустивший погрешности в ответе и при выполнении заданий, но обладающий достаточными знаниями для их устранения под руководством преподавателя;</li> <li>- оценка «неудовлетворительно» выставляется обучающемуся, обнаружившему пробелы в знаниях основного учебного материала, допустившему принципиальные ошибки в выполнении заданий, которые не позволят ему продолжить обучение без дополнительных занятий по соответствующей дисциплине.</li> </ul>
--

### **3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций в ходе освоения образовательной программы**

#### **3.1 Задания для текущего контроля успеваемости**

##### **Задания практических работ**

Практическая работа № 1. По теме: «Этапы проектирования программных продуктов»

Практическая работа № 2. По теме: «Обзор современных интегрированных сред разработки программных продуктов»

Практическая работа № 3 По теме: «Действия: планирование итераций, контроль итераций, мониторинг итераций»

Практическая работа № 4 По теме: «Инструментальные средства, используемые при оптимизации производительности»

Практическая работа № 4 По теме: «Основные способы тестирования программных продуктов»

Практическая работа № 5 По теме: «Хранилище конфигурации»

##### **Типовые задания для проведения промежуточной аттестации обучающихся**

1. Жизненный цикл ПС. Содержание основных этапов жизненного цикла ПС.
2. Анализ и разработка требований к ПС.
3. Ролевые кластеры проектной группы и их функции.
4. Специфические особенности ПС ВТ. ПС – новый вид товарной продукции.
5. Разработка внешних спецификаций на ПС.
6. Цели и порядок внутреннего проектирования ПС.
7. Прогнозирование технико-экономических показателей проектов ПС.
8. Модульная структура ПС.
9. Внешнее проектирование модулей.
10. Проектирование и кодирование модулей.
11. Стил программирования.
12. Принципы и методы тестирования ПС.
13. Проектирование теста.
14. Общая характеристика методов тестирования.
15. Ручные методы тестирования.
16. Машинные методы тестирования.
17. Методы структурного тестирования
18. Методы функционального тестирования.
19. Тестирование модулей.

20. Тестирование комплексов программ.
21. Отладка программ.
22. Документирование ПС.
23. Состав документации на ПС.
24. Испытания и сертификация ПС.
25. Методы, технология, средства обеспечения сертификации ПС.

### **3.2 Задания для промежуточной аттестации**

#### **Контрольные вопросы к экзамену**

1. Практическое применение HP Quality Center
2. Практическое применение HP Quick Test
3. IBM Rational Purify
4. IBM Rational Quantify
5. Профилировщики ANTS Profiler. JetBrains dotTrace.
6. IBM Rational PureCoverage
7. Практическое применение IBM Rational Robot
8. Практическое применение IBM Rational Functional Tester
9. Практическое применение IBM Rational Performance Tester
10. Средства нагрузочного тестирования QALoad, WAPT, JMeter и аналоги.
11. Система контроля версий SVN
12. Система контроля версий Perforce
13. Системы автоматической сборки, обзор, сравнение возможностей (рассмотреть минимум семь систем)
14. Система автоматической сборки FinalBuilder
15. Возможности Microsoft Team Foundation Server
16. Особенности современных методологий и технологий разработки ПС.
17. Направления развития и модели концепции открытых систем.
18. Технологии параллельного программирования.
19. Компонентные технологии и разработка распределенного ПО.
20. Руководство программным проектом.
21. Технологии коллективной разработки ПО.

## Итоговый тест

### Вопросы на проверку компетенции ПК-4.

Способен управлять инфраструктурой коллективной среды разработки

<b>ОЦЕНОЧНОЕ СРЕДСТВО (тестирование)</b>	<b>Контролируемая компетенция</b>
1. Технология разработки программного обеспечения это . . . А) комплекс организационных мер, направленных на разработку программных продуктов Б) операции и приемы разработки программных продуктов В) вычислительные комплексы Г) интернет	<b>ПК-4.1</b>
2. Жизненный цикл программного обеспечения это . . . А) период времени эксплуатации программного продукта Б) период времени проектирования программного продукта В) период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации Г) период времени, который начинается с момента разработки программного продукта и заканчивается в момент передачи его в эксплуатацию	<b>ПК-4.1</b>
3. Классическими моделями жизненного цикла программного обеспечения являются: А) спиральная Б) водопадная В) каскадная Г) итерационная	<b>ПК-4.1</b>
4. В какой модели жизненного цикла программного обеспечения переход на следующую стадию проектирования осуществляется только после того, как будет завершена работа на текущей стадии	<b>ПК-4.1</b>
5. Если проектируемый программный проект простой и детально определен, то целесообразно в процессе проектирования использовать модель жизненного цикла ПО А) спиральную Б) гибкую В) каскадную Г) итерационную	<b>ПК-4.1</b>
6. В итерационной модели жизненного цикла программного продукта . . . А) каждая итерация соответствует работам по отдельному этапу проектирования программной системы Б) риски могут увеличиваться по мере проектирования программного продукта В) количество итераций не регламентировано Г) на каждой итерации реализуется определенная функциональность программной системы	<b>ПК-4.1</b>
7. Для эффективного управления процессом разработки программных систем необходимо сформировать подход, который обеспечивал: А) управление и мониторинг деятельности команды проекта Б) распределение работ между участниками проекта В) виртуальную разработку Г) определение набора критериев качества программного продукта	<b>ПК-4.1</b>
8. С точки зрения процессов разработки программного обеспечения незрелой называют компанию, где А) процесс создания программного обеспечения зависят от таланта конкретных разработчиков Б) процесс создания программного обеспечения не зависят от таланта конкретных разработчиков В) принимаемые решения не зависят от таланта конкретных разработчиков Г) принимаемые решения зависят от таланта конкретных разработчиков	<b>ПК-4.1</b>
9. С точки зрения процессов разработки программного обеспечения зрелой называют компанию, где А) компании разрабатывают ясные процедуры управления проектами Б) в компании действуют корпоративные стандарты на процессы создания программных продуктов В) принимаемые решения зависят от таланта конкретных разработчиков Г) процесс создания программного обеспечения не зависят от таланта конкретных разработчиков	<b>ПК-4.1</b>

<p>10. Основными положениями гибкого подхода к созданию ПО являются:</p> <p>А) люди и взаимодействие важнее процессов и программных средств  Б) работающее программное обеспечение важнее исчерпывающей документации  В) готовность к изменениям важнее следования первоначальному плану  Г) всеобъемлющее тестирование важнее временных затрат</p>	<b>ПК-4.1</b>
<p>11. Гибкие методологии разработки программного обеспечения . .</p> <p>А) ориентированы на минимизацию рисков  Б) реализуются короткими итерациями  В) каждая предполагает выпуск новой версии продукта  Г) каждая итерация заканчивается выпуском готового продукта</p>	<b>ПК-4.1</b>
<p>12. Методология управления жизненным циклом приложений представляет собой. . .</p> <p>А) методологию водопадного проектирования программного обеспечения  Б) концепцию управления программным проектом на всех этапах его жизни  В) концепцию управления программными системами  Г) методологию итерационного проектирования программных продуктов</p>	<b>ПК-4.1</b>
<p>13. Компания Hewlett-Packard является поставщиком решений по управлению жизненным циклом приложений</p> <p>А) ALM on SaaS  Б) Open ALM  В) Rational® ClearQuest®  Г) Visual Studio 2012 и Team Foundation Server</p>	<b>ПК-4.1</b>
<p>14. Компания IBM является поставщиком решений по управлению жизненным циклом приложений</p> <p>А) ALM on SaaS  Б) Open ALM  В) Rational® ClearQuest®  Г) Visual Studio 2012 и Team Foundation Server</p>	<b>ПК-4.1</b>
<p>15. Компания Microsoft является поставщиком решений по управлению жизненным циклом приложений</p> <p>А) ALM on SaaS  Б) Open ALM  В) Rational® ClearQuest®  Г) Visual Studio 2012 и Team Foundation Server</p>	<b>ПК-4.1</b>
<p>16. Методология разработки программного обеспечения Microsoft Solutions Framework используется при разработке ИТ-решений и описывает</p> <p>А) управление рабочими процессами  Б) управление командами разработчиков  В) управление компьютерами  Г) управление сетями</p>	<b>ПК-4.1</b>
<p>17. ИТ-решение компании Microsoft - это . . .</p> <p>А) программные средства общего назначения  Б) набор программных средств для сетевых операционных систем  В) скоординированная поставка набора элементов, необходимых для удовлетворения бизнес-потребности конкретного заказчика  Г) набор серверных решений для управления информационными системами</p>	<b>ПК-4.1</b>
<p>18. Управление жизненным циклом приложения в Visual Studio базируется на следующих принципах</p> <p>А) мобильность  Б) продуктивность  В) интеграция  Г) расширяемость</p>	<b>ПК-4.1</b>
<p>19. Продуктивность управления жизненным циклом приложения в Visual Studio обеспечивается . .</p> <p>А) возможностью совместной работы членов команды и заинтересованных лиц  Б) управлением сложностью программного продукта  В) универсальностью членов команды  Г) четким планом-графиком работ</p>	<b>ПК-4.1</b>
<p>20. Интеграция при управлении жизненным циклом приложения в Visual Studio обеспечивается . . .</p> <p>А) использованием языка программирования C#  Б) возможностями по предоставлению всем участникам проекта информации о состоянии дел</p>	<b>ПК-4.1</b>

В) хранением всех рабочих элементов проекта в базе данных Team Foundation Server Г) виртуализацией рабочей среды	
21. Расширяемость при управлении жизненным циклом приложения в Visual Studio обеспечивается ... А) многоядерными процессорами серверов Б) облачными технологиями В) интегрированной средой разработки - IDE Г) API-интерфейсом служб Team Foundation Server	<b>ПК-4.1</b>
22. Архитектурные инструменты в Visual Studio 2012 Ultimate позволяют создавать следующие виды схем А) функциональные схемы Б) схема активности В) схема последовательностей Г) схема классов	<b>ПК-4.1</b>
23. Архитектурные инструменты в Visual Studio 2012 Ultimate НЕ позволяют создавать следующие виды схем А) схема слоев Б) схема компонентов В) схемы соединений Г) схема вариантов использования	<b>ПК-4.1</b>
24. Какие схемы определяют функциональность системы и описывают с точки зрения пользователей их возможные действия с программным продуктом А) схемы (диаграммы) классов UML Б) схемы (диаграммы) последовательностей UML В) схемы (диаграммы) вариантов использования UML Г) схемы (диаграммы) активности UML	<b>ПК-4.1</b>
25. Какие схемы описывают бизнес-процесс или программный процесс в виде потока работ через последовательные действия А) схемы (диаграммы) компонентов UML Б) схемы (диаграммы) последовательностей UML В) схемы (диаграммы) вариантов использования UML Г) схемы (диаграммы) активности UML	<b>ПК-4.1</b>
26. Какие схемы описывают распределение программных составляющих приложения, позволяя наглядно отобразить на высоком уровне структуру компонентов и служб А) схемы (диаграммы) компонентов UML Б) схемы (диаграммы) слоев UML В) схемы (диаграммы) вариантов использования UML Г) схемы (диаграммы) активности UML	<b>ПК-4.1</b>
27. Какие схемы используются для описания логической архитектуры системы А) схемы (диаграммы) компонентов UML Б) схемы (диаграммы) классов UML В) схемы (диаграммы) вариантов использования UML Г) схемы (диаграммы) слоев UML	<b>ПК-4.1</b>
28. Какие метрики кода позволяет вычислять Visual Studio 2012? А) сложность организации циклов Б) глубина наследования В) отклик класса Г) индекс удобства поддержки	<b>ПК-4.1</b>
29. Какие метрики кода НЕ позволяет вычислять Visual Studio 2012? А) строки кода Б) низкое зацепление классов В) объединение классов Г) глубина наследования	<b>ПК-4.1</b>
30. Укажите правильные утверждения в отношении инструментов профилирования А) профилировщики с выборкой делают постоянные снимки выполняющегося приложения и записывают его состояние Б) профилировщики с выборкой делают периодические снимки выполняющегося приложения и записывают его состояние В) профилировщики с инструментированием добавляют маркеры отслеживания только в начало каждой исследуемой функции Г) профилировщики с инструментированием добавляют маркеры отслеживания в начало и конец каждой исследуемой функции	<b>ПК-4.1</b>



31. Укажите НЕ правильные утверждения в отношении инструментов профилирования А) профилирование реализует процесс наблюдения поведения приложения Б) профилирование реализует процесс записи показателей о поведении приложения В) профилировщик позволяет обнаружить у приложения проблемы с производительностью Г) профилировщик позволяет обнаружить ошибки кода	<b>ПК-4.1</b>
32. Какие виды тестов может использовать разработчик в Visual Studio 2012 Ultimate А) модульные тесты Б) нагрузочные тесты В) веб тесты производительности Г) тесты пользовательского интерфейса	<b>ПК-4.1</b>
33. Какие тесты позволяют быстро проверить наличие логических ошибок в методах классов А) модульные тесты Б) нагрузочные тесты В) веб тесты производительности Г) тесты пользовательского интерфейса	<b>ПК-4.1</b>
34. Какие тесты используются для исследования работоспособности приложения путем моделирования множества пользователей, которые работают с программой одновременно А) модульные тесты Б) нагрузочные тесты В) веб тесты производительности Г) тесты пользовательского интерфейса	<b>ПК-4.1</b>
35. Какие тесты позволяют автоматически сформировать код теста, путем записи действий пользователя при работе с приложением, и впоследствии выполнять эти тесты автоматически А) модульные тесты Б) нагрузочные тесты В) веб тесты производительности Г) тесты пользовательского интерфейса	<b>ПК-4.1</b>
36. Укажите программный инструмент, который позволяет создавать планы тестирования, формировать, добавлять и удалять тестовые случаи А) Lab Management Б) Team Foundation Server В) Microsoft Test Manager Г) Visual Studio 2012	<b>ПК-4.1</b>
37. Построение SADT-модели включает в себя выполнение следующих действий А) Написание программного обеспечения для разрабатываемой системы по требованиям заказчика Б) Сбор информации об объекте, определение его границ В) Определение цели и точки зрения модели, построение, обобщение и декомпозиция диаграмм Г) Представление исследуемой системы в графическом виде Д) Представление исследуемого объекта средствами системного моделирования Е) Критическая оценка, рецензирование и комментирование Ж) Разработка, отладка и тестирование программного обеспечения З) Использование графических пакетов для представления системы в виде модели	<b>ПК-4.1</b>
38. CASE-средства классифицируются по следующим признакам: А) По применяемым методологиям и моделям систем и БД Б) По используемому программному обеспечению В) По этапам жизненного цикла программного обеспечения Г) По степени интегрированности с СУБД Д) По уровням детализации и декомпозиции проектируемой системы Е) По доступным платформам Ж) По используемым языкам программирования З) По степени сложности моделируемой системы	<b>ПК-4.1</b>
39. К малым интегрированным средствам моделирования относятся: А) ARIS Toolset Б) Design/IDEF В) ERwin Г) BPwin Д) Designer/2000 Е) Paradigm Plus Ж) Model Mart	<b>ПК-4.1</b>

3) Rational Rose	
40. К средним интегрированным средствам моделирования относятся: А) Rational Rose Б) Design/IDEF В) BPwin Г) Designer/2000 Д) ARIS Toolset Е) Model Mart Ж) Paradigm Plus З) ERwin	<b>ПК-4.1</b>
41. Объектно-ориентированная методология (ООМ) включает в себя составные части: А) Объектно-ориентированный анализ Б) Объектно-ориентированный подкласс В) Объектно-ориентированное проектирование Г) Объектно-ориентированная парадигма Д) Объектно-ориентированная экспозиция Е) Объектно-ориентированное моделирование Ж) Объектно-ориентированное программирование З) Объектно-ориентированная декомпозиция	<b>ПК-4.1</b>
42. К основным понятиям объектно-ориентированного подхода относятся А) Обобщение Б) Полиморфизм В) Инкапсуляция Г) Реализация Д) Агрегирование Е) Наследование Ж) Ассоциация З) Композиция	<b>ПК-4.1</b>
43. К инструментальным средствам объектно-ориентированного анализа и проектирования относятся А) Rational Rose Б) Model Mart В) MS Visio Г) ARIS Д) IDEF1X Е) Erwin Ж) BPwin З) JAM	<b>ПК-4.1</b>
44. К инструментальным средствам представления функциональных моделей относятся: А) JAM Б) Model Mart В) MS Visio Г) ARIS Д) IDEF0 Е) Erwin Ж) BPwin З) Rational Rose	<b>ПК-4.1</b>
45. Методологии, поддерживаемые в BPwin: А) IDEF1X Б) IDEF0 В) IDEF1 Г) IDEF3 Д) IDEFX Е) IDEF5 Ж) DFD З) DFD1X	<b>ПК-4.1</b>
46. Диаграмма IDEF0 может содержать следующие типы диаграмм: А) Диаграмму классов Б) Контекстную диаграмму, диаграмму декомпозиции В) Диаграмму компонентов Г) Диаграмму дерева узлов Д) Диаграмму взаимодействий	<b>ПК-4.1</b>

<p>Е) Диаграмму только для экспозиции (FEO)  Ж) Диаграмму последовательности, диаграмму кооперации  З) Диаграмму узлов</p>	
<p>47. Уровни логической модели:  А) Диаграмма сущность  Б) Диаграмма связь  В) Диаграмма пакетов  Г) Диаграмма сущность-связь  Д) Модель данных, основанная на классах  Е) Модель данных, основанная на ключах  Ж) Полная операционная модель  З) Полная атрибутивная модель</p>	<b>ПК-4.1</b>
<p>48. ВРwin допускает следующие переходы с одной нотации на другую:  А) IDEF3 → DFD  Б) DFD → IDEF0  В) IDEF0 → DFD  Г) DFD → DFD  Д) IDEF3 → IDEF0  Е) IDEF0 → IDEF3  Ж) IDEF3 → IDEF3  З) DFD → IDEF3</p>	<b>ПК-4.1</b>
<p>49. DFD описывает:  А) Функции обработки стрелок (arrow)  Б) Функции обработки информации (работы)  В) Внешние ссылки (external references), объекты, сотрудников или отделы, которые участвуют в обработке информации  Г) Документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке информации  Д) Функции обработки внешних ссылок  Е) Внешние ссылки (external references), таблицы для хранения документов (хранилище данных, data store)  Ж) Функции обработки документов  З) Документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке внешних стрелок</p>	<b>ПК-4.1</b>
<p>50. Инструментальное средство ERwin позволяет:  А) Редактировать и отлаживать программы  Б) Проектировать на физическом и логическом уровне модели данных  В) Управлять процессом конструирования ПО  Г) Проектировать диаграммы вариантов использования и взаимодействий  Д) Проводить процессы прямого и обратного проектирования баз данных  Е) Управлять процессом трансляции и отладки программ  Ж) Выравнивать модель и содержимое системного каталога после редактирования  З) Проектировать контекстные диаграммы и диаграммы декомпозиции</p>	<b>ПК-4.1</b>
<p>51. Для создания моделей ERwin используют международно признанные системы обозначений (нотации):  А) IDEF0  Б) IDEF1X  В) IDEF3  Г) DFD  Д) IE  Е) DM  Ж) IDEFDFD  З) IDEF3</p>	<b>ПК-4.1</b>
<p>52. К основным компонентам диаграммы ERwin относятся:  А) Сущности  Б) Переходы  В) Атрибуты  Г) Классы</p>	<b>ПК-4.1</b>

<ul style="list-style-type: none"> <li>Д) Слияния</li> <li>Е) Разветвления</li> <li>Ж) Использования</li> <li>З) Связи</li> </ul>	
<p>53. Язык UML – это:</p> <ul style="list-style-type: none"> <li>А) Язык программирования высокого уровня</li> <li>Б) Унифицированный язык моделирования</li> <li>В) Язык для разработки систем искусственного интеллекта</li> <li>Г) Unified Modeling Language</li> <li>Д) Язык управления базами данных</li> <li>Е) Язык для визуализации, специфицирования, конструирования и документирования артефактов программных систем</li> <li>Ж) Язык создания запросов в базах данных</li> <li>З) Язык программирования низкого уровня</li> </ul>	<b>ПК-4.1</b>
<p>54. Моделирование в UML позволяет решать задачи:</p> <ul style="list-style-type: none"> <li>А) Анализа и синтеза систем управления</li> <li>Б) Разработать и отладить программное обеспечение</li> <li>В) Визуализировать систему в ее текущем или желательном для нас состоянии</li> <li>Г) Провести тестирование разработанного программного обеспечения</li> <li>Д) Описать структуру или поведение системы; получить шаблон, позволяющий сконструировать систему</li> <li>Е) Смоделировать разрабатываемую информационную систему</li> <li>Ж) Документировать принимаемые решения, используя полученные модели</li> <li>З) Рассчитать экономическую эффективность от внедрения программного обеспечения</li> </ul>	<b>ПК-4.1</b>
<p>55. UML, как язык документирования, помимо исполняемого кода производит и другие продукты, включающие:</p> <ul style="list-style-type: none"> <li>А) Требования, архитектуру, проектные решения</li> <li>Б) Спецификацию технических средств</li> <li>В) Дизайн, исходный код, проектные планы,</li> <li>Г) Требования к уровню квалификации разработчиков</li> <li>Д) Набор заданий для тестирования программного обеспечения</li> <li>Е) Требования к уровню квалификации персонала сопровождения</li> <li>Ж) Тесты, прототипы, релизы (версии)</li> <li>З) Требования к выбору языка программирования</li> </ul>	<b>ПК-4.1</b>
<p>56. Механизмы расширения UML включают:</p> <ul style="list-style-type: none"> <li>А) Исключения</li> <li>Б) Стереотипы</li> <li>В) Дополнения</li> <li>Г) Управления</li> <li>Д) Помеченные значения</li> <li>Е) Слияния</li> <li>Ж) Ограничения</li> <li>З) Объединения</li> </ul>	<b>ПК-4.1</b>
<p>57. В объектно-ориентированном моделировании между классами существуют типы связей:</p> <ul style="list-style-type: none"> <li>А) Слияние</li> <li>Б) Линейность</li> <li>В) Зависимость</li> <li>Г) Разветвление</li> <li>Д) Цикличность</li> <li>Е) Обобщение</li> <li>Ж) Ассоциация</li> <li>З) Агрегация</li> </ul>	<b>ПК-4.1</b>
<p>58. В состав графического представления класса в языке UML входят части:</p> <ul style="list-style-type: none"> <li>А) Отношения</li> <li>Б) Имя</li> <li>В) Связи</li> <li>Г) Атрибуты</li> <li>Д) Описание</li> <li>Е) Сущности</li> <li>Ж) Операции</li> <li>З) Механизмы</li> </ul>	<b>ПК-4.1</b>
<p>59. Инструментальные средства разработки программ – это:</p>	<b>ПК-4.1</b>

<p>А) Средства создания новых программ  Б) Сервисные средства разработки ПО  В) Аналитические средства разработки ПО  Г) Программное обеспечение, предназначенное для разработки и отладки новых программ  Д) Средства отладки ПО  Е) Средства тестирования ПО  Ж) Аппаратные и программные инструменты разработки нового ПО  З) Технические инструментальные средства разработки ПО</p>	
<p>60. Аппаратные инструментальные средства разработки ПО – это:  А) Система для разработки новых программ на конкретном языке программирования  Б) Средства создания и редактирования текстов программ  В) Микропроцессор и подключаемые (внешние) устройства  Г) Устройства вычислительной системы, специально предназначенные для поддержки разработки ПО  Д) Периферийные устройства, микропроцессор вычислительного комплекса, предназначенные для разработки нового ПО  Е) Программное обеспечение, написанное на языках программирования низкого уровня  Ж) Программы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ  З) Программы, используемые для корректировки и тестирования других прикладных или системных программ</p>	<b>ПК-4.1</b>
<p>61. Транслятор – это:  А) Программа, выполняющая перевод программы с одного языка программирования на другой  Б) Комплекс программ мультимедийных технологий  В) Программа, которая выполняет перевод программы с одного языка программирования на машинные коды  Г) Программа-переводчик с одного иностранного языка на другой  Д) Техническое устройство передачи и преобразования аудио и видеосигналов  Е) Техническое устройство для кодирования и декодирования информации  Ж) Программное обеспечение для обеспечения защиты информации на компьютере  З) Одно из основных средств автоматизации программирования для преобразования программы, написанный на машинно-независимом языке, в программу на машинном языке конкретной ЭВМ</p>	<b>ПК-4.1</b>
<p>62. Компилятор – это:  А) Один из видов трансляторов  Б) Прикладное программное обеспечение  В) Специальная утилита системного ПО  Г) Операционная оболочка  Д) Переводит в коды сразу всю программу и создает независимый исполняемый файл  Е) Программное обеспечение, используемое в издательских системах  Ж) Программа, которая переводит программу, написанную на языке программирования высокого уровня в программу на машинном языке не участвуя в ее исполнении  З) Переводит в машинные коды 1 строчку программы и сразу ее выполняет</p>	<b>ПК-4.1</b>
<p>63. Интерпретатор:  А) Программа для создания и редактирования электронных таблиц  Б) Программа, анализирующая команды или операторы исходной программы и немедленно выполняющая их  В) Переводит в коды сразу всю программу и создает независимый исполняемый файл  Г) Переводит в машинные коды 1 строчку программы и сразу ее выполняет  Д) Программа для создания и редактирования текстовых документов  Е) Один из видов трансляторов  Ж) Программа создания и управления базами данных  З) Программа создания файлов мультимедиа</p>	<b>ПК-4.1</b>
<p>64. Компоновщик – это:  А) Программа для компоновки и оформления тестовых документов  Б) Редактор связей  В) Комплекс программ, для создания и ведения баз данных  Г) Программа, которая из одного или нескольких объектных модулей с привлечением библиотечных программ и стандартных подпрограмм формирует загрузочный модуль  Д) Программное обеспечение для создания презентаций  Е) Программа сборки загрузочного модуля из полученных в результате отдельной</p>	<b>ПК-4.1</b>

<p>компиляции объектных модулей с автоматическим поиском и присоединением библиотечных подпрограмм и процедур</p> <p>Ж) Программа для поиска синтаксических и семантических ошибок в программе</p> <p>З) Программа</p>	
<p>65. Отладчик:</p> <p>А) Программа, облегчающая программисту выполнение отладки разрабатываемых им программ</p> <p>Б) Программа для создания системы защиты файла</p> <p>В) Программа создания системы защиты от вирусных атак</p> <p>Г) Программа, помогающая анализировать поведение отлаживаемой программы, обеспечивая ее трассировку</p> <p>Д) Операционная оболочка для создания и управления файловыми структурами</p> <p>Е) Системное программное обеспечение для настройки операционной системы</p> <p>Ж) Программа создания и редактирования графических файлов</p> <p>З) Программа, позволяющая выполнять остановы в заданных точках, просмотреть текущие значения переменных и изменять их значения</p>	<b>ПК-4.1</b>
<p>66. К этапам развития технологии разработки программного обеспечения относятся:</p> <p>А) «Процедурное» программирование</p> <p>Б) Программирование на алгоритмических языках высокого уровня</p> <p>В) Структурный подход к программированию</p> <p>Г) Программирование на языках низкого уровня</p> <p>Д) Компонентный подход и CASE-технологии</p> <p>Е) Машинно-ориентированное программирование</p> <p>Ж) Машинно-независимое программирование</p> <p>З) Подход к разработке ПО, основанный на стратегии поиска</p>	<b>ПК-4.1</b>
<p>67. Структурный подход к программированию – это:</p> <p>А) Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения</p> <p>Б) Создание программного обеспечения на основе структурной схемы решаемой задачи</p> <p>В) Подход, требующий разработки структурной схемы алгоритма и программы решения задачи</p> <p>Г) Подход, в основе которого лежит декомпозиция (разбиение на части) сложных систем с целью последующей реализации в виде отдельных небольших (до 40-50 операторов) подпрограмм</p> <p>Д) Подход к решению задачи, требующий создание структурной схемы этапов работ по разработке программного обеспечения</p> <p>Е) Процесс создания программного обеспечения на основе структурной схемы исследуемого объекта или процесса</p> <p>Ж) Технология разработки программного обеспечения на базе структурной схемы развития языков программирования</p> <p>З) Подход, требующий представления задачи в виде иерархии подзадач простейшей структуры</p>	<b>ПК-4.1</b>
<p>68. Объектный подход к программированию – это:</p> <p>А) Технология создания сложного программного обеспечения, основанная на представлении задачи исследования как объекта</p> <p>- Технология создания сложного программного обеспечения, предназначенного для автоматизации технологических объектов</p> <p>Б) 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</p> <p>В) Технология создания сложного программного обеспечения, основанная на представлении программы как единого объекта</p> <p>Г) Технология создания сложного программного обеспечения, позволяющая вести практически независимую разработку отдельных частей (объектов) программы</p> <p>Д) Технология создания сложного программного обеспечения, основанная на объектном представлении кода программы</p> <p>Е) Технология создания сложного программного обеспечения, в основе которой лежат новые способы организации программ, основанные на механизмах наследования, полиморфизма, композиции, наполнения</p> <p>Ж) Технология создания сложного программного обеспечения, основанная на объектно-ориентированном программировании</p>	<b>ПК-4.1</b>
<p>69. Компонентный подход:</p>	<b>ПК-4.1</b>

<p>А) Предполагает построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения</p> <p>Б) Предполагает взаимодействие между компонентами через стандартизованные двоичные интерфейсы и позволяет использовать исполняемые файлы в любом языке программирования, поддерживающем соответствующую технологию</p> <p>В) Позволяет рассматривать объект исследования, как структуру, состоящую из отдельных компонент</p> <p>- способ написания исходного кода программного обеспечения</p> <p>Г) Позволяет собрать объекты-компоненты в динамически вызываемые библиотеки или исполняемые файлы, и распространять в двоичном виде (без исходных текстов)</p> <p>Д) Способ отладки и тестирования программного обеспечения</p> <p>Е) Способ внедрения и опытной эксплуатации программного обеспечения.</p> <p>Ж) Метод выработки требований к разработке программного обеспечения</p>	
<p>70. К методам выявления требований относятся:</p> <p>А) Беседы с первыми руководителями предприятия, для которого разрабатывается программное обеспечение</p> <p>Б) Анализ научной и технической литературы, посвященной вопросам разработки программного обеспечения</p> <p>В) Личные встречи и беседы со всеми сотрудниками предприятия</p> <p>Г) Анализ технической документации и на основе нее разработка требований к системе</p> <p>Д) На начальном этапе требования не выявляются, а формируются по мере разработки программного обеспечения</p> <p>Е) Интервьюирование и анкетирование, мозговой штурм и отбор идей</p> <p>Ж) Совещания, посвященные требованиям, создание прототипов</p> <p>З) Раскадровки, прецеденты, обыгрывание ролей</p>	<b>ПК-4.1</b>
<p>71. Требования к разрабатываемой системе должны включать:</p> <p>А) Разработку программного обеспечения и выработка требований к изменению работы системы заказчика</p> <p>Б) 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)</p> <p>В) Построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения</p> <p>Г) Описание выполняемых системой функций</p> <p>- Технологию создания сложного программного обеспечения, основанную а объектном представлении кода программы</p> <p>Д) Ограничения в процессе разработки (директивные сроки завершения отдельных этапов, имеющиеся ресурсы, организационные процедуры и мероприятия, обеспечивающие защиту информации)</p> <p>Е) Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения</p> <p>Ж) Технологию разработки программного обеспечения на базе структурной схемы развития языков программирования</p>	<b>ПК-4.1</b>
<p>72. Для различных представлений проектируемой системы используют типы моделей:</p> <p>А) Статическая модель</p> <p>Б) Динамическая модель</p> <p>В) Модель классов</p> <p>Г) Модель декомпозиции</p> <p>Д) Модель размещения</p> <p>Е) Модель состояний</p> <p>Ж) Модель взаимодействия</p> <p>З) Модель агрегации</p>	<b>ПК-4.1</b>
<p>73. Классификация имитационных моделей:</p> <p>А) Статистическая</p> <p>Б) Адаптивная</p> <p>В) Статическая или динамическая</p> <p>Г) Структурная</p> <p>Д) Сетерминированная или стохастическая</p> <p>Е) Непрерывная или дискретная</p> <p>Ж) Объединенная</p> <p>З) Декомпозиционная</p>	<b>ПК-4.1</b>
<p>74. Разработка и сопровождение ИС в конкретной организации и конкретном проекте</p>	<b>ПК-4.1</b>

<p>должна поддерживаться стандартами:</p> <p>А) Стандарт организации  Б) Стандарт конкретного проекта  В) Стандарт проектирования  Г) Стандарт оценки  Д) Стандарт оформления проектной документации  Е) Стандарт аудита  Ж) Стандарт оформления разработки  З) Стандарт пользовательского интерфейса</p>	
<p>75. Результатами проектирования архитектуры являются:</p> <p>А) Модель административного интерфейса  Б) Модель процессов  В) Модель потоков  Г) Модель классов  Д) Модель данных  Е) Модель пользовательского интерфейса  Ж) Модель компонентов  З) Модель узлов</p>	<b>ПК-4.1</b>
<p>76. Какие технологии разработки программ используются в современном программировании:</p> <p>А) Визуальные  Б) Событийные  В) Структурные  Г) Объектно-ориентированные  Д) Модульные  Е) Текстуальные  Ж) Графические  З) Машинно-ориентированное</p>	<b>ПК-4.1</b>
<p>77. Объектно-ориентированное проектирование использует инструментальные средства:</p> <p>А) Rational Rose  Б) Vpwin  В) ARIS  Г) Idef1X  Д) Erwin  Е) MS Visio  Ж) Jam</p>	<b>ПК-4.1</b>
<p>78. Проектирование функциональных моделей поддерживается инструментальными средствами:</p> <p>А) Jam  Б) Model Mart  В) MS visio  Г) ERwin  Д) Idef0  Е) Aris  Ж) Rational rose  З) VPwin</p>	<b>ПК-4.1</b>
<p>79. IEEE – это:</p> <p>А) Коммерческая организация ученых и исследователей  Б) Просто принятое обозначение, расшифровки не имеет  В) Обозначение всемирной компьютерной сети  Г) Всемирная некоммерческая техническая профессиональная ассоциация ученых и исследователей  Д) Такая аббревиатура нигде не используется  Е) Institute Of Electrical and Electronic Engineers, Inc  Ж) Американская организация ученых-экономистов  З) Институт инженеров радиотехники и электротехники</p>	<b>ПК-4.1</b>
<p>80. Ядро знаний SWEBOOK – это:</p> <p>А) ГОСТ на разработку программного обеспечения  Б) Нормативный документ, разработанный IEEE  В) ГОСТ на разработку информационных систем  Г) Документ, устанавливающий правовые отношения между заказчиком и разработчиком программного обеспечения</p>	<b>ПК-4.1</b>



<p>Д) Основополагающий научно-технический документ, который отображает мнение специалистов в области программной инженерии</p> <p>Е) Документ, устанавливающий методику тестирования и испытания программного обеспечения</p> <p>Ж) Документ, который согласуется с современными регламентированными процессами жизненного цикла ПО стандарта ISO/IEC 12207</p> <p>З) ГОСТ на разработку и комплектацию сопровождающей документации</p>	
<p>81. Этапы разработки консалтинговых проектов включают в себя:</p> <p>А) Анализ первичных требований и планирование работ</p> <p>Б) Снятие программного продукта с эксплуатации</p> <p>В) Декомпозицию задачи на подзадачи</p> <p>Г) Разработку спецификации и документации</p> <p>Д) Проведение обследования деятельности предприятия</p> <p>Е) Тестирование и сопровождение программного обеспечения</p> <p>Ж) Построение моделей деятельности предприятия (модели AS – IS – “как есть” и модели TO – BE – “как должно быть”)</p> <p>З) Разработку программного обеспечения</p>	<b>ПК-4.1</b>
<p>82. Концепции, лежащие в основе модульного программирования:</p> <p>А) Объем реализации и время исполнения (реакции)</p> <p>Б) Мера автоматизма в работе реализации и инструментах разработки</p> <p>В) Визуальность и тестируемость разработки</p> <p>Г) Функциональная декомпозиция, пространственная и временная группировка информации (модульность)</p> <p>Д) Упрощение связей</p> <p>Е) Комментируемость функций и данных</p> <p>Ж) Надежность, устойчивость</p> <p>З) Безопасность</p>	<b>ПК-4.1</b>
<p>83. Для архитектурного проектирования в Visual Studio используются инструменты визуального проектирования, которые предназначены для следующего:</p> <p>А) визуализации архитектурных аспектов проектируемой системы</p> <p>Б) создания моделей структуры и поведения системы</p> <p>В) документирования принятых решений</p> <p>Г) разработки шаблонов для проектирования системы</p>	<b>ПК-4.1</b>
<p>84. Технологии коллективной разработки программных систем позволяют эффективно совместно _____ программный код.</p>	<b>ПК-4.1</b>
<p>85. В рамках технологий коллективной разработки программных систем используются специализированные _____ для управления версиями</p>	<b>ПК-4.1</b>
<p>86. Одним из основных преимуществ технологий коллективной разработки является возможность _____ кода между разработчиками.</p>	<b>ПК-4.1</b>
<p>87. Системы управления версиями позволяют отслеживать и контролировать изменения в _____ кода.</p>	<b>ПК-4.1</b>
<p>88. Для эффективной коллективной разработки программных систем используются инструменты для _____ задач и работы над кодом.</p>	<b>ПК-4.1</b>
<p>89. Технологии коллективной разработки программных систем обеспечивают возможность синхронизации и _____ работы разработчиков.</p>	<b>ПК-4.1</b>
<p>90. При использовании технологий коллективной разработки программных систем, разработчики могут вносить и отслеживать _____ в коде.</p>	<b>ПК-4.1</b>
<p>91. Одной из основных задач технологий коллективной разработки программных систем является разрешение _____ в коде.</p>	<b>ПК-4.1</b>
<p>92. Системы управления версиями позволяют восстанавливать предыдущие _____ кода при необходимости.</p>	<b>ПК-4.1</b>
<p>93. Для эффективной коллективной разработки программных систем важно наличие _____ и обмена опытом между разработчиками.</p>	<b>ПК-4.1</b>
<p>94. Технологии коллективной разработки программных систем способствуют повышению _____ между разработчиками.</p>	<b>ПК-4.1</b>
<p>95. При использовании технологий коллективной разработки программных систем разработчики могут одновременно работать над _____ версиями кода.</p>	<b>ПК-4.1</b>
<p>96. Одним из основных принципов технологий коллективной разработки является _____</p>	<b>ПК-4.1</b>

доступ к исходному коду.	
97. Технологии коллективной разработки программных систем обеспечивают возможность автоматического _____ кода.	<b>ПК-4.1</b>
98. При использовании систем управления версиями возможно создание различных _____ для разработчиков.	<b>ПК-4.1</b>
99. Технологии коллективной разработки программных систем обеспечивают возможность _____ обратной связи между разработчиками.	<b>ПК-4.1</b>
100. При использовании технологий коллективной разработки программных систем, разработчики могут _____ изменения в коде.	<b>ПК-4.1</b>
101. Системы управления версиями позволяют отслеживать _____ между различными версиями кода.	<b>ПК-4.1</b>
102. Технологии коллективной разработки программных систем обеспечивают возможность параллельной _____ разработки различных компонентов программы.	<b>ПК-4.1</b>
103. При использовании технологий коллективной разработки программных систем возможно эффективное _____ и интеграция изменений от разных разработчиков.	<b>ПК-4.1</b>

**Ключи к заданиям по дисциплине «Технологии коллективной разработки программных систем»**

1	А	53	Б
2	В	54	В
3	Б, В	55	А
4	водопадной	56	Б
5	В	57	В
6	В	58	Б
7	А	59	А
8	А	60	В
9	Б	61	А
10	А	62	А
11	А	63	Б
12	Б	64	Б
13	А	65	А
14	В	66	А
15	Г	67	А
16	А	68	Б
17	В	69	А
18	Б	70	Е
19	А	71	Б
20	Б	72	В
21	В	73	В
22	Б	74	В
23	В	75	Б
24	В	76	А
25	Г	77	А
26	А	78	Б
27	Г	79	Г
28	А	80	Б
29	Б	81	А
30	Б	82	Г
31	Г	83	А
32	А	84	изменять
33	А	85	инструменты
34	Б	86	совместного
35	Г	87	исходного
36	В	88	управления
37	Б	89	координации
38	А	90	изменения
39	В	91	конфликтов
40	Б	92	версий
41	А	93	коммуникации
42	Б	94	сотрудничества
43	А	95	параллельными
44	Б	96	общий
45	Б	97	автоматизированного
46	Б	98	веток
47	Г	99	обратной
48	В	100	отслеживать
49	Б	101	различия
50	Б	102	одновременной
51	Б	103	объединение
52	А		