

**Комплект оценочных материалов по дисциплине
«Моделирование и анализ программного обеспечения»**

Задания закрытого типа

Задания закрытого типа на выбор правильного ответа

1. Выберите один правильный ответ.

Какова основная цель моделирования в процессе разработки ПО?

- А) Создание кода без предварительного проектирования
- Б) Формирование концептуальной модели системы
- В) Замена этапа тестирования
- Г) Исключительно визуализация данных

Правильный ответ: Б

Компетенции (индикаторы): ПК-2

2. Выберите один правильный ответ.

Какой вид моделирования используется для представления структуры данных системы?

- А) Функциональное моделирование
- Б) Информационное моделирование
- В) Моделирование взаимодействия
- Г) Процессное моделирование

Правильный ответ: Б

Компетенции (индикаторы): ПК-2

3. Выберите один правильный ответ.

Какой из методов моделирования описывает поведение системы с точки зрения взаимодействия пользователей и системы?

- А) Статическое моделирование
- Б) Динамическое моделирование
- В) Структурное моделирование
- Г) Функциональное моделирование

Правильный ответ: Б

Компетенции (индикаторы): ПК-2

4. Выберите один правильный ответ.

В рамках моделирования программного обеспечения, что такое "архитектура системы"?

- А) Данный документ
- Б) Схема баз данных
- В) Высокоуровневое описание структуры системы
- Г) Этап тестирования

Правильный ответ: В

Компетенции (индикаторы): ПК-2

Задания закрытого типа на установление соответствия

1. Установите правильное соответствие. Каждому элементу левого столбца соответствует только один элемент правого столбца.

- | Языки моделирования | Области применения |
|---------------------|---------------------------------------|
| 1) UML | А) Описание бизнес-процессов |
| 2) BPMN | Б) Функциональное моделирование |
| 3) IDEF0 | В) Визуализация структуры базы данных |
| 4) ERD | Г) Описание архитектуры ПО |

Правильный ответ:

1	2	3	4
Г	А	Б	В

Компетенции (индикаторы): ПК-2

2. Установите правильное соответствие. Каждому элементу левого столбца соответствует только один элемент правого столбца.

- | Ключевые этапы анализа ПО | Характеристики этапов |
|---------------------------|---|
| 1) Сбор требований | А) Проверка полноты и корректности требований |
| 2) Анализ требований | Б) Определение целей системы и ограничений |
| 3) Проектирование | В) Разработка архитектуры и структуры ПО |
| 4) Валидация | Г) Выявление функциональных и нефункциональных требований |

Правильный ответ:

1	2	3	4
Г	Б	В	А

Компетенции (индикаторы): ПК-2

3. Установите правильное соответствие. Каждому элементу левого столбца соответствует только один элемент правого столбца.

- | Основные роли в процессе моделирования ПО | Функции |
|---|--|
| 1) Аналитик | А) Разрабатывает архитектуру системы и её компоненты |
| 2) Архитектор ПО | Б) Анализирует требования, создаёт модели системы |
| 3) Разработчик | В) Пишет программный код и интегрирует компоненты |

4) Тестировщик

Г) Проверяет корректность работы системы

Правильный ответ:

1	2	3	4
Б	А	В	Г

Компетенции (индикаторы): ПК-2

Задания закрытого типа на установление правильной последовательности

1. Расположите этапы анализа программного обеспечения в правильном порядке:

- А) Оптимизация кода и архитектуры
- Б) Повторная проверка системы после оптимизации
- В) Анализ исходного кода
- Г) Поиск проблемных мест и узких мест в производительности

Правильный ответ: В, Г, А, Б

Компетенции (индикаторы): ПК-2

2. Расположите этапы процесса моделирования программного обеспечения в правильном порядке:

- А) Валидация моделей
- Б) Анализ требований
- В) Использование моделей для проектирования
- Г) Создание моделей

Правильный ответ: Б, Г, А, В

Компетенции (индикаторы): ПК-2

3. Установите правильную последовательность создания модели в Enterprise Architect:

- А) Настройка связей между элементами
- Б) Экспорт диаграммы или отчета
- В) Добавление элементов модели
- Г) Создание нового проекта

Правильный ответ: Г, В, А, Б

Компетенции (индикаторы): ПК-2

Задания открытого типа

Задания открытого типа на дополнение

1. Напишите пропущенное слово.

_____ – это процесс создания абстрактного представления системы, которое помогает понять, анализировать или предсказать её поведение.

Правильный ответ: Моделирование.

Компетенции (индикаторы): ПК-2

2. Напишите пропущенное слово.

_____ – это процесс создания предварительной версии системы или её компонентов для визуализации идей и получения обратной связи от пользователей

Правильный ответ: Прототипирование

Компетенции (индикаторы): ПК-2

3. Напишите пропущенное словосочетание.

_____ – это процесс устранения избыточности при моделировании данных

Правильный ответ: Нормализация.

Компетенции (индикаторы): ПК-2

4. Напишите пропущенное словосочетание.

_____ – это проверка того, соответствует ли программное обеспечение установленным требованиям и спецификациям.

Правильный ответ: Тестирование на соответствие.

Компетенции: ПК-1, ПК-2

5. Напишите пропущенное слово.

Неполные требования могут привести к _____ в процессе разработки.

Правильный ответ: проблемам.

Компетенции (индикаторы): ПК-2

6. Напишите пропущенное словосочетание.

_____ – это вид анализа, который позволяет определять и минимизировать риски проекта?

Правильный ответ: Анализ рисков.

Компетенции (индикаторы): ПК-2

7. Напишите пропущенное словосочетание.

_____ моделирование – это процесс определения структуры программной системы, включая основные компоненты, их взаимосвязи, взаимодействие с внешними сервисами и пользователями, а также принципы проектирования.

Правильный ответ: Архитектурное.

Компетенции (индикаторы): ПК-2

8. Напишите пропущенное словосочетание

Диаграмма _____ в UML отображает динамическое поведение системы, показывая взаимодействие между объектами в определенной последовательности во времени. Ее используют для моделирования сценариев использования, а также для анализа и проектирования систем, где важна последовательность выполнения операций.

Правильный ответ: последовательностей

Компетенции (индикаторы): ПК-2

Задания открытого типа с кратким свободным ответом

1. Как называется методология функционального моделирования процессов?

Правильный ответ: IDEF0 / SADT / Structured Analysis and Design Technique.

Компетенции (индикаторы): ПК-2.

2. Как называется графическое представление потоков данных в системе?

Правильный ответ: DFD / Data Flow Diagram / Диаграмма потоков данных / Диаграммы потоков данных.

Компетенции (индикаторы): ПК-2

3. Напишите пропущенное словосочетание.

_____ – это процесс создания абстрактных моделей программной системы, отображающих различные её аспекты

Правильный ответ: Программное моделирование / Моделирование ПО / Моделирование программного обеспечения .

Компетенции (индикаторы): ПК-2

4. Какой термин обозначает анализ требований в контексте пользовательского опыта?

Правильный ответ: UX-анализ / User Experience analysis / Анализ UX

Компетенции (индикаторы): ПК-2

5. Какой метод используется для моделирования взаимодействий пользователя с системой?

Правильный ответ: Use Case / Метод Use Case / Диаграмма Use Case / Диаграммы Use Case / Диаграмма вариантов использования / Варианты использования.

Компетенции (индикаторы): ПК-2

Задания открытого типа с развернутым ответом

1. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите ответ, используя четкие компактные формулировки.

Вопрос: В чем отличие статического и динамического моделирования программного обеспечения? Как эти подходы дополняют друг друга?

Время выполнения 35 минут

Ожидаемый результат:

Определение понятий.

Статическое моделирование описывает структурные аспекты программного обеспечения, включая классы, объекты, их связи и архитектуру системы. Оно применяется на этапе проектирования для определения структуры ПО.

Динамическое моделирование фокусируется на поведении системы во времени, описывая последовательность взаимодействий между объектами, изменения состояний и выполнение процессов.

Основные различия

Статическое моделирование показывает, из чего состоит система, описывая ее элементы и их связи. Оно используется при анализе и проектировании архитектуры, а также фиксирует структуру системы.

Динамическое моделирование показывает, как система функционирует во времени. Оно помогает анализировать процессы, выявлять исключительные ситуации и тестировать поведение системы в различных сценариях.

Для статического моделирования применяются диаграммы классов, компонентов, объектов, а для динамического – диаграммы последовательностей, состояний, активности.

Пример статического моделирования – ER-диаграмма базы данных, где определяются связи между сущностями. Пример динамического моделирования – процесс авторизации пользователя, представленный в виде диаграммы последовательностей.

Как статическое и динамическое моделирование дополняют друг друга?

1) Формирование полной модели системы.

Статическое моделирование описывает основные структуры системы, а динамическое моделирование определяет, как эти структуры взаимодействуют в процессе работы.

2) Выявление логических ошибок

Если в статической модели указаны методы, которые не используются в динамических сценариях, это может указывать на избыточность функционала. Если же динамическая модель требует объект, который не определен в статической модели, это означает, что проектирование выполнено с ошибками.

3) Оптимизация проектирования и тестирования

Статическая модель определяет основные структуры, на основе которых создаются тестовые сценарии. Динамическая модель помогает тестировщикам

понимать последовательность операций и выявлять потенциальные ошибки во взаимодействиях.

4) Оптимизация архитектуры

Если статическая структура оказывается слишком сложной или неэффективной для бизнес-процессов, динамическое моделирование помогает выявить проблемы и предложить улучшения. Например, если процесс авторизации требует слишком много взаимодействий между компонентами, можно упростить архитектуру, объединив несколько шагов.

Пример использования.

Моделирование программного обеспечения для системы бронирования отелей.

Статическая модель включает классы Hotel, Room, Guest, Reservation и их взаимосвязи.

Динамическая модель представляет сценарий бронирования:

Гость отправляет запрос на бронирование.

Система проверяет доступность номера.

Если номер свободен, создается запись бронирования.

Гость получает подтверждение.

Как в этом примере модели дополняют друг друга?

Без статической модели непонятно, какие объекты участвуют в системе. Без динамической модели невозможно определить, как они взаимодействуют в процессе бронирования.

Выводы.

- статическое и динамическое моделирование – это два взаимодополняющих подхода, которые обеспечивают полное понимание структуры и поведения программного обеспечения;

- статическое моделирование помогает построить четкую архитектуру системы;

- динамическое моделирование показывает, как система будет работать в реальных сценариях;

- вместе они помогают выявлять ошибки, оптимизировать проектирование и тестирование ПО.

Использование обоих подходов позволяет минимизировать ошибки, улучшить документацию и сделать систему более надежной и предсказуемой.

Критерии оценивания:

- правильное определение обоих типов моделирования;

- корректное объяснение их различий и взаимодополняемости;

- наличие примера, иллюстрирующего оба подхода в одной системе.

Ответ должен показывать связь между статическим и динамическим моделированием, а не просто перечислять их определения. Если в ответе есть только одно из понятий или отсутствует пример – оценка должна снижаться.

Студент может привести примеры из разных областей, но важно, чтобы ответ включал пример применения подходов и их взаимодействие. Ответ

может варьироваться по деталям, разные формулировки допустимы, но общий смысл должен быть верным.

Компетенции (индикаторы): ПК-2

2. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите ответ, используя четкие компактные формулировки.

Вопрос: Какие преимущества дает использование формальных методов в моделировании и анализе программного обеспечения? Проиллюстрировать использование формальных методов в разработке программного обеспечения для беспилотных летательных аппаратов или на любом другом практическом примере.

Время выполнения: 35 минут.

Ожидаемый результат:

1. Что такое формальные методы?

Формальные методы – это математически обоснованные техники и инструменты, используемые для спецификации, проектирования, анализа и верификации программного обеспечения. Они помогают гарантировать корректность и надежность ПО за счет строгого описания поведения системы.

Основные компоненты формальных методов:

- формальные спецификации – математические описания поведения системы (например, Z-нотация, VDM, Alloy);
- верификация моделей – проверка соответствия модели заданным требованиям (например, Model Checking, Hoare Logic);
- доказательство корректности – математическое подтверждение правильности алгоритмов и программ (например, метод Флойда-Хоара).

Применение формальных методов в моделировании и анализе ПО.

1) Спецификация требований

Формальные методы позволяют однозначно и недвусмысленно описывать требования к программной системе, избегая разночтений. Например, в банковских системах формальная спецификация помогает предотвратить ошибки при расчетах транзакций.

2) Анализ архитектуры и поведения системы

С помощью формального моделирования можно проверить, как система будет вести себя в различных сценариях, не дожидаясь фактической разработки кода. Например, Model Checking используется в разработке микропроцессоров для проверки корректности работы регистров.

3) Верификация алгоритмов и безопасности

Формальные методы позволяют доказать, что алгоритм не содержит ошибок. Это особенно важно для критически важных систем, таких как авионика или медицинские устройства. Например, метод LTL (Linear Temporal Logic) применяется для проверки безопасности авиационного ПО.

4) Обнаружение потенциальных дефектов на ранних этапах

Формальные методы помогают находить ошибки еще на стадии проектирования, что снижает затраты на их исправление. В отличие от

тестирования, которое выявляет лишь конкретные случаи сбоев, формальные методы позволяют доказать корректность всей системы.

5) Повышение надежности и отказоустойчивости ПО

Использование формальных методов позволяет гарантировать отсутствие определенных типов ошибок (например, гонок состояний в многопоточных системах). Например, при разработке ядер операционных систем (Microsoft Hyper-V, SEL4) используются формальные методы для доказательства безопасности кода.

Пример: Использование формальных методов в разработке программного обеспечения для беспилотных летательных аппаратов (БПЛА)

Компания разрабатывает программное обеспечение для беспилотного дрона, который должен автономно прокладывать маршрут, избегая препятствия и соблюдая воздушные правила. Ошибки в коде могут привести к столкновениям с другими объектами, выходу за границы безопасной зоны или даже падению дрона.

Как применяются формальные методы для данного примера?

1) Формальная спецификация позволяет четко описать, как дрон должен себя вести в любой ситуации:

- описываются математически обоснованные правила движения дрона с учетом высоты, скорости, ограничений воздушного пространства;

- используется логика темпоральных действий (Temporal Logic) для описания последовательности событий (например, «если препятствие обнаружено, дрон должен изменить курс в течение 1 секунды»).

2) Model Checking (проверка модели) помогает проверить, что алгоритмы уклонения от препятствий работают безошибочно:

- проверяется, что при любых входных данных (например, внезапное появление препятствия) дрон всегда выполняет корректный маневр уклонения;

- исключается вероятность бесконечных циклов, при которых дрон застрянет в одном месте.

3) Доказательство корректности навигационного алгоритма - исключает ситуации, в которых дрон не сможет принять решение или выйдет за безопасные границы:

- формально доказывается, что дрон не выйдет за допустимые границы маршрута (например, не нарушит запретные зоны полета);

- проверяется, что система управления питанием не допустит отключения дрона в воздухе (например, что система вовремя отправит его на зарядку).

Результат:

- исключены ошибки, которые могли бы привести к аварийным ситуациям;

- проверено поведение дрона в миллионах возможных сценариев, чего невозможно достичь классическими тестами;

– подтверждена корректная интеграция навигационной системы с датчиками и автопилотом.

Использование формальных методов в разработке ПО для БПЛА гарантирует безопасность и предсказуемость полета, исключая аварийные ситуации и сбои в навигации. Без формальных методов реальное тестирование потребовало бы тысячи реальных полетов, а вероятность критических ошибок была бы выше.

Критерии оценивания:

– дано корректное определение формальных методов в моделировании ПО;

– описано применение формальных методов в моделировании и анализе ПО ;

– приведен практический пример (программное обеспечение дрона);

– описаны конкретные преимущества формального моделирования;

– ответ логично структурирован и содержит четкие выводы.

Рекомендации проверяющему:

Ответ может варьироваться по деталям, разные формулировки допустимы, но общий смысл должен быть верным.

Студент должен объяснить, что такое формальные методы, как они применяются в моделировании и анализе программного обеспечения и какие преимущества они дают по сравнению с неформальными подходами.

Пример из реальной практики также обязателен. Приводимый студентом пример может любым, относиться к любой предметной области, важно, чтобы пример демонстрировал практическую ценность формальных методов. Если пример не содержит объяснений того, как формальные методы обеспечивают, например, безопасность БПЛА, – оценка снижается.

Компетенции (индикаторы): ПК-2

Экспертное заключение

Представленный комплект оценочных материалов по дисциплине «Моделирование и анализ программного обеспечения» соответствует требованиям ФГОС ВО.

Предлагаемые оценочные материалы адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 09.04.04 Программная инженерия.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

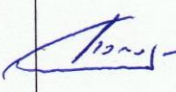
Разработанные и представленные для экспертизы оценочные материалы рекомендуются к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической комиссии
института компьютерных систем
и информационных технологий



Ветрова Н.Н.

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)
1.	Дополнен комплект оценочных материалов	протокол заседания кафедры компьютерных систем и сетей № <u>8</u> от <u>10.03.2025</u>	 С.В. Попов