



Федеральное государственное бюджетное образовательное учреждение высшего образования
«ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ ДВАЖДЫ ГЕРОЯ
СОВЕТСКОГО СОЮЗА, ЛЕТЧИКА-КОСМОНАВТА А.А. ЛЕОНОВА»

«УТВЕРЖДАЮ»

И.о. проректора

А.В. Троицкий

«__» _____ 2023г.

**ИНСТИТУТ
ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ
КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**РАБОЧАЯ ПРОГРАММА
ДИСЦИПЛИНЫ
«АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»**

Направление подготовки: 09.03.04 Программная инженерия

Направленность (профиль): Проектирование и разработка программного обеспечения

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев 2023

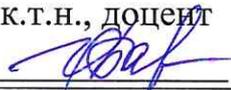
Рабочая программа является составной частью основной профессиональной образовательной программы и проходит рецензирование со стороны работодателей в составе профессиональной образовательной программы. Рабочая программа актуализируется и корректируется ежегодно.

Автор Ражева Д.П. Рабочая программа дисциплины: Архитектура программного обеспечения. – Королев МО: «Технологический университет», 2023

Рецензент: к. т. н., Г.Н. Исаева

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки бакалавров 09.03.04 Программная инженерия, учебного плана, утвержденного Ученым советом Технологического университета, Протокол № 9 от 11 апреля 2023 года.

Рабочая программа рассмотрена и одобрена на заседании кафедры:

| | | | | |
|--|--|------|------|------|
| Заведующий кафедрой (ФИО, ученая степень, звание, подпись) | Баранова О.М., к.т.н., доцент  | | | |
| Год утверждения (переподтверждения) | 2023 | 2024 | 2025 | 2026 |
| Номер и дата протокола заседания кафедры | № 7 от 03.04.2023 | | | |

Рабочая программа согласована:

Руководитель ОПОП  О.М. Баранова, к.т.н.

Рабочая программа рекомендована на заседании УМС:

| | | | | |
|--------------------------------------|-------------------------|------|------|------|
| Год утверждения (переподтверждения) | 2023 | 2024 | 2025 | 2026 |
| Номер и дата протокола заседания УМС | № 5 от 11.04.2023 г. | | | |

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОПОП

Целью изучения дисциплины является

получение обучающимися знаний и приобретение практических навыков в области конструирования программного обеспечения, а также формирование у студента навыков сбора, обработки и представления исходных данных для принятия проектных решений; разработки концептуальных, информационно-логических и функциональных моделей программного обеспечения; объектно-ориентированного анализа и проектирования.

В процессе обучения студент приобретает и совершенствует следующие компетенции:

профессиональные компетенции (ПК):

ПК-3 - Способен управлять архитектурой программного обеспечения, в том числе интегрированного;

ПК-5 - Способен выполнять работы по созданию, модификации и сопровождению интеграционных решений.

Основными задачами дисциплины являются:

1. ознакомление с современными методами и технологиями конструирования программного обеспечения;
2. изучение способов конструирования программного обеспечения с использованием языка моделирования UML;
3. формирования умений и навыков выработки конструкторских решений;
4. формирование навыков работы в современных инструментальных средах поддержки процесса конструирования программных систем.

Показатель освоения компетенции отражают следующие индикаторы:

Необходимые знания:

- Знает понятие архитектуры программного обеспечения;
- Знает основные особенности архитектуры интегрированного программного обеспечения;
- Знает принципы определения требований к интеграционному решению;
- Знает основные методы, способы и средства интеграции программных продуктов.

• Необходимые умения:

- Определяет требования к программному продукту с точки зрения архитектуры;
- Применяет различные способы и методы моделирования архитектурного решения;
- Определяет требования к программному продукту с точки зрения архитектуры;
- Применяет различные способы и методы моделирования архитектурного решения;

Трудовые действия:

- Выявляет и согласовывает требования к программной системе с точки зрения архитектуры;
- Выбирает и моделирует архитектурное решение для реализации программной системы;
- Осуществляет поддержку процесса согласования требований к интеграционному решению;
- Конфигурирует интеграционное решение.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Архитектура программного обеспечения» относится к части, формируемой участниками образовательных отношений, основной профессиональной образовательной программы подготовки по направлению подготовки бакалавриата 09.03.04 «Программная инженерия».

Изучение данной дисциплины базируется на знаниях и умениях, полученных в рамках ранее изученных дисциплин «Информатика», «Информационные технологии в профессиональной деятельности», «Адаптированные информационные технологии» и компетенциях УК-2, УК-6, ОПК-2, ОПК-6, ОПК-7, ОПК-8.

Знания и умения, полученные при освоении дисциплины «Архитектура программного обеспечения», применяются при изучении последующих дисциплин «Базы данных», «Проектирование программного обеспечения» и при прохождении практики и государственной итоговой аттестации.

3. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 часа.

Таблица 1

| Виды занятий | Всего часов | Семестр третий | Семестр | Семестр | Семестр |
|---------------------------------------|--------------------|-----------------------|----------------|----------------|----------------|
| Общая трудоемкость | 144 | 144 | | | |
| Аудиторные занятия | 48 | 48 | | | |
| Лекции (Л) | 16 | 16 | | | |
| Практические занятия (ПЗ) | 32 | 32 | | | |
| Лабораторные работы (ЛР) | - | - | | | |
| Практическая подготовка | 32 | 32 | | | |
| Самостоятельная работа | 96 | 96 | | | |
| Курсовые, расчетно-графические работы | - | - | | | |
| Контрольная работа, домашнее задание | + | + | | | |
| Вид итогового контроля | экзамен | экзамен | | | |

4. Содержание дисциплины

4.1. Темы дисциплины и виды занятий

Таблица 2

| Наименование тем | Лекции, час. оч | Прак. занятия, час. оч | Лаборато рные занятия, час. оч | Занятия в интеракт ивной форме, час оч | Код компете нций |
|---|-----------------------|---------------------------------|--|--|------------------------|
| Тема 1. Введение в конструирование программного обеспечения. Общие концепции разработки архитектуры ПО. | 4 | 8 | - | 2 | ПК-3, ПК-5 |
| Тема 2. Методы «защитного конструирования». Рефакторинг программного обеспечения. | 4 | 8 | - | 4 | ПК-3, ПК-5 |
| Тема 3. Обобщенные шаблоны распределения обязанностей компонентов системы. Порождающие объектно-ориентированные шаблоны проектирования. | 4 | 8 | - | 4 | ПК-3, ПК-5 |
| Тема 4. Структурные шаблоны проектирования. Поведенческие шаблоны проектирования. Анти-шаблоны проектирования систем. | 4 | 8 | - | 2 | ПК-3, ПК-5 |
| ИТОГО: | 16 | 32 | - | 12 | |

4.2. Содержание тем дисциплины

Тема 1. Введение в конструирование программного обеспечения. Общие концепции разработки архитектуры ПО.

Место, цели и задачи конструирования в процессе разработки программного обеспечения. Взаимосвязь конструирования с другими дисциплинами программной инженерии: проектирование, тестирование, управления конфигурациями, качество программного обеспечения. Основные составляющие компоненты конструирования программного обеспечения: основы конструирования (минимизация сложности, ожидание изменений,

конструирование с возможностью проверки, стандарты в конструировании), управление конструированием (модели конструирования, планирование конструирования, изменения в конструировании), практические соображения (проектирование в конструировании, языки конструирования, кодирование, тестирование, повторное использование, качество, интеграция)

Тема 2. Методы «защитного конструирования». Рефакторинг программного обеспечения.

Понятия «Абстрактного типа данных». Методы создания качественных интерфейсов классов, принципы построения качественных абстрактных интерфейсов. Защита программного обеспечения от неправильных входных данных. Механизм «Утверждений» и принципы его использования. Способы обработки ошибок в программном обеспечении. Механизм исключительных ситуаций и способы его использования. Изоляция повреждений, вызванных ошибками. Отладка программного обеспечения. Принципы рефакторинга. Способы выявления проблемных участков кода. Методы рефакторинга: улучшения структуры методов в классах, организация данных, упрощение взаимодействия классов, упрощение условных выражений, упрощение вызовов методов, решение задач обобщения, методы «крупного» рефакторинга. Инструментальные средства проведения рефакторинга.

Тема 3. Обобщенные шаблоны распределения обязанностей компонентов системы. Порождающие объектно-ориентированные шаблоны проектирования.

Понятие шаблонов проектирования, задачи, решаемые при их помощи, история создания. Классификация шаблонов проектирования: принципы ООП, шаблоны проектирования, идиомы программирования. Преимущества применения шаблонов и возможные проблемы. Шаблоны GRASP: информационный эксперт, создатель, контроллер, низкая связанность, сильное сцепление, полиморфизм, искусственный объект, посредник, устойчивость к изменениям. Задачи порождающих шаблонов проектирования. Опасность явного создания объектов классов. Каталог порождающих шаблонов: Абстрактная фабрика (Abstract Factory, Factory), Одиночка (Singleton), Прототип (Prototype), Строитель (Builder), Фабричный метод

(Factory Method). Примеры применения порождающих шаблонов проектирования.

Тема 4. Структурные шаблоны проектирования. Поведенческие шаблоны проектирования. Анти-шаблоны проектирования систем.

Задачи структурных шаблонов проектирования. Каталог структурных шаблонов: Адаптер (Adapter), Заместитель (Proxy), Декоратор (Decorator), Компоновщик (Composite), Мост (Bridge), Приспособленец (Flyweight), Фасад (Facade). Примеры применения структурных шаблонов проектирования. Критерии выбора подходящего структурного шаблона. Задачи поведенческих шаблонов проектирования. Каталог поведенческих шаблонов: Наблюдатель (Observer), Команда (Command), Интерпретатор (Interpreter), Итератор (Iterator), Посетитель (Visitor), Посредник (Mediator), Состояние (State), Стратегия (Strategy), Хранитель (Memento), Цепочка обязанностей (Chain of Responsibility), Шаблонный метод (Template Method). Примеры применения поведенческих шаблонов проектирования.

Понятие анти-шаблона. Классификация анти-шаблонов: анти-шаблоны в управлении разработкой ПО, анти-шаблоны в проектировании ПО, антишаблоны в объектно-ориентированном программировании, антишаблоны в программировании, методологические анти-шаблоны, антишаблоны управления конфигурацией, организационные анти-шаблоны. Способы минимизации последствий.

5. Перечень учебно-методического обеспечения для самостоятельной работы по дисциплине

Методические указания для самостоятельной работы обучающихся по освоению дисциплины представлены в Приложении 2.

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Структура фонда оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине «Архитектура программного обеспечения» приведена в Приложении 1 к данной рабочей программе.

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Гагарина, Л. Г. Введение в архитектуру программного обеспечения: учебное пособие / Л. Г. Гагарина, А. Р. Федоров, П. А.

Федоров. — Москва: ФОРУМ: ИНФРА-М, 2020. — 320 с. — (Высшее образование). - ISBN 978-5-8199-0649-1. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1046281>.

2. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул; под ред. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2023. — 400 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-8199-0707-8. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1971872>

Дополнительная литература:

1. Вылегжанина, А.О. Информационно-технологическое и программное обеспечение управления проектом: учебное пособие / А.О. Вылегжанина. – М. – Берлин: Директ-Медиа, 2015. –429 с. – Режим доступа: http://biblioclub.ru/index.php?page=book_view&book_id=362892.
2. Антамошкин, О.А. Программная инженерия. Теория и практика: учебник / О.А. Антамошкин. – Красноярск: Сибирский федеральный университет, 2012. –247 с. – Режим доступа: http://biblioclub.ru/index.php?page=book_view&book_id=363975
3. Ехлаков, Ю.П. Управление программными проектами : учебник / Ю.П. Ехлаков ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). - Томск : Томский государственный университет систем управления и радиоэлектроники, 2015. - 217 с. – Режим доступа: <http://biblioclub.ru/index.php?page=book&id=480634>

8. Перечень ресурсов информационно-телекоммуникационной сети «интернет», необходимой для освоения дисциплины

- <https://openedu.ru/course/urfu/SYSTENG/> - «Открытое образование», Каталог курсов, MOOK: «Системная инженерия».
- <http://www.biblioclub.ru/> -университетская библиотека онлайн

9. Методические указания для обучающихся по освоению дисциплины

Методические указания для обучающихся по освоению дисциплины приведены в приложении 2 к данной рабочей программе.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Программные продукты: MS Windows 7 и выше или другая ОС, Open source для реализации ИС, Python, Adobe Acrobat Professional, WinZIP, MSOffice или Open source для работы с текстовыми документами.

Информационные справочные системы:

1. Ресурсы информационно-образовательной среды Университета.
2. Рабочая программа и методическое обеспечение по дисциплине «Архитектура программного обеспечения».

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Лекционные занятия:

- Аудитория, оснащенная презентационной техникой (проектор, SmartBoard или экран);
- Комплект электронных презентаций по темам лекций
- Рабочее место преподавателя, оснащенное компьютером с доступом в Интернет

Практические занятия:

- Компьютерная аудитория, оснащенная ПК с необходимым ПО (MS Windows 7 и выше или другая ОС, MSOffice или Open source для работы с текстовыми документами, антивирусное ПО, архиватор), а также проектором для интерактивного обучения;
- Рабочее место преподавателя, оснащенное компьютером с доступом в Интернет;
- Рабочие места обучающихся, оснащенные компьютерами с доступом в Интернет;
- Классная доска с комплектом маркеров.

**ИНСТИТУТ
ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ
КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО
ДИСЦИПЛИНЕ
«АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»**

Направление подготовки: 09.03.04 Программная инженерия

Направленность (профиль): Проектирование и разработка программного обеспечения

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев 2023

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

| № п/п | Индекс компетенции | Содержание компетенции (или ее части) | Раздел дисциплины, обеспечивающий формирование компетенции (или ее части) | В результате изучения раздела дисциплины, обеспечивающего формирование компетенции, обучающийся приобретает: | | |
|-------|--------------------|---|---|--|--|--|
| | | | | Необходимые знания | Необходимые умения | Трудовые действия |
| 1. | ПК-3 | способен управлять архитектурой программного обеспечения, в том числе интегрированного | Темы 1-4 | Знает понятие архитектуры программного обеспечения; Знает основные особенности архитектуры интегрированного программного обеспечения | Определяет требования к программному продукту с точки зрения архитектуры; Применяет различные способы и методы моделирования архитектурного решения; | Выявляет и согласовывает требования к программной системе с точки зрения архитектуры; Выбирает и моделирует архитектурное решение для реализации программной системы; |
| 2. | ПК-5 | Способен выполнять работы по созданию, модификации и сопровождению интеграционных решений | Темы 1-4 | Знает принципы определения требований к интеграционному решению; Знает основные методы, способы и средства интеграции программных продуктов | Применяет принципы определения требований к интеграционному решению; Использует основные методы, способы и средства интеграции программных продуктов; | Осуществляет поддержку процесса согласования требований к интеграционному решению; Конфигурирует интеграционное решение; |

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

| Код компетенции | Инструменты, оценивающие сформированность компетенции | Показатель оценивания компетенции | Критерии оценки |
|-----------------|---|---|--|
| ПК-3, ПК-5 | Контрольная работа | А) полностью сформирована (компетенция освоена на <u>высоком</u> уровне) – 5 баллов Б) частично сформирована: А) компетенция освоена на <u>продвинутом</u> уровне – 4 балла; В) компетенция освоена на <u>базовом</u> уровне – 3 балла; С) В) не сформирована (компетенция не | Проводится в форме письменной работы Время, отведенное на процедуру – 60 мин. Неявка – 0. Критерии оценки: 1. Соответствие ответа заявленной тематике (0-5 баллов). Максимальная сумма баллов - 5 баллов. |

| Код компетенции | Инструменты, оценивающие сформированность компетенции | Показатель оценивания компетенции | Критерии оценки |
|-----------------|---|-----------------------------------|---|
| | | сформирована) – 2 и менее баллов | Результаты оценочной процедуры представляются обучающимся в срок не позднее 1 недели после проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал. |

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

3.1. Примерная тематика задач, выносимых на контрольную работу

1. Понятие архитектуры программного обеспечения.
2. Концепция конструирования программного обеспечения.
3. Анализ существующих методов конструирования программного обеспечения.
4. Методы «защитного конструирования».
5. Рефакторинг программного обеспечения.
6. Сравнительный анализ видов шаблонов конструирования программного обеспечения.
7. Обобщенные шаблоны распределения обязанностей компонентов системы.
8. Порождающие объектно-ориентированные шаблоны проектирования.
9. Структурные шаблоны проектирования.
10. Поведенческие шаблоны проектирования.
11. Анти-шаблоны проектирования систем.
12. Понятие модификации.
13. Описание требований к программным продуктам.
14. Основные подходы к созданию интеграционных решений.
15. Методы создания качественных интерфейсов классов, принципы построения качественных абстрактных интерфейсов.

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Формой контроля знаний по дисциплине «Архитектура программного обеспечения» являются две текущие аттестации в форме тестов и итоговая аттестация в форме экзамена.

| Неделя текущего контроля | Вид оценочного средства | Код компетенций, оценивающий знания, умения, навыки | Содержание оценочного средства | Требования к выполнению | Срок сдачи (неделя семестра) | Критерии оценки по содержанию и качеству с указанием баллов |
|--------------------------|-------------------------|---|--------------------------------|--|---|---|
| 7-8 | тестирование | ПК-3, ПК-5 | 20 вопросов | Компьютерное тестирование; время, отведенное на процедуру - 30 минут | Результаты тестирования предоставляются в день проведения процедуры | Критерии оценки определяются процентным соотношением. Не явка - Удовлетворительно - от 51% правильных ответов. Хорошо - от 70%. Отлично – от 90%. Максимальная оценка – 5 баллов. |
| 15-16 | тестирование | ПК-3, ПК-5 | 20 вопросов | Компьютерное тестирование; время, отведенное на процедуру – 30 минут | Результаты тестирования предоставляются в день проведения процедуры | Критерии оценки определяются процентным соотношением. Не явка -0 Удовлетворительно - от 51% правильных ответов. Хорошо - от 70%. Отлично – от 90%. Максимальная оценка – 5 баллов. |
| В соответствии с КУГ | Экзамен | ПК-3, ПК-5 | 2 вопроса | Проводится в устной форме, путем ответа на вопросы. Время, отведенное на процедуру – 40 минут. | Результаты предоставляются в день проведения экзамена | Критерии оценки: «Отлично»: • знание основных понятий предмета; • умение использовать и применять полученные знания на практике; • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответ на вопросы билета. «Хорошо»: • знание основных понятий предмета; • умение использовать и применять полученные знания на практике; • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответы на вопросы билета |

| Неделя текущего контроля | Вид оценочного средства | Код компетенций, оценивающий знания, умения, навыки | Содержание оценочного средства | Требования к выполнению | Срок сдачи (неделя семестра) | Критерии оценки по содержанию и качеству с указанием баллов |
|--------------------------|-------------------------|---|--------------------------------|-------------------------|------------------------------|---|
| | | | | | | <ul style="list-style-type: none"> • неправильно решено практическое задание «Удовлетворительно»: • демонстрирует частичные знания по темам дисциплин; • незнание неумение использовать и применять полученные знания на практике; • не работал на практических занятиях; • «Неудовлетворительно»: • демонстрирует частичные знания по темам дисциплин; • незнание основных понятий предмета; • неумение использовать и применять полученные знания на практике; • не работал на практических занятиях; • не отвечает на вопросы. |

4.1. Примерная тематика тестовых заданий

1. При конструировании программного обеспечения процесс решения задачи составляет
 - а) 90 — 95%;
 - б) 50%;
 - в) 5 — 10%;
 - г) 10% - 50%.

2. При конструировании программного обеспечения на этапе разработки или выбора алгоритма решения реализуется следующее:
 - а) архитектурная обработка программы;
 - б) выбор языка программирования;
 - в) совершенствование программы;
 - г) проектирование программы.

3. Как называется процесс разбиения одной сложной задачи на несколько простых подзадач?
 - а) абстракция;
 - б) декомпозиция;
 - в) реинжиниринг;

г) рефакторинг.

4. Архитектура программной системы

- а) декомпозиция решения для выделенного спектра задач домена на подсистемы или иерархию подсистем;
- б) определение системы в терминах вычислительных составляющих (подсистем) и интерфейсов между ними, которое отражает правила декомпозиции проблемы на составляющие;
- в) соответствующие вариации состава выделенных компонент.

5. Конструирование программного обеспечения предназначено для (возможны несколько вариантов ответа)

- а) создания рабочей программы системы посредством кодирования, верификации, тестирования
- б) разработки программы на языке программирования
- в) разработки алгоритма
- г) разработки модели системы

6. В основах конструирования заложены следующие операции (возможны несколько вариантов ответа)

- а) конструирование с возможностью проверки
- б) минимизация сложности
- в) сокращение времени выполнения программ
- г) ожидание изменений

7. Основными методами, направленными на достижение требуемого результата конструирования, являются (возможны несколько вариантов ответа):

- а) обзор, оценка кода (code review)
- б) модульное тестирование (unit-testing)
- в) структурирование кода с применением автоматизированных средств тестирования (automated testing)
- г) ограниченное применение сложных для понимания языковых структур

8. Стандарты, которые применяются при конструировании, включают (возможны несколько вариантов ответа):

- а) коммуникационные методы (стандарты форматов документов и содержания)
- б) языки программирования и соответствующие стили кодирования
- в) платформы программных средств
- г) инструменты для разработки кода программ
- д) стандарты языков программирования

9. Основные техники обеспечения качества, используемые в процессе конструирования, включают (возможны несколько вариантов ответа):

- а) модульное (unit) и интеграционное (integration) тестирование
- б) разработка с первичностью тестов (test-first development - тесты пишутся до конструирования кода)
- в) пошаговое кодирование (деятельность по конструированию кода разбивается на мелкие шаги, только после тестирования результатов которых производится переход к следующему шагу кодирования; известен также как итеративное кодирование с тестированием)
- г) использование процедур утверждений (assertion)
- д) отладка (в привычном понимании - debugging)
- е) технические обзоры и оценки (review)
- ё) статический анализ
- ж) динамический анализ

10. Технология конструирования программного обеспечения (ТКПО) – это

- а) система инженерных принципов для создания экономичного ПО,
- б) система принципов для использования языков программирования
- в) система разработки алгоритмов, моделей и программ
- г) использование case технологий

11. Методы (ТКПО) обеспечивают решение следующих задач (возможны несколько вариантов ответа):

- а) планирование и оценка проекта;
- б) анализ системных и программных требований;
- в) проектирование алгоритмов, структур данных и программных структур;
- г) кодирование;
- д) тестирование;
- е) сопровождение.

12. Системный анализ – это

- а) определение роли каждого элемента в компьютерной системе, взаимодействие элементов друг с другом
- б) разработка программы в соответствии с требованиями
- в) разработка программы на системном языке
- г) разработка программы на языке операционной системы

13. Стратегии конструирования ПО (возможны несколько вариантов ответа):

- а) однократный проход (водопадная стратегия)
- б) инкрементная стратегия
- в) эволюционная стратегия.
- г) структурный подход

13. Языки конструирования (возможны несколько вариантов ответа):

- а) Java
- б) C#
- в) C++

г) Si ++

14. Агрегация — это

- а) отношения, утверждает наличие связи между понятиями, не уточняя зависимости их содержания и объемов;
- б) возможность для некоторого класса находиться одновременно в связи с одним элементом из определенного множества классов;
- в) объединение нескольких понятий в новое понятие, существенные признаки нового понятия при этом могут быть либо суммой компонент или существенно новыми (отношение «доля — целое»).

15. Ассоциация — это

- а) возможность для некоторого класса находиться одновременно в связи с одним элементом из определенного множества классов;
- б) объединение нескольких понятий в новое понятие, существенные признаки нового понятия о этом могут быть либо суммой компонент или существенно новыми (отношение «доля — целое»);
- в) самое общее отношение, утверждает наличие связи между понятиями, не уточняя зависимости их содержания и объемов.

16. Валидация — это

- а) обеспечение соответствия разработки требованиям ее заказчиков.
- б) проверка правильности трансформации проекта в код реализации;
- в) выявление всех ошибок.

17. Верификация — это

- а) обеспечение соответствия разработки требованиям ее заказчиков;
- б) проверка правильности трансформации проекта в программу;
- в) действия на каждой стадии жизненного цикла с проверки и подтверждения соответствия стандартам.

18. Термин «проект» в инженерии программного обеспечения используется для обозначения

- а) процесса разработки ПО
- б) архитектуры ПО
- в) команды разработчиков
- г) результата проектирования

19. Необходимость внедрения командного метода разработки программ обосновывается

- а) уменьшением времени разработки программного продукта
- б) повышением эффективности работ
- в) уменьшением стоимости программного продукта
- г) повышением качества ПО за счет внедрения CASE-технологий

20. Первичными целями программного продукта является его

- а) безопасность
- б) корректность
- в) надежность
- г) соответствие требованиям заказчика

4.2. Типовые вопросы, выносимые на экзамен

1. Понятия и принципы методологии конструирования
2. Качество программных систем.
3. Среды разработки.
4. Основные этапы и стадии проектирования.
5. Стратегии проектирования.
6. Спецификации.
7. Диаграммы Варнье-Орра
8. Функциональные схемы.
9. Блок-схемы.
10. Синтаксические диаграммы.
11. Структурные преобразования. Простые преобразования.
12. Структурные преобразования. Дублирование элементов.
13. Структурные преобразования. Введение переменной состояния.
14. Архитектура программной системы.
15. Модульность
16. Проектирование модуля.
17. Структурное программирование.
18. Проверка правильности программ.
19. Тестирование, доказательство, контроль, испытание и др.
20. Базовые правила тестирования.
21. Отладка.
22. UML-проектирование.
23. Стандартизация.

Итоговое начисление баллов по дисциплине осуществляется в соответствии с разработанной и внедренной балльно-рейтинговой системой контроля и оценивания уровня знаний и внеучебной созидательной активности обучающихся.

**ИНСТИТУТ
ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ
И ТЕХНОЛОГИЙ**

КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО
ОСВОЕНИЮ ДИСЦИПЛИНЫ
«АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»**

Направление подготовки: 09.03.04 Программная инженерия

Направленность (профиль): Проектирование и разработка программного обеспечения

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев 2023

1. Общие положения

Цель изучения дисциплины является

получение обучающимися знаний и приобретение практических навыков в области конструирования программного обеспечения, а также формирование у студента навыков сбора, обработки и представления исходных данных для принятия проектных решений; разработки концептуальных, информационно-логических и функциональных моделей программного обеспечения; объектно-ориентированного анализа и проектирования.

Основными **задачами** дисциплины являются

1. ознакомление с современными методами и технологиями конструирования программного обеспечения;
2. изучение способов конструирования программного обеспечения с использованием языка моделирования UML;
3. формирования умений и навыков выработки конструкторских решений;
4. формирование навыков работы в современных инструментальных средах поддержки процесса конструирования программных систем.

2. Указания по проведению практических занятий

Практическое занятие 1. Введение в конструирование программного обеспечения (ПО)

Вид практического занятия: изучение основ конструирования с использованием персонального компьютера (ПК).

Образовательные технологии: программное и компьютерное обучение.

Цель занятия: изучить историю развития архитектуры ПО.

Вопросы для обсуждения:

1. Развитие архитектуры ПО
2. Основы конструирования ПО

Продолжительность занятия: 4 часа.

Практическое занятие 2. Общие концепции конструирования.

Вид практического занятия: изучение основных концепций с использованием персонального компьютера (ПК).

Образовательные технологии: программное и компьютерное обучение.

Цель занятия: изучение основ конструирования программного обеспечения.

Вопросы для обсуждения:

1. Понятия архитектуры ПО
2. Понятия конструирования ПО
3. Основные подходы к конструированию ПО

Продолжительность занятия: 4

Практическое занятие 3. Фазы тестирования

Вид практического занятия: проведение тестового анализа с использованием персонального компьютера (ПК).

Образовательные технологии: программное и компьютерное обучение.

Цель занятия: изучить основные фазы: планирование тестирования; подготовка тестового примера данных, среды; выполнение теста; регистрация ошибок; отслеживание и закрытие.

Вопросы для обсуждения:

1. Понятие тестового анализа.
2. Исследование основных фаз тестирования.

Продолжительность занятия: 4 часа.

Практическое занятие 4. Исследование модульного тестирования

Вид практического занятия: исследование методики тестирования отдельных модулей или компонентов программы с использованием персонального компьютера (ПК).

Образовательные технологии: программное и компьютерное обучение.

Цель занятия: изучение этапа разработки кода ПО и проведение анализа работы каждой единицы программного кода.

Цель этого тестирования состоит в том, чтобы убедиться, что каждая единица программного кода работает должным образом. Обычно оно проводится на этапе разработки кода приложения. Модульные тест-кейсы изолируют фрагмент кода и проверяют его правильность.

Вопросы для обсуждения:

1. Модульные тест-кейсы.
2. Методика модульного тестирования.

Продолжительность занятия: 4 часа.

3. Указания по проведению лабораторного практикума

Не предусмотрен учебным планом.

4. Указания по проведению самостоятельной работы студентов

Цель самостоятельной работы: подготовка к практическим занятиям, обзорам по предложенным темам, подготовка к промежуточной аттестации, выполнение и защиту контрольных работ, курсового проекта, подготовка к зачётам, а также подготовка бакалавров к самостоятельному научному творчеству.

Задачи самостоятельной работы:

1. Расширить знания в области программного обеспечения.
2. Систематизировать знания в области конструирования программного обеспечения.
3. Овладеть навыками решения различных задач в области проведения тестирования программного обеспечения.

| № пп | Наименование блока (раздела) дисциплины | Виды СР |
|------|---|--|
| 1. | Тема 1. Введение в конструирование программного обеспечения. Общие концепции разработки архитектуры ПО. | Подготовка рефератов и докладов, углубление знаний по пройденной теме. |
| 2. | Тема 2. Методы «защитного конструирования». Рефакторинг программного обеспечения. | Подготовка рефератов и докладов, углубление знаний по пройденной теме. |
| 3. | Тема 3. Обобщенные шаблоны распределения обязанностей компонентов системы. Порождающие объектно-ориентированные шаблоны проектирования. | Подготовка рефератов и докладов, углубление знаний по пройденной теме. |

| № пп | Наименование блока (раздела) дисциплины | Виды СР |
|---------|---|--|
| 4. | Тема 4. Структурные шаблоны проектирования. Поведенческие шаблоны проектирования. Анти-шаблоны проектирования систем. | Подготовка рефератов и докладов, углубление знаний по пройденной теме. |

5. Указания по проведению контрольных работ

Основной целью контрольной работы является закрепление основных положений дисциплины. Контрольная работа может включать в себя рассмотрение теоретических вопросов дисциплины, а также их практическое приложение.

5. 1. Требования к структуре

Структура контрольной работы должна способствовать раскрытию темы: иметь титульный лист, содержание, введение, основную часть, заключение, список литературы.

5. 2. Требования к содержанию (основной части)

1. Во введении обосновывается актуальность темы, определяется цель работы, задачи и методы исследования (для теоретических вопросов) и методы решения задачи (для практических заданий).

2. При определении целей и задач необходимо правильно их формулировать. Так, в качестве цели не следует употреблять глагол «сделать». Правильно будет использовать глаголы: «раскрыть», «определить», «установить», «показать», «выявить» и т. д.

3. Основная часть работы включает 2-4 вопроса, каждый из которых посвящается решению задач, сформулированных во введении, и заканчивается констатацией итогов (для теоретических вопросов) и решение задачи по теме с описанием основных этапов и листингом.

4. Приветствуется иллюстрация содержания работы таблицами, графическим материалом (рисунками, схемами, скрин-шотами и т. п.).

5. Необходимо давать ссылки на используемую Вами литературу.

6. Заключение должно содержать сделанные автором работы выводы, итоги исследования и результаты решения задачи.

7. Вслед за заключением идет список литературы, который должен быть составлен в соответствии с установленными требованиями. Если в работе имеются приложения, они оформляются на отдельных листах, и должны быть соответственно пронумерованы.

5. 3. Требования к оформлению

Объём контрольной работы – 10-15 печатных страниц формата А4, напечатанного с одной стороны текста. Титульный лист – принятый в Университете для оформления подобных видов работ. Оформляется в MS Word или другом текстовом редакторе по следующим правилам:

1. Шрифт TimesNewRoman, кегль 12-14, интервал между строками 1,5 строки, поля: верхнее и нижнее по 2 см, левое – 3 см, правое – 1 см. Отступ первой строки – 1,25см.
2. Все заголовки оформляются стилями заголовков. При этом необходимо изменить шрифт на TimesNewRoman, кегль до 16 (в зависимости от уровня заголовка), цвет черный.
3. Содержание (оглавление) оформляется по всем требованиям текстового процессора
4. Обязательное наличие списка используемых источников. При этом в тексте указать в квадратных скобках номер используемого источника (литературы)

6. Перечень основной и дополнительной учебной литературы

Основная литература:

3. Гагарина, Л. Г. Введение в архитектуру программного обеспечения: учебное пособие / Л. Г. Гагарина, А. Р. Федоров, П. А. Федоров. — Москва: ФОРУМ: ИНФРА-М, 2020. — 320 с. — (Высшее образование). - ISBN 978-5-8199-0649-1. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1046281>.
4. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул; под ред. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2023. — 400 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-8199-0707-8. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1971872>

Дополнительная литература:

4. Вылегжанина, А.О. Информационно-технологическое и программное обеспечение управления проектом: учебное пособие / А.О. Вылегжанина.

- М. – Берлин: Директ-Медиа, 2015. –429 с. – Режим доступа: http://biblioclub.ru/index.php?page=book_view&book_id=362892.
5. Антамошкин, О.А. Программная инженерия. Теория и практика: учебник / О.А. Антамошкин. – Красноярск: Сибирский федеральный университет, 2012. –247 с. – Режим доступа: http://biblioclub.ru/index.php?page=book_view&book_id=363975
6. Ехлаков, Ю.П. Управление программными проектами : учебник / Ю.П. Ехлаков ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). - Томск : Томский государственный университет систем управления и радиоэлектроники, 2015. - 217 с. – Режим доступа: <http://biblioclub.ru/index.php?page=book&id=480634>

6. Перечень ресурсов информационно-телекоммуникационной сети «интернет»

- <https://openedu.ru/course/urfu/SYSTENG/> - «Открытое образование», Каталог курсов, MOOK: «Системная инженерия».
- <http://www.biblioclub.ru/> -университетская библиотека онлайн

7. Перечень информационных технологий

Программные продукты: MS Windows 7 и выше или другая ОС, Open source для реализации ИС, Python, Adobe Acrobat Professional, WinZIP, MSOffice или Open source для работы с текстовыми документами.

Информационные справочные системы:

1. Ресурсы информационно-образовательной среды Университета.
2. Рабочая программа и методическое обеспечение по дисциплине «Архитектура программного обеспечения».