

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»**
(МИ ВлГУ)

Кафедра *ЭиВТ*

«УТВЕРЖДАЮ»
Заместитель директора по УР
Д.Е. Андрианов
_____ 16.06.2020

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Технология разработки программного обеспечения

Направление подготовки

09.03.01 Информатика и вычислительная техника

Профиль подготовки

Вычислительные машины, комплексы, системы и сети

Семестр	Трудоем- кость, час./зач. ед.	Лек- ции, час.	Практи- ческие занятия, час.	Лабора- торные работы, час.	Консуль- тация, час.	Конт- роль, час.	Всего (контак- тная работа), час.	СРС, час.	Форма промежу- точного контроля (экз., зач., зач. с оц.)
7	108 / 3	16	16	16	1,6	0,25	49,85	58,15	Зач.
8	108 / 3	12		12	3,2	0,35	27,55	53,8	Экз.(26,65)
Итого	216 / 6	28	16	28	4,8	0,6	77,4	111,95	26,65

Муром, 2020 г.

1. Цель освоения дисциплины

Цель дисциплины: обучение студентов систематизированным представлениям о проектировании программных систем, о методах проектирования структуры программного обеспечения, о современных методах и средствах проектирования информационных систем.

Основными задачами изучения дисциплины является знакомство с основными этапами жизненного цикла ПО, технологиями функционального и объектно-ориентированного проектирования ПО, приобретение навыков работы со средствами автоматизированной разработки ПО.

В результате изучения дисциплины студенты должны:

- иметь представление о проблемах и направлениях развития технологии программирования; об основных методах и средствах автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения; о направлениях развития методов и программных средств коллективной разработки программного обеспечения.

- знать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного обеспечения; методы организации работы в коллективах разработчиков программного обеспечения; основные положения метрологии программных продуктов, принципы построения, проектирования и использования средств для измерений характеристик и параметров программ, программных систем и комплексов

- уметь использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемами работы с инструментальными средствами, поддерживающими создание программного обеспечения; методы организации работы в коллективах разработчиков программного обеспечения; основные положения метрологии программных продуктов, принципы построения, проектирования и использования средств для измерений характеристик и параметров программ, программных систем и комплексов.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Технология разработки программного обеспечения» базируется на знаниях, полученных студентами по естественнонаучным, общепрофессиональным и специальным дисциплинам: «Информатика», «Программирование», «Дискретная математика», «Теория автоматов», «Архитектура микропроцессоров и программирование на языке ассемблеров», «Математическая логика и теория алгоритмов». Углубление и расширение вопросов, изложенных в данном курсе, будет осуществляться во время работы студентов на дисциплине «Системы управления базами данных», а также в ходе написания выпускной квалификационной работы.

3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
ПК-5 Способен разрабатывать требования, проектировать программное обеспечение, разрабатывать стратегии тестирования, проектировать	ПК-5.1 Осуществляет проектирование программного обеспечения с применением современных средств и технологий.	Знать о проблемах и направлениях развития технологии программирования; методы проектирования программного обеспечения; основные методы и средства автоматизации проектирования программного обеспечения;	вопросы к устному опросу

пользовательские интерфейсы		<p>методы организации работы в коллективах разработчиков программного обеспечения (ПК-5.1)</p> <p>Уметь использовать методы проектирования и производства программного продукта; строить модели вычислительных процессов с учетом особенностей архитектуры вычислительных систем, на которых планируется использовать разрабатываемое ПО; ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования; разрабатывать основные программные документы (ПК-5.1)</p> <p>Владеть навыком использования методов проектирования программного обеспечения; навыком использования средств автоматизации проектирования программного обеспечения; навыками разработки ПО с помощью современных интегрированных сред (ПК-5.1)</p>	
	<p>ПК-5.3 Осуществляет анализ предметной области и формулирует требования к программному обеспечению.</p>	<p>Знать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного обеспечения; основные методы и средства испытаний и оценки качества программного обеспечения, основные существующие стандарты качества программного обеспечения (ПК-5.3)</p> <p>Уметь проводить отладку и тестирование программных модулей, строить современные автоматизированные системы обработки информации, составлять эксплуатационную документацию на ПО (ПК-5.3)</p> <p>Владеть инструментами по отладке программного обеспечения, навыком и</p>	

		<p>инструментами отладки и тестирования программного обеспечения; навыком использования программных средств автоматизации тестирования и измерения характеристик программного обеспечения; инструментами по управлению контролю версий ПО; (ПК-5.3)</p>	
	<p>ПК-5.2 Осуществляет интеграцию программных модулей и компонент и проверку работоспособности программного обеспечения.</p>	<p>Знать модели и этапы жизненного цикла программного обеспечения характеристики качества программного обеспечения; методы разработки требований к программному обеспечению; тенденции развития современных технологий программирования; основные стандарты в области инфокоммуникационных систем и технологий, в том числе стандарты Единой системы программной документации; основы объектно-ориентированного подхода к программированию (ПК-5.2) Уметь проводить системный анализ моделируемой системы; разрабатывать спецификации требований к программному обеспечению; использовать методы проектирования и производства программного продукта (ПК-5.2) Владеть навыками проведения имитационного моделирования; методиками анализа результатов моделирования; навыком анализа и разработки спецификации требований к программному обеспечению; навыком использования программных средств автоматизации спецификации и документирования программных компонент (ПК-5.2)</p>	

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 часов.

4.1. Форма обучения: очная

Уровень базового образования: среднее общее.

Срок обучения 4г.

4.1.1. Структура дисциплины

№ п/п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником						Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)	
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация			Контроль
1	Задача проектирования программных систем	7	2		4					20	устный опрос, сдача отчета по лабораторной работе
2	Организация процесса проектирования программного обеспечения	7	8	16	4					20	устный опрос, сдача отчета по лабораторной и практической работам
3	Декомпозиция, абстракция, спецификация	7	6		4					16	устный опрос, сдача отчета по лабораторной работе
4	Управление разработкой ПО	7			4					2,15	устный опрос, сдача отчета по лабораторной работе
Всего за семестр		108	16	16	16			1,6	0,25	58,15	Зач.
5	Обеспечение качества ПС	8	6		8					13,85	устный опрос, сдача отчета по лабораторной работе
6	Методы защиты программ и данных	8	4		4					9	устный опрос, сдача отчета по лабораторной работе
7	Проектирование интерфейса с пользователем	8	2							30,95	устный опрос
Всего за семестр		108	12		12			3,2	0,35	53,8	Экз.(26,65)
Итого		216	28	16	28			4,8	0,6	111,95	26,65

4.1.2. Содержание дисциплины

4.1.2.1. Перечень лекций

Семестр 7

Раздел 1. Задача проектирования программных систем

Лекция 1.

Введение. Задача проектирования программных систем. Основные понятия и определения. Модели ЖЦ ПС (2 часа).

Раздел 2. Организация процесса проектирования программного обеспечения

Лекция 2.

Проектирование ПО. Стандарты (2 часа).

Лекция 3.

Требования к ПС. Функциональная спецификация (2 часа).

Лекция 4.

Методы проектирования структуры. Декомпозиция и абстракция (2 часа).

Лекция 5.

Компьютерная поддержка разработки сопровождения ПС (2 часа).

Раздел 3. Декомпозиция, абстракция, спецификация

Лекция 6.

Спецификации процедурных абстракций (2 часа).

Лекция 7.

Спецификации абстракций данных (2 часа).

Лекция 8.

Формальные спецификации. Языки спецификаций (2 часа).

Семестр 8

Раздел 5. Обеспечение качества ПС

Лекция 9.

Качество ПС. Спецификация качества (2 часа).

Лекция 10.

Тестирование и отладка. Основы доказательства правильности (2 часа).

Лекция 11.

Управление разработкой ПО (2 часа).

Раздел 6. Методы защиты программ и данных

Лекция 12.

Документирование ПС (2 часа).

Лекция 13.

Методы защиты программ и данных (2 часа).

Раздел 7. Проектирование интерфейса с пользователем

Лекция 14.

Проектирование интерфейса с пользователем (2 часа).

4.1.2.2. Перечень практических занятий

Семестр 7

Раздел 2. Организация процесса проектирования программного обеспечения

Практическое занятие 1

Определение требований к ПО. Анализ потребностей клиента. Определение классов пользователей (2 часа).

Практическое занятие 2

Проектирование автоматизированной системы (2 часа).

Практическое занятие 3

Определение образа и границ проекта (2 часа).

Практическое занятие 4

Моделирование требований. Диаграммы вариантов использования (2 часа).

Практическое занятие 5

Моделирование требований. Диаграммы потоков данных (2 часа).

Практическое занятие 6

Диаграммы «сущность-связь». Диаграмма переходов состояний (2 часа).

Практическое занятие 7

Спецификация требований к автоматизированной системе. Документирование требований (2 часа).

Практическое занятие 8

Объектно-ориентированный подход к разработке ПО. Диаграммы классов (2 часа).

4.1.2.3. Перечень лабораторных работ

Семестр 7

Раздел 1. Задача проектирования программных систем

Лабораторная 1.

Построение функциональной модели (SADT). Моделирование потоков данных (DFD) (4 часа).

Раздел 2. Организация процесса проектирования программного обеспечения

Лабораторная 2.

Построение объектно-ориентированной модели (UML) (4 часа).

Раздел 3. Декомпозиция, абстракция, спецификация

Лабораторная 3.

Спецификация и процедурная абстракция (4 часа).

Раздел 4. Управление разработкой ПО

Лабораторная 4.

Автономное тестирование программного модуля (4 часа).

Семестр 8

Раздел 5. Обеспечение качества ПС

Лабораторная 5.

Системы управления версиями ПО (4 часа).

Лабораторная 6.

Управление разработкой (4 часа).

Раздел 6. Методы защиты программ и данных

Лабораторная 7.

Самодокументирование программ (4 часа).

4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Задача проектирования программных систем.
2. Организация процесса проектирования программного обеспечения.
3. Сопровождение.
4. Декомпозиция, абстракция, спецификация.
5. Модель зрелости процессов создания ПО.
6. Обеспечение качества ПС.
7. Выявление и уменьшение рисков.
8. Проектирование справочных систем и сообщений.
9. Метрики проектирования пользовательского интерфейса.
10. Применение шаблонов в проектировании ПО.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Не планируется.

4.1.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

5. Образовательные технологии

В процессе изучения дисциплины применяется контактная технология преподавания (за исключением самостоятельно изучаемых студентами вопросов). При проведении лабораторных работ применяется имитационный или симуляционный подход. Шаги решения задач студентам демонстрируются при помощи мультимедийной техники. В дальнейшем студенты самостоятельно решают аналогичные задания.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Фонды оценочных материалов (средств) приведены в приложении.

7. Учебно-методическое и информационное обеспечение дисциплины.

7.1. Основная учебно-методическая литература по дисциплине

1. Кулямин, В. В. Технологии программирования. Компонентный подход : учебное пособие / В. В. Кулямин. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 590 с. — ISBN 978-5-4497-0884-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/102071.html> (дата обращения: 23.10.2021). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/102071.html> - <https://www.iprbookshop.ru/102071.html>

2. Терехов, А. Н. Технология программирования : учебное пособие / А. Н. Терехов. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 148 с. — ISBN 978-5-4497-0702-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/97587.html> (дата обращения: 07.11.2020). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/97587.html>

3. Грекул, В. И. Проектирование информационных систем : учебное пособие / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 299 с. — ISBN 978-5-4497-0689-8. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/97577.html> (дата обращения: 07.11.2020). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/97577.html>

4. Кознов, Д. В. Введение в программную инженерию : учебное пособие / Д. В. Кознов. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 305 с. — ISBN 978-5-4497-0311-8. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89428.html> (дата обращения: 23.10.2021). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/102071.html> - <https://www.iprbookshop.ru/102071.htm>

5. Куклина, И. Г. Методы и средства проектирования информационных систем : учебное пособие / И. Г. Куклина, К. А. Сафонов. — Нижний Новгород : Нижегородский государственный архитектурно-строительный университет, ЭБС АСВ, 2020. — 84 с. — ISBN 978-5-528-00419-8. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/107378.html> (дата обращения: 23.10.2021). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/107378.html> - <https://www.iprbookshop.ru/107378.html>

6. Артюшина Л. А. Технологии и методы программирования: учебное пособие для студентов вузов / Л. А. Артюшина, А. А. Воронина [электронный ресурс] . - 2013. - <http://e.lib.vlsu.ru:80/handle/123456789/2516>

7. Технология программирования: Практикум для студентов образовательной программы 09.03.01 Информатика и вычислительная техника / сост. Бейлекчи Д.В.

[Электронный ресурс]. – Электрон. текстовые дан. (2,1 Мб). - Муром.: МИ ВлГУ, 2016. - <https://www.mivlgu.ru/iop/mod/resource/view.php?id=8859>

7.2. Дополнительная учебно-методическая литература по дисциплине

1. Дерябкин, В. П. Проектирование информационных систем по методологии UML с использованием Qt-технологии программирования : учебное пособие / В. П. Дерябкин, В. В. Козлов. — Самара : Самарский государственный технический университет, ЭБС АСВ, 2017. — 156 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/83601.html> (дата обращения: 07.11.2020). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/83601.html>

2. Леоненков, А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose : учебное пособие / А. В. Леоненков. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 317 с. — ISBN 978-5-4497-0667-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/97554.html> (дата обращения: 07.11.2020). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/97554.html>

3. Непейвода, Н. Н. Стили и методы программирования : учебное пособие / Н. Н. Непейвода. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 295 с. — ISBN 978-5-4497-0938-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/102065.html> (дата обращения: 23.10.2021). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/102065.html>

4. Сеницын, С. В. Верификация программного обеспечения : учебное пособие / С. В. Сеницын, Н. Ю. Налютин. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 367 с. — ISBN 978-5-4497-0653-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/97540.html> (дата обращения: 23.10.2021). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/97540.html>

7.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института (www.mivlgu.ru/iop), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;
- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:

1. Лекции по дисциплине "Технология программирования" / Сайт кафедры системного программирования факультета ВМК МГУ: [Электронный ресурс]. Режим доступа: <http://sp.cmc.msu.ru/info/3/techprog.htm>

2. Интернет-портал НОУ «ИНТУИТ» (курсы по тематике дисциплины): [Электронный ресурс]. Режим доступа: <http://www.intuit.ru/>

3. Информационно-образовательный портал МИ ВлГУ <https://www.mivlgu.ru/iop/>

4. Электронная библиотечная система «IPRbooks» <http://www.iprbookshop.ru>

5. Электронная библиотека ВлГУ <http://e.lib.vlsu.ru/>

6. Электронная библиотека «ЭВРИКА» <http://elib.mivlgu.local/>

7. Научная электронная библиотека "eLibrary" <http://elibrary.ru>
Программное обеспечение:
Ramus Educational (freeware);
StarUML (freeware).
Draw.io standalone (freeware).
Doxygen (freeware)
Tortoise Hg (Mercurial) (freeware)
Git / Tortoise Git (freeware)

7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

iprbookshop.ru
e.lib.vlsu.ru:80
mivlgu.ru
sp.cmc.msu.ru
intuit.ru
e.lib.vlsu.ru
elib.mivlgu.local
elibrary.ru
mivlgu.ru/iop

8. Материально-техническое обеспечение дисциплины

Лаборатория систем автоматизированного проектирования
Компьютеры Kraftway Credo KC 36; Проектор ACER P1100 DLP Projector EMEA;
Экран настенный; Акустическая система; Интерактивная доска Hitachi StarBoard FX-82W.

9. Методические указания по освоению дисциплины

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя, каким дополнительным пособиям следует отдать предпочтение; ведет конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями.

На практических занятиях пройденный теоретический материал подкрепляется решением задач по основным темам дисциплины. Занятия проводятся в компьютерном классе, используя специальное программное обеспечение. Каждой подгруппе обучающихся преподаватель выдает задачу, связанную с разработкой и программной реализацией алгоритмов обработки информации. В конце занятия обучающие демонстрируют полученные результаты преподавателю и при необходимости делают работу над ошибками.

До выполнения лабораторных работ обучающийся изучает соответствующий раздел теории. Перед занятием студент знакомится с описанием заданий для выполнения работы, внимательно изучает содержание и порядок проведения лабораторной работы. Лабораторная работа проводится в компьютерном классе. Обучающиеся выполняют индивидуальную задачу компьютерного моделирования в соответствии с заданием на лабораторную работу. Полученные результаты исследований сводятся в отчет и защищаются по традиционной методике в классе на следующем лабораторном занятии. Необходимый теоретический материал, индивидуальное задание, шаги выполнения лабораторной работы и требование к отчету приведены в методических указаниях, размещенных на информационно-образовательном портале института.

Самостоятельная работа оказывает важное влияние на формирование личности будущего специалиста, она планируется обучающимся самостоятельно. Каждый обучающийся самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием дисциплины. Он выполняет внеаудиторную работу и изучение разделов, выносимых на самостоятельную работу, по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Форма заключительного контроля при промежуточной аттестации – экзамен. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению *09.03.01 Информатика и вычислительная техника* и профилю подготовки *Вычислительные машины, комплексы, системы и сети*
Рабочую программу составил *к.т.н., преподаватель Бейлекчи Д.В.*_____

Программа рассмотрена и одобрена на заседании кафедры *ЭиВТ* протокол № 24 от 27.05.2020 года.
Заведующий кафедрой *ЭиВТ* _____ *Кропотов Ю.А.*
(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета ФРЭКС
протокол № 9 от 11.06.2020 года.
Председатель комиссии ФРЭКС _____ Белов А.А.
(Подпись)

Лист актуализации рабочей программы дисциплины

Программа одобрена на 2021/2022 учебный год.

Протокол заседания кафедры № 32 от 19.05.2021 года.

Заведующий кафедрой ЭиВТ _____ Белов А.А.
(Подпись)

Программа одобрена на 2022/2023 учебный год.

Протокол заседания кафедры № 34 от 11.05.2022 года.

Заведующий кафедрой ЭиВТ _____ Белов А.А.
(Подпись)

Программа одобрена на _____ учебный год.

Протокол заседания кафедры № _____ от _____ 20__ года.

Заведующий кафедрой _____
(Подпись) _____ (Ф.И.О.)

Фонд оценочных материалов (средств) по дисциплине
Технология разработки программного обеспечения

1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине

Список вопросов для устного опроса:

1. Дайте определение терминам «система», «системное проектирование».
2. Объясните аббревиатуры SADT, IDEF0.
3. Какие типы связей возможны между функциями. Подробно расскажите о наиболее желательных из них.
4. Каково назначение DFD?
5. Какие элементы присутствуют на диаграмме потоков данных? Объясните их назначение и графическое представление.
6. Как рекомендуют именовать элементы диаграммы?
7. Что такое миниспецификация. Когда и зачем её применяют?
8. Что такое «вариант использования»?
9. Какие секции входят в графическое обозначение класса?
10. Какие виды связей могут быть использованы на диаграмме классов и каково их назначение?
11. Какие уровни видимости поддерживаются в UML? Поясните их смысл.
12. В каких случаях используют диаграммы состояний?
13. Какие диаграммы применяют для описания поведения ПО?
14. В чём состоит назначение абстракций?
15. От чего мы абстрагируемся, пользуясь абстракцией через параметризацию?
16. Объясните назначение абстракции через спецификацию.
17. Какие преимущества предоставляет абстракция через спецификацию?
18. Объясните назначение процедурных абстракций.
19. Перечислите свойства спецификаций.
20. Что такое отладка программного средства?
21. Что такое тестирование программного средства?
22. Что является целью тестирования программ?
23. Возможно ли исчерпывающее тестирование?
24. Какие задачи не решает тестирование?
25. Что такое автономная отладка программного средства?
26. Что такое ведущий отладочный модуль?
27. Что такое отладочный имитатор программного модуля?
28. Что подразумевают под понятием стиль? Что входит в понятие стиль?
29. Что такое репозиторий в системах управления версиями?
30. Что такое процедура слияния в системах управления версиями?
31. В чём заключаются правила стандартизации стиля? Что считается хорошим стилем в оформлении программ?
32. Что определяет стиль оформления текстов модулей?
33. Почему отсутствие комментариев в программе называют худшей ошибкой программиста?
34. Что такое система управления версиями?
35. Какие функции обеспечивают системы управления версиями?
36. Какие бывают виды систем управления версиями?
37. Какие системы управления версиями наиболее популярны в настоящее время?
38. С чего начинается процесс моделирования в SADT?
39. Каких правил надо придерживаться при SADT-моделировании?
40. Поясните назначение итеративного рецензирования.
41. Какие признаки говорят о необходимости декомпозиции текущей диаграммы? Какие правила надо соблюдать при детализации?

42. Какие существуют критерии завершения процесса декомпозиции?
43. Как строится диаграмма вариантов использования и какую информацию она содержит?
44. Как отображаются действия на диаграмме состояний?
45. Как задать условие перехода на диаграмме состояний?
46. Перечислите элементы диаграммы состояний покажите их графическое изображение, поясните их назначение.
47. Какого порядка действий желательно придерживаться при построении спецификации.
48. Какую роль играют процедурные абстракции при построении абстракции данных?
49. Из чего состоит спецификация абстракции данных.
50. Из чего состоит абстракция данных? В чём преимущества типа данных в сравнении с набором подпрограмм, работающих с данными конкретного типа?
51. Операции каких категорий присутствуют в абстракции данных?
52. Каких правил рекомендуют придерживаться при именовании объектов? Приведите пример ошибок стилового оформления программ?
53. Каких правила необходимо придерживаться при составлении пояснительных комментариев?
54. Какие функции обеспечивает распределенная система управления версиями?
55. Перечислите элементы SADT-модели, их назначение, графическое представление.
56. В чём состоит метод составления спецификаций через пары утверждений?
57. Как проверить полноту типа данных?
58. Перечислите основные пункты освещаемые в вводных комментариях?
59. Перечислите методы функционального тестирования.
60. Перечислите методы структурного тестирования.
61. Перечислите методы работы с системами документирования исходного кода
62. Перечислите методы работы с системами контроля версий.

Общее распределение баллов текущего контроля по видам учебных работ для студентов

Рейтинг-контроль 1	Устный опрос	до 20
Рейтинг-контроль 2	Устный опрос	до 30
Рейтинг-контроль 3	Устный опрос	до 30
Посещение занятий студентом	контроль посещаемости	до 15
Дополнительные баллы (бонусы)	за своевременную защиту всех лабораторных	до 5
Выполнение семестрового плана самостоятельной работы	нет	0

2. Промежуточная аттестация по дисциплине

Перечень вопросов к экзамену / зачету / зачету с оценкой.

Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)

Примерные тестовые вопросы для промежуточной аттестации студентов на экзамене.

ПК-5:

Блок 1 (знать).

Внешним описанием ПС называют

- документ, определяющий назначение ПС характеристики качества
- документ, содержащий всю информацию, которую необходимо знать пользователю для применения ПС
- документ, определяющий задачи разработчиков ПС
- все три варианта

К способам разработки определения требований к ПС относят

- управляемая пользователем разработка
- контролируемая пользователем разработка
- независимая от пользователя разработк
- все варианты

Под жизненным циклом ПС понимают

- период разработки и эксплуатации, начиная от момента возникновения замысла ПС и заканчивая прекращением всех видов его использования
- момент возникновения ПС и момент непосредственной его эксплуатации
- использование ПС для решения практических задач на компьютере путем выполнения ее программ
- хранение, внедрение и сопровождение ПС

Свойства процедурных абстракций:

- Локальность -> каждая реализация может быть рассмотрена независимо от остальных
- Модифицируемость -> одна реализация может быть заменена другой без изменения других программ
- Простота -> подпрограмма должна обладать определённым и легко объяснимым назначением, независимым от контекста её использования

Критерии качества программных систем:

- мера, характеризующая способность ПС выполнять возложенные на него функции в течение определенного отрезка времени. (Временная эффективность)
- мера, характеризующая способность ПС выполнять возложенные на него функции при определенных ограничениях на используемые ресурсы (используемую память). (Эффективность по ресурсам)
- свойство, характеризующее программы ПС с точки зрения организации взаимосвязанных их частей в единое целое определенным образом (Структурированность)

Свойства процедурных абстракций:

- каждая реализация может быть рассмотрена независимо от остальных (Локальность)
- одна реализация может быть заменена другой без изменения других программ (Модифицируемость)
- подпрограмма должна обладать определённым и легко объяснимым назначением, независимым от контекста её использования (Простота)

Критерии качества программных систем:

- свойство, характеризующее способность ПС продолжать корректное функционирование, несмотря на задание неправильных (ошибочных) входных данных. -> Устойчивость
- свойство, характеризующее способность ПС противостоять преднамеренным или нечаянным деструктивным (разрушающим) действиям пользователя.-> Защищенность
- свойство, характеризующее наличие в составе ПС информации, необходимой и достаточной для понимания назначения ПС, принятых предположений, существующих ограничений, входных данных и результатов работы отдельных компонент, а также текущего состояния программ в процессе их функционирования. -> Информативность

Критерии качества программных систем:

- свойство, характеризующее способность ПС к использованию большего объема памяти для хранения данных или расширению функциональных возможностей отдельных компонент. -> Расширяемость
- свойство, характеризующее ПС с точки зрения организации его программ из таких дискретных компонент, что изменение одной из них оказывает минимальное воздействие на другие компоненты -> Модульность
- мера, характеризующая ПС с точки зрения простоты внесения необходимых изменений и доработок на всех этапах и стадиях жизненного цикла ПС. -> Модифицируемость

Критерии качества программных систем:

- свойство, характеризующее степень обладания ПС всеми необходимыми частями и чертами, требующимися для выполнения своих явных и неявных функций. -> Завершенность
- мера, характеризующая приемлемость величины погрешности в выдаваемых программами ПС результатах с точки зрения предполагаемого их использования. -> Точность
- свойство, характеризующее способность ПС выполнять предписанные функции без помощи или поддержки других компонент программного обеспечения. -> Автономность

Дисциплина, определяющая подсистемы, компоненты и способы их соединения, задающая ограничения, при которых система должна функционировать, выбирающая наиболее эффективное сочетание людей, машин и программного обеспечения для реализации системы.

- Функциональное проектирование
- Системное проектирование
- Оптимальное проектирование
- Объектное проектирование

Какие секции входят в графическое обозначение класса?

- Имя, атрибуты, операции
- Имя, зависимости, предел видимости
- Имя, атрибуты, список экземпляров класса
- Имя, список экземпляров класса, методы

Описание системы с помощью SADT называется

- Моделью
- Проектом
- Методологией
- Процессом

Комментарии к программе, которые содержат указания по вызову программы, список и назначение основных переменных, список используемых подпрограмм, сведения о времени выполнения программы, требуемый объем памяти это:

- Вводные комментарии
- Пояснительные комментарии
- Оглавление
- Процедурные комментарии

Рекомендуемое соотношение инструкция:комментарий при оформлении программы:

- 2:1
- 1:1
- 3:1
- 4:1

Набор приёмов и методов программирования, которые используют программисты, чтобы получить правильные, эффективные, удобные для применения и легко читаемые программы. Это:

- Стиль программирования
- Методология программирования
- Технология программирования
- Ничего из вышеперечисленного

Для повышения надежности программных средств снабжают программы дополнительной информацией, с использованием которой можно существенно повысить уровень контроля ПС. Такую информацию задают в форме неформализованных или формализованных утверждений, привязываемых к различным фрагментам программ. Такие утверждения называются

- Обоснованиями программы
- Комментариями программы
- Свойствами программы
- Архитектурой программы

Эти обоснования программ могут объяснять мотивы принятия тех или иных решений, что может существенно облегчить поиск и исправление ошибок, а также изучение программ при их сопровождении

- Неформализованные
- Формализованные
- Отладочные
- Вводные

Эти обоснования программы позволяют доказывать некоторые свойства программ как вручную, так и контролировать (устанавливать) их автоматически

- Неформализованные
- Формализованные
- Отладочные
- Вводные

Для доказательства свойства программы используются свойства простых операторов языка программирования и свойствами управляющих конструкций, с помощью которых строится программа из простых операторов. Эти свойства называют:

- Правилами верификации программ
- Триадами Хоара
- Предусловиями
- Постусловиями

Программное изделие – это:

- программа, расположенная на определенном носителе и являющаяся продуктом промышленного производства
- экземпляр или копия разработанного ПС
- программа плюс документация, гарантия качества, рекламные материалы, обучение, распространение и сопровождение
- все ответы правильные

Методология проектирования, соединяющая в себе процесс объектной декомпозиции и приёмы представления логической и физической, а также статической и динамической моделей проектируемой системы

- Объектно-ориентированное проектирование
- Объектно-ориентированный анализ

- Процедурно-ориентированное программирование
- Алгоритмическое программирование

Блок 2 (уметь).

Укажите соответствия между названиями элементов, присутствующими на диаграмме потоков данных, и их графическим обозначением:

- сущность
- процесс
- хранилище
- актер

Признаками сложности диаграммы потоков данных (в смысле контекста) могут быть

- наличие небольшого количества внешних сущностей
- распределённая природа системы
- многофункциональность системы с уже сложившейся или выявленной группировкой функций в отдельные подсистемы
- наличие большого количества комментариев

Автономное тестирование модуля целесообразно осуществлять в четыре последовательно выполняемых шага (перечислить шаги в порядке выполнения):

- На основании спецификации отлаживаемого модуля подготовьте тесты для каждой возможности и каждой ситуации, для каждой границы областей допустимых значений всех входных данных, для каждой области изменения данных, для каждой области недопустимых значений всех входных данных и каждого недопустимого условия -> 1
- Проверьте текст модуля, чтобы убедиться, что каждое направление любого разветвления будет пройдено хотя бы на одном тесте -> 2
- Проверьте текст модуля, чтобы убедиться, что для каждого цикла существуют тесты, обеспечивающие, по крайней мере, четыре следующие ситуации\): тело цикла не выполняется ни разу, один раз, два раза и максимальное число раз -> 3
- Проверьте текст модуля, чтобы убедиться, что существуют тесты, проверяющие чувствительность к отдельным особым значениям входных данных -> 4

Автономное тестирование модуля целесообразно осуществлять в четыре последовательно выполняемых шага.

- На основании спецификации отлаживаемого модуля подготовьте тесты для каждой возможности и каждой ситуации, для каждой границы областей допустимых значений всех входных данных, для каждой области изменения данных, для каждой области недопустимых значений всех входных данных и каждого недопустимого условия. -> 1 шаг
- Проверьте текст модуля, чтобы убедиться, что каждое направление любого разветвления будет пройдено хотя бы на одном тесте. -> 2 шаг
- Проверьте текст модуля, чтобы убедиться, что для каждого цикла существуют тесты, обеспечивающие, по крайней мере, четыре следующие ситуации\): тело цикла не выполняется ни разу, один раз, два раза и максимальное число раз. -> 3 шаг
- Проверьте текст модуля, чтобы убедиться, что существуют тесты, проверяющие чувствительность к отдельным особым значениям входных данных. -> 4 шаг

Цельная программа как основной класс архитектур программных средств

- Состоит из некоторой упорядоченной совокупности программных подсистем
- Выбирают обычно в том случае, когда ПС должно выполнять одну какую-либо ярко выраженную функцию и ее реализация не представляется слишком сложной
- Состоит из набора программ, такого, что все программы этого набора применяются к одной и той же информационной среде
- Все ответы правильные

Какие действия происходят на первом слое операционной системы

- обработка всех прерываний и выделение центрального процессора программам
- осуществляется управление страничной организацией памяти
- осуществляется связь с консолью (пультом управления) оператора
- осуществляется буферизация входных и выходных потоков данных

Какие интеллектуальные возможности человека выделяет Дейкстра при разработке ПС?

- способность к перебору
- способность к абстракции
- способность к математической индукции
- все варианты правильны

Методология программирования определяется как...

- Совокупность производственных процессов, приводящую к созданию требуемого программного средства, а также описание этой совокупности процессов.
- Совокупность механизмов, применяемых в процессе разработки программного обеспечения и объединенных одним общим философским подходом.
- Ни один вариант не подходит
- Оба определения правильны

Программная документация позволяет понять

- какие функции выполняет та или иная программа
- как подготовить исходные данные и запустить требуемую программу в процесс ее выполнения
- каков эффект выполнения этой программы
- все варианты правильны

На диаграмме потоков данных информационные потоки порождают

- Внешние сущности
- Работы
- Хранилища данных
- Подсистемы

На диаграмме потоков данных за преобразование информации отвечает

- Процесс
- Подсистема
- Внешняя сущность
- Поток данных

На диаграмме потоков данных определяет информацию, передаваемую через некоторое соединение от источника к приемнику

- Поток данных
- Хранилище данных
- Внешняя сущность
- Подсистема

В настоящее время критериями качества ПС не принято считать

- функциональность
- легкость применения
- сопровождаемость
- статичность

Диаграмма, позволяющая проследить, каким образом происходит обмен информацией как между бизнес-функциями внутри системы, так и самой системы с внешней информационной средой называется

- Диаграмма потоков данных
- Диаграмма последовательностей
- Диаграмма классов
- Диаграмма состояний

Управляемая пользователем разработка определения требований к ПС

- требования к ПС формулируются разработчиком при участии представителя пользователей
- требований к ПС определяются заказчиком, представляющим организацию пользователей
- требования к ПС утверждаются представителем пользователя
- требования к ПС определяются без какого-либо участия пользователя (на полную ответственность разработчика)

Для моделирования требований к системе и для моделирования контекста системы используют

- Диаграммы вариантов использования
- Диаграммы взаимодействия
- Диаграммы прецедентов
- Диаграммы Сущность-Связь

С-документированность как примитива качества ПС:

- свойство, характеризующее наличие, полноту, понятность, доступность и наглядность учебной, инструктивной и справочной документации, необходимой для применения ПС
- свойство, характеризующее с точки зрения наличия документации, отражающей требования к ПС и результаты различных этапов разработки данной ПС, включающие возможности, ограничения и другие черты ПС, а также их обоснование
- свойство, характеризующее программы ПС с точки зрения организации взаимосвязанных их частей в единое целое определенным образом
- свойство, характеризующее наличие в составе ПС информации, необходимой и достаточной для понимания назначения ПС, принятых предположений, существующих ограничений, входных данных и результатов работы отдельных компонент, а также текущего состояния программ в процессе их функционирования

Наиболее желательные связи между функциями в SADT-модели

- Логическое связывание
- Временная связность
- Последовательная
- Функциональная

Стиль оформления программы включает (отметьте нужное):

- правила именования объектов программы (имён подпрограмм, переменных, типов)
- стиль оформления текстов модулей
- правила оформления модулей
- правила проектирования интерфейса
- правила расстановки комментариев к программе

Блок 3 (владеть).

В результате разработки проекта с помощью UML CASE-средства Rational Rose формируются следующие документы (отметьте соответствие):

- для моделирования статической структуры классов и связей между ними -> Classes Diagram
- для моделирования процесса обмена сообщениями между объектами (диаграммы взаимодействия) -> Sequence & Collaboration Diagram
- для моделирования поведения объектов системы при переходе из одного состояния в другое -> State Diagram
- для моделирования иерархии подсистем системы и физической архитектуры системы (диаграммы реализации) -> Component & Deployment Diagram

В результате разработки проекта с помощью UML CASE-средства Rational Rose формируются следующие документы (отметьте соответствие):

- для моделирования требований к системе -> Use Case Diagram - диаграммы вариантов использования (прецедентов)
- для моделирования поведения системы в рамках различных вариантов использования или моделирования деятельности -> Activity Diagram – диаграммы деятельности
- для моделирования статической структуры классов и связей между ними -> Classes Diagram – диаграммы классов
- для моделирования процесса обмена сообщениями между объектами (диаграммы взаимодействия) -> Sequence & Collaboration Diagram – диаграммы последовательностей и коопераций

В результате разработки проекта с помощью UML CASE-средства Rational Rose формируются следующие документы:

- диаграммы классов – для моделирования статической структуры классов и связей между ними -> Classes Diagram
- диаграммы последовательностей и коопераций – для моделирования процесса обмена сообщениями между объектами (диаграммы взаимодействия) -> Sequence & Collaboration Diagram
- диаграммы состояний – для моделирования поведения объектов системы при переходе из одного состояния в другое -> State Diagram

В результате разработки проекта с помощью UML CASE-средства Rational Rose формируются следующие документы:

- диаграммы вариантов использования (прецедентов) для моделирования требований к системе -> Use Case Diagram
- диаграммы деятельности для моделирования поведения системы в рамках различных вариантов использования или моделирования деятельности -> Activity Diagram
- диаграммы классов – для моделирования статической структуры классов и связей между ними -> Classes Diagram

Процесс выполнения его программ на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ. Это ... (Тестирование)

Решение о завершении детализации процесса и использовании миниспецификации при построении диаграммы потоков данных принимается аналитиком исходя из следующих критериев:

- наличие у процесса большого количества входных и выходных потоков данных
- возможности описания преобразования данных процессом в виде последовательного алгоритма

- выполнения процессом единственной логической функции преобразования входной информации в выходную
- нет возможности описания логики процесса при помощи миниспецификации небольшого объема

Какие стадии жизненного цикла ПС различают в рамках водопадного подхода:

- Разработка ПС, производство программных изделий и эксплуатация
- Разработка ПС, сборка программ из компонент и эксплуатация
- Разработка ПС, кодирование ПС, производство программных изделий и применение
- все ответы правильные

Что не включает в себя сопровождение ПС:

- доработка и модификации
- извещения пользователей о внесенных изменениях
- генерация и/или воспроизведение программ и программных документов ПС
- сбора информации о качестве ПС в эксплуатации

Какой формулой можно выразить структуру внешнего описания ПС?

- Внешнее описание ПС = определение требований + спецификация качества ПС+ функциональная спецификация ПС
- Внешнее описание ПС = определение требований + спецификация качества ПС + список тестов
- Внешнее описание ПС = спецификация качества ПС+ определение требований + функциональная спецификация ПС
- Внешнее описание ПС = техническая спецификация ПС+ спецификация качества ПС+ функциональная спецификация

Какой формулой можно выразить структуру внешнего описания ПС?

- Внешнее описание ПС = определение требований+ спецификация качества ПС+ функциональная спецификация ПС
- Внешнее описание ПС = определение требований+ спецификация качества ПС+ список тестов
- Внешнее описание ПС = спецификация качества ПС+ список тестов+ функциональная спецификация ПС
- Внешнее описание ПС = техническая спецификация ПС+ спецификация качества ПС+ функциональная спецификация

Какой формулой можно выразить структуру внешнего описания ПС

- Внешнее описание ПС = определение требований + спецификация качества ПС+ функциональная спецификация ПС
- Внешнее описание ПС = определение требований + спецификация качества ПС
- Внешнее описание ПС = определение требований + прототип ПС+ функциональная спецификация ПС
- нет правильного ответа

Выполнение оператора цикла ПОКА Q ДЕЛАТЬ S ВСЕ ПОКА завершается когда целочисленная функция F, зависящая от состояния информационной среды, удовлетворяет следующим условиям:

- если для данного состояния информационной среды истинен предикат Q, то ее значение положительно
- она убывает при изменении состояния информационной среды в результате выполнения оператора S
- после каждой итерации значение F уменьшается
- значение F перед выполнением цикла больше значения F после выполнения цикла

В языке UML определены три уровня видимости

- public — любой внешний класс, который «видит» данный, может пользоваться его общими свойствами
- protected — только любой потомок данного класса может пользоваться его защищёнными свойствами
- private — только данный класс может пользоваться этими свойствами
- blocked –внешний класс блокируется объектами, которые «видят» его общие свойства

Размер модуля определяется

- числом содержащихся в нем операторов
- числом связей
- числом содержащихся в нем строк
- мерой его зависимости по данным от других модулей

Функциональность как понятие качества программного средства:

- позволяет минимизировать усилия пользователя по подготовке исходных данных, применению ПС и оценке полученных результатов
- способность ПС выполнять набор функций, удовлетворяющих заданным или подразумеваемым потребностям пользователей
- отношение уровня услуг, предоставляемых ПС пользователю при заданных условиях, к объему используемых ресурсов
- способность ПС быть перенесенным из одной среды (окружения) в другую, в частности, с одной ЭВМ на другую

ПК-9:

Блок 1 (знать).

Методология, при которой требования к системе воспринимаются с точки зрения классов и объектов, выявленных в предметной области

- Объектно-ориентированный анализ
- Объектно-ориентированное проектирование
- Системное проектирование
- Полиморфизм

Для точного определения логики сценария выполнения прецедента используются:

- Диаграммы последовательностей
- Диаграмма классов
- Диаграмма взаимодействия
- Все три типа диаграмм

Эта абстракция позволяет представить фактически неограниченный набор различных вычислений одной программой, которая есть абстракция всех этих наборов

- Абстракция через параметризацию
- Абстракция через спецификацию
- Абстракция данных
- Ничего из вышеперечисленного

Эта абстракция позволяет абстрагироваться от процесса вычислений, описанных в теле процедуры, до уровня знания лишь того, что данная процедура должна в итоге реализовать

- Абстракция через параметризацию

- Абстракция через спецификацию
- Абстракция данных
- Ничего из вышеперечисленного

Эта абстракция состоит из набора объектов и набора операций, характеризующих поведение этих объектов

- Абстракция данных
- Абстракция через параметризацию
- Абстракция через спецификацию
- Ничего из вышеперечисленного

Какой подход не относится к основным подходам к организации процесса создания и использования ПС

- Водопадный подход
- Прототипирование
- Комплексное программирование
- Исследовательское программирование

Прототипирование как основной подход к организации процесса создания и использования ПС:

- Предполагает, что ПС конструируется, главным образом, из компонент, которые уже существуют
 - Разработка формальных спецификаций ПС и превращение их в программы путем корректных преобразований
 - моделирует начальную фазу исследовательского программирования вплоть до создания рабочих версий программ, предназначенных для проведения экспериментов с целью установить требования к ПС
 - Предполагает быструю реализацию рабочих версий программ ПС, выполняющих лишь в первом приближении требуемые функции

Способность ПС безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью это

- Качественное ПС
- Идеальное ПС
- Надежное ПС
- Стабильное ПС

Надежность является неотъемлемым атрибутом ПС, рассматривают технологию программирования как технологию разработки надежных ПС. Это означает, что ...

- Рассматривают все процессы разработки ПС, начиная с момента возникновения замысла ПС.
 - Важны вопросы не только построения программных конструкций, но и вопросы описания функций и принимаемых решений с точки зрения их человеческого (неформального) восприятия.
 - В качестве продукта технологии принимается надежная (далеко не всегда правильная) ПС.
 - Все варианты правильны

Диаграмма декомпозиции это:

- Диаграмма, состоящая из одного блока и его дуг и определяющая границы системы
- Диаграмма, описывающая каждую подсистему и их взаимодействие
- Диаграмма с сопроводительной документацией

- Инструмент разработки моделей данных, обеспечивающий стандартный способ определения данных и отношений между ними

Для описания документооборота и обработки информации в системе используется

- Диаграммы потоков данных
- Функциональная модель
- Диаграмма вариантов использования
- Диаграмма последовательностей

Совокупность взаимодействующих компонент и взаимосвязей между ними.

- Система
- Субъект
- Объект
- Проект

Для описания документооборота и обработки информации в системе используется:

- Диаграммы потоков данных
- Функциональная модель
- Диаграмма вариантов использования
- Диаграмма последовательностей

Документ, достаточно точно определяющий задачи разработчиков ПС:

- спецификация ПС
- определение требований к ПС
- внешнее описание ПС
- прототип ПС

Упрощенная версия требуемого ПС для выявления действительных потребностей пользователей:

- прототип ПС
- спецификация ПС
- внешнее описание ПС
- нет правильного ответа

Модели, ориентированные на объекты системы:

- Модели данных
- Функциональные модели
- Классы
- Программы

Стандартизованный набор достаточно простых свойств ПС, однозначно интерпретируемых разработчиками называется:

- критерии качества
- примитивы качества
- спецификации качества
- все ответы правильные

Концептуальная схема базы данных, описание сети датчиков и приборов, которой должна управлять разрабатываемая ПС являются примером одной из трех частей функциональной спецификации:

- описания внешней информационной среды
- описание нежелательных (исключительных) ситуаций, которые могут возникнуть при выполнении программ ПС
- определение функций ПС

- нет правильных ответов

Что значит способность человека к абстракции?

- Возможностью последовательного переключения внимания с одного предмета на другой
- Возможность замены множества элементов одним
- Возможность разбиения элементов
- Способность быстрого восприятия текстовой информации

Завершимость выполнения программы - это ...

- Отсутствие в ней заикливания при тех или иных исходных данных
- Когда функция убывает при изменения состояния информационной среды в результате выполнения оператора S
- Когда целочисленная функция, зависящая от состояния информационной среды убывает при изменении состояния информационной среды в результате выполнения
- Ничего из перечисленного

В состав функциональной модели в SADT входят:

- Функциональный блок
- Интерфейсная дуга
- Декомпозиция
- Глоссарий
- Абстракция

Контекстная диаграмма это:

- Диаграмма, состоящая из одного блока и его дуг и определяющая границы системы
- Диаграмма декомпозиции
- Диаграмма, описывающая каждую подсистему и их взаимодействие
- Диаграмма с сопроводительной документацией

Блок 2 (уметь).

В диаграммах вариантов использования диалог между активным субъектом и системой, а также отображение функции системы, предоставляемые в распоряжение субъекта, позволяют моделировать

- Прецеденты
- Актеры
- Сущности
- Хранилища

Дефектом программы можно назвать

- Частный случай ошибки в ПС, когда программа не соответствует своей функциональной спецификации
- Полное отсутствие функциональной спецификации
- Ошибка, после которой программа больше не может выполнять ни одной из своих функций
- Все варианты правильны

В диаграммах вариантов использования последовательность выполняемых системой транзакций, которая приводит к получению некоего осязаемого результата, в котором заинтересован определённый активный субъект это

- Вариант использования
- Прецедент
- Алгоритм

- Блок-схема

В функциональную спецификацию входит:

- описания внешней информационной среды, к которой должны применяться программы разрабатываемой ПС
- описание нежелательных (исключительных) ситуаций, которые могут возникнуть при выполнении программ ПС, и реакций на эти ситуации, которые должны обеспечить соответствующие программы
- Прототип ПО
- Ничего из перечисленного

С чего начинается процесс моделирования в SADT?

- с определения контекста
- с определения наиболее абстрактного уровня описания системы в целом
- с изучения прототипа
- определения поведения системы

Пусть информационная среда IS состоит из переменной X и остальной части информационной среды RIS: $IS = (X, RIS)$. Тогда имеет место свойство: $Q(F(X, RIS), RIS)$ $X := F(X, RIS)$ $Q(X, RIS)$, где $F(X, RIS)$ – некоторая однозначная функция, Q – предикат. К какому свойству операторов относится теорема?

- Для пустого оператора
- Для оператора присваивания
- Свойство следования
- Свойство разветвления

Модель, представляющая собой систему функций с требуемой степенью детализации, при этом функции отражают свои взаимоотношения через объекты системы:

- Функциональная модель
- Модель данных
- Информационная модель
- Математическая модель

В функциональную спецификацию входит:

- описания внешней информационной среды, к которой должны применяться программы разрабатываемой ПС
- описание нежелательных (исключительных) ситуаций, которые могут возникнуть при выполнении программ ПС, и реакций на эти ситуации, которые должны обеспечить соответствующие программы.
- Ничего из перечисленного

Могут ли формализованные обоснования доказывать некоторые свойства программ как в ручную, так и контролировать их автоматически?

- Не могут
- Могут
- Могут контролировать их только автоматически
- Могут только вручную

Методология, при которой требования к системе воспринимаются с точки зрения классов и объектов, выявленных в предметной области:

- Объектно-ориентированный анализ
- Объектно-ориентированное проектирование
- Системное проектирование

- Полиморфизм

Как доказать завершенность программы?

- Достаточно уметь доказывать завершенность оператора цикла
- Достаточно уметь доказывать справедливость цикла
- Достаточно уметь доказывать предикат, для которого справедлива импликация
- Все три варианта

Изучение и проверка разработчиками архитектуры ПС и текстов программ, а также разработчиками документации по применению и разработчиками комплекта тестов, называют...

- Смежный контроль внешнего описания сверху
- Смежный контроль внешнего описания снизу
- Ручная имитация
- Статистический просмотр

Изучение и проверка разработчиками архитектуры ПС и текстов программ, а также разработчиками документации по применению и разработчиками комплекта тестов это

- Смежный контроль спецификации качества сверху
- Смежный контроль внешнего описания снизу
- Смежный контроль функциональной спецификации
- Смежный контроль спецификации качества сверху и снизу

С чего начинается процесс моделирования в SADT?

- с определения контекста
- с определения наиболее абстрактного уровня описания системы в целом;
- с определения функциональной спецификации
- обеспечение однозначности интерпретации представления информации

Моделирование позволяет решить следующие задачи

- визуализировать систему в её текущем или желательном для нас состоянии
- определить структуру или поведение системы
- получить шаблон, облегчающий конструирование системы
- изменить структуру системы

Блок 3 (владеть).

Информационно прочный модуль - это

- модуль, выполняющий (реализующий) одну какую-либо определенную функцию
- модуль, с зависимостью по данным от других модулей
- модуль, в котором результат (эффект) обращения к нему зависит только от значений его параметров (и не зависит от предыстории обращений к нему)
- модуль, выполняющий (реализующий) несколько операций (функций) над одной и той же структурой данных

К методам верификации программного обеспечения относятся:

- Экспертиза
- Статистический анализ
- Подготовка прототипов
- Моделирование

Представление фактов и идей в формализованном виде, пригодном для передачи и переработке в некоем процессе это

- Информация

- Данные
- Документация
- Файл

На чем основывается логический подход к спецификации семантики функций?

- Использовании равенств
- Использование предикатов
- Использовании таблиц
- Нет правильного ответа

Модели, ориентированные на объекты системы

- Модели данных
- Функциональные модели
- Классы
- Программы

Этот метод тестирования базируется на том, что все тесты основываются на спецификации системы или её компонентов. Программная реализация считается неизвестной. Это ... тестирование. (Функциональное)

Этот метод тестирования базируется на принципе: каждая команда каждой программы программного средства должна проработать хотя бы на одном тесте. Это ... тестирование. (Структурное)

Этот метод тестирования базируется на том, что все тесты основываются на спецификации системы или её компонентов. Программная реализация считается неизвестной. Это ... тестирование. (функциональное)

Этот метод тестирования базируется на принципе: каждая команда каждой программы программного средства должна проработать хотя бы на одном тесте. Это ... тестирование. (Структурное)

Подробное описание объектов системы, связанных системными функциями, это:

- Модель данных
- Функциональная модель
- Статическая модель
- Динамическая модель

В чём используется ряд средств, позволяющих фиксировать синтаксис и выразить семантику описываемых функций?

- языком спецификаций
- семантика
- формальный язык
- язык спецификаций и формальный язык

Свойство, характеризующее наличие, полноту, понятность, доступность и наглядность учебной, инструктивной и справочной документации, необходимой для применения ПС, называется...

- Документированность
- Информативность
- Точность
- Завершенность

Модель, представляющая собой систему функций с требуемой степенью детализации, при этом функции отражают свои взаимоотношения через объекты системы:

- Функциональная модель
- Модель данных
- Информационная модель
- Математическая модель

Признаками сложности диаграммы потоков данных (в смысле контекста) могут быть

- наличие небольшого количества внешних сущностей
- распределённая природа системы
- многофункциональность системы с уже сложившейся или выявленной группировкой функций в отдельные подсистемы

Подробное описание объектов системы, связанных системными функциями, это:

- Модель данных
- Функциональная модель
- Статическая модель
- Динамическая модель

В состав модели потоков данных входит

- Работы
- Внешние сущности
- Хранилища данных
- Классы

ПК-10:

Блок 1 (знать).

Методические материалы, характеризующих процедуры оценивания

Для проведения контроля успеваемости и промежуточной аттестации студентов с выставлением промежуточных результатов за соответствующие контрольные недели применяются:

- сдача отчетов по выполненным лабораторным работам;
 - сдача отчета по выполненной части практических работ,
- при этом проводится устный опрос преподавателем студентов на основе типовых вопросов, представленных в п.6.3.

Для подготовки при выполнении и к опросам при сдаче практических и лабораторных работ студентам рекомендуется пользоваться следующими методическими указаниями:

1. Методические указания для лабораторных занятий доступны по ссылке:
<https://www.mivlgu.ru/iop/mod/resource/view.php?id=15068>
2. Методические указания для практических занятий доступны по ссылке:
<https://www.mivlgu.ru/iop/mod/resource/view.php?id=15071>

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	Уровень сформированности компетенций
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки	Высокий уровень

		работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	<i>Продвинутый уровень</i>
50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	<i>Пороговый уровень</i>
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	<i>Компетенции не сформированы</i>

3. Задания в тестовой форме по дисциплине

Примеры заданий:

ПК-5.1

Дисциплина, определяющая подсистемы, компоненты и способы их соединения, задающая ограничения, при которых программная система должна функционировать, выбирающая наиболее эффективное сочетание людей, машин и программного обеспечения для реализации системы:

- Системное проектирование
- Функциональное проектирование
- Оптимальное проектирование
- Объектное проектирование

Методология проектирования, соединяющая в себе процесс объектной декомпозиции и приёмы представления логической и физической, а также статической и динамической моделей проектируемой системы

- Объектно-ориентированное проектирование
- Объектно-ориентированный анализ

- Процедурно-ориентированное программирование
- Алгоритмическое программирование

Какой подход НЕ относится к основным подходам к организации процесса создания и использования программного средства

- Комплексное программирование
- Водопадный подход
- Прототипирование
- Исследовательское программирование

Абстракция, которая позволяет представить фактически неограниченный набор различных вычислений одной программой, которая есть абстракция всех этих наборов, называется абстракция через ... (параметризацию)

Аббревиатура методологии структурного анализа и проектирования, с представлением систем в виде иерархии функций с использованием специального графического языка описания моделей систем, интегрирующая процесс моделирования, управления конфигурацией проекта и руководство проектом: ... (SADT)

ПК-5.2

Под жизненным циклом программного средства (ПС) понимают

- период разработки и эксплуатации, начиная от момента возникновения замысла ПС и заканчивая прекращением всех видов его использования
- момент возникновения ПС и момент непосредственной его эксплуатации
- использование ПС для решения практических задач на компьютере путем выполнения ее программ
- хранение, внедрение и сопровождение ПС

Программное изделие – это:

- все ответы правильные
- программное обеспечение прошедшее испытание, имеющее соответствующий комплект программных документов и готовое к серийному производству
- программа, расположенная на определенном носителе и являющаяся продуктом промышленного производства
- программа и документация с гарантией качества, рекламными материалами, обучением, способами распространения и сопровождением

На диаграмме потоков данных первичные информационные потоки порождаются

- внешними сущностями
- работами
- хранилищами данных
- подсистемами

Что НЕ включает в себя сопровождение программного средства (ПС):

- генерация или воспроизведение программ и программных документов ПС
- доработка и модификация ПС
- извещения пользователей о внесенных изменениях в ПС
- сбора информации о качестве ПС в эксплуатации

Этот метод тестирования базируется на том, что все тесты основываются на спецификации системы или её компонентов. Программная реализация считается неизвестной. Это ... (какое?) тестирование. (функциональное)

Этот метод тестирования базируется на принципе: каждая команда каждой программы программного средства должна проработать хотя бы на одном тесте. Это ... (какое?) тестирование. (структурное)

Аббревиатура языка графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур: ... (UML)

Процесс определения, выполняют ли программные средства и их компоненты требования, наложенные на них в последовательных этапах жизненного цикла разрабатываемой программной системы, называется ... (ответ в именительном падеже) (верификация)

ПК-5.3

В функциональную спецификацию программной системы входит:

- описания внешней информационной среды, к которой должны применяться программное обеспечение
- описание нежелательных (исключительных) ситуаций, которые могут возникнуть при выполнении программ, и реакций на эти ситуации, которые должны обеспечить соответствующие программы.
- критерии качества программного средства

Упрощенная версия программного средства (ПС) для выявления действительных потребностей пользователей называется ... ПС (ответ в именительном падеже): (прототип)

В настоящее время критериями качества ПС не принято считать

- статичность
- переносимость
- функциональность
- удобство применения
- сопровождаемость

Критерии качества программного обеспечения (ПО) (отметьте соответствие):

- мера, характеризующая способность ПО выполнять возложенные на него функции в течение определенного отрезка времени.
(Временная эффективность)
- мера, характеризующая способность ПО выполнять возложенные на него функции при определенных ограничениях на используемую память.
(Эффективность по ресурсам)
- свойство, характеризующее программы ПО с точки зрения организации взаимосвязанных их частей в единое целое определенным образом (Структурированность)
- ... Устойчивость

В технологии проектирования SADT диаграммы DFD определяют для предметной области потоки ... (чего?) (данных)

В результате объектно-ориентированной разработки проекта с использованием UML для моделирования поведения объектов системы при переходе из одного состояния в другое формируются диаграмма ... (чего?): (состояния)

Критерий качества, определяющий способность программного средства безотказно выполнять определенные функции при заданных условиях в течение заданного периода

времени с достаточно большой вероятностью, это – ... (ответ в именительном падеже)
(надежность)

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке <https://www.mivlgu.ru/iop/question/edit.php?courseid=648&cat=4024%2C22018&qpage=0&category=25534%2C22018&qshowtext=0&recurse=0&recurse=1&showhidden=0>

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.