

Комплект оценочных материалов по междисциплинарному курсу
МДК 01.02 Поддержка и тестирование программных модулей
09.02.07 Информационные системы и программирование

Задания закрытого типа

Задания закрытого типа на выбор правильного ответа

Выберите один правильный ответ

1. Каждая команда программы должна быть выполнена хотя бы один раз в соответствии с методом тестирования ...

- А) покрытие операторов;
- Б) покрытие решений;
- В) покрытие условий;
- Г) покрытие решений/условий.

Правильный ответ: А

Компетенции: ОК 1, ОК 9, ПК 1.4.

2. Каждое направление перехода в программе должно быть реализовано, по крайней мере, один раз в соответствии с методом тестирования ...

- А) покрытие операторов;
- Б) покрытие решений;
- В) покрытие условий;
- Г) покрытие решений/условий.

Правильный ответ: В

Компетенции: ОК 2, ОК 9, ПК 1.4.

3. Все возможные результаты каждого условия в решении должны быть выполнены, по крайней мере, один раз в соответствии с методом тестирования ...

- А) покрытие операторов;
- Б) покрытие решений;
- В) покрытие условий;
- Г) покрытие решений/условий.

Правильный ответ: Г

Компетенции: ОК 2, ОК 9, ПК 1.4.

4. Модульное тестирование предполагает ...

А) тестирование минимально возможного для тестирования компонента, например отдельного класса или функции

Б) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы

В) тестирование интегрированной системы на ее соответствие исходным требованиям

Г) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика

Правильный ответ: А

Компетенции: ОК 1, ОК 9, ПК 1.4.

5. Интеграционное тестирование предполагает ...

А) тестирование минимально возможного для тестирования компонента, например отдельного класса или функции

Б) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы

В) тестирование интегрированной системы на ее соответствие исходным требованиям

Г) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

Правильный ответ: Б

Компетенции: ОК 2, ОК 9, ПК 1.4.

6. Системное тестирование предполагает ...

А) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы

Б) тестирование интегрированной системы на ее соответствие исходным требованиям

В) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика

Г) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

Правильный ответ: Б

Компетенции: ОК 1, ОК 9, ПК 1.4.

7. Альфа-тестирование предполагает ...

А) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы

Б) тестирование интегрированной системы на ее соответствие исходным требованиям

В) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика

Г) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

Правильный ответ: В

Компетенции: ОК 1, ОК 2, ОК 9, ПК 1.4.

8. Бета-тестирование предполагает ...

А) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы

Б) тестирование интегрированной системы на ее соответствие исходным требованиям

В) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика

Г) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

Правильный ответ: Г

Компетенции: ОК 2, ОК 9, ПК 1.4.

9. Какой из следующих документов описывает функциональные требования к программному обеспечению?

А) Техническое задание

Б) План тестирования

В) Отчет о тестировании

Г) Пользовательская документация

Правильный ответ: А

Компетенции: ОК 10, ПК 1.4.

Задания закрытого типа на установление соответствия

1. Установите соответствие между названием типа тестов по знанию кода и его описанием

Тип теста по знанию кода		Описание теста
1)	Черный ящик	А) тестирование основанное на анализе внутренней структуры компонентов или системы. У тестировщика есть доступ к внутренней структуре и коду приложения.
2)	Белый ящик	Б) комбинация методов, состоящая в том, что к части кода архитектуры у тестировщика есть, а к части кода – нет.
3)	Серый ящик	В) тестирование системы, функциональное или нефункциональное, без знания внутренней структуры и компонентов системы. У тестировщика нет доступа к внутренней структуре и коду приложения либо в процессе тестирования он не обращается к ним.

Правильный ответ

1	2	3
В	А	Б

Компетенции: ОК 1, ОК 2, ОК 9, ПК 1.4.

2. Установите соответствие между проблемами в коде и способами их исправления

Проблема в коде	Способ исправления
1) Длинный метод	А) Вынести повторяющиеся части в отдельный метод
2) Повторяющийся код	Б) Переименовать переменные на понятные
3) Непонятные названия переменных	В) Разбить условие на части и дать им названия
4) Сложное условие	Г) Разбить на несколько меньших методов

Правильный ответ

1	2	3	4
Б	В	Г	А

Компетенции: ОК 2, ОК 9, ПК 1.5.

3. Установите соответствие между функционалом отладчика и описанием действия

	Функционал отладчика	Описание действия
1)	Точка останова	А) Переход к следующей строке кода, не входя в вызываемые функции.
2)	Пошаговое выполнение	Б) Приостановка выполнения программы в указанном месте, чтобы исследовать состояние.
3)	Просмотр переменных	В) Отображение списка активных функций, вызванных в текущий момент, для отслеживания потока управления.
4)	Стек вызовов	Г) Позволяет перейти внутрь вызываемой функции для более детального анализа.
5)	"Шаг с обходом"	Д) Отображение и изменение значений переменных в текущем контексте (области видимости) программы.
6)	"Шаг с заходом"	Е) Продолжение выполнения программы до следующей точки останова или завершения программы.
7)	Продолжить выполнение	Ж) Построчное выполнение кода, позволяющее контролировать каждый шаг и изменение состояния программы.

Правильный ответ

1	2	3	4	5	6	7
Б	Ж	Д	В	А	Г	Е

Компетенции: ОК 2, ОК 9, ПК 1.3.

4. Установите соответствие между методами покрытия с их описаниями

Название метода

Описание метода

- | | |
|---|--|
| 1) Метод покрытия операторов | А) Согласно методу каждое направление перехода должно быть реализовано, по крайней мере, один раз. Он включает в себя критерий покрытия операторов, так как при выполнении всех направлений переходов выполнятся все операторы, находящиеся на этих направлениях. |
| 2) Метод покрытия решений (покрытия переходов) | Б) Согласно этому методу тестирования является выполнение каждого оператора программы хотя бы один раз |
| 3) Метод покрытия условий | В) Критерий этого метода удовлетворяет также и критериям покрытия решений, покрытия условий и покрытия решений/условий. |
| 4) Метод покрытия решений/условий | Г) В соответствии с этим методом записывается число тестов, достаточное для того, чтобы все возможные результаты каждого условия в решении выполнялись, по крайней мере, один раз. |
| 5) Метод комбинаторного покрытия условий | Д) Критерий этого метода требует такого достаточного набора тестов, чтобы все возможные результаты каждого условия выполнялись по крайней мере один раз, все результаты каждого решения выполнялись по крайней мере один раз и, кроме того, каждой точке входа передавалось управление по крайней мере один раз. |

Правильный ответ

1	2	3	4	5
Б	А	Д	В	Г

Компетенции: ОК 1, ОК 2, ОК 9, ПК 1.4.

5. Установите соответствие между действиями и их целями при улучшении кода

Действие		Цель	
1)	Убрать ненужные комментарии	А)	Упростить дальнейшую поддержку
2)	Упростить сложные выражения	Б)	Сделать код более читаемым
3)	Удалить неиспользуемый код	В)	Облегчить понимание логики
4)	Группировать похожие операции	Г)	Уменьшить размер программы

Правильный ответ

1	2	3	4
Г	А	Б	В

Компетенции: ОК 1, ОК 2, ОК 9, ПК 1.5.

6. Соотнесите тип ошибки с методом или инструментом, который наиболее эффективно помогает её обнаружить

Тип ошибки		Метод/Инструмент обнаружения	
1)	Ошибка сегментации	А)	Тестирование с использованием различных наборов входных данных и граничных условий.
2)	Утечка памяти	Б)	Инструменты статического анализа кода, которые могут выявлять неиспользуемые переменные и объекты.
3)	Логическая ошибка	В)	Отладчик, позволяющий пошагово выполнять код и анализировать состояние переменных.
4)	Необработанное исключение	Г)	Программы-анализаторы, выявляющие проблемы, связанные с некорректным выделением и освобождением памяти.
5)	Неиспользуемая переменная	Д)	Запуск программы в отладчике под управлением операционной системы с мониторингом выделенной памяти.

Правильный ответ

1	2	3	4	5
---	---	---	---	---

Д	Г	В	А	Б
---	---	---	---	---

Компетенции: ОК 2, ОК 9, ПК 1.3, ПК 1.4.

Задания закрытого типа на установление правильной последовательности

1. Установите правильную последовательность этапов процесса тестирования ПО

- А) Подготовка тестовых данных
- Б) Исполнение тестов
- В) Анализ результатов
- Г) Документирование результатов

Правильный ответ: А, Б, В, Г

Компетенции: ОК 1, ОК 2, ПК 1.4.

2. Установите правильную последовательность этапов жизненного цикла разработки программного обеспечения

- А) Разработка
- Б) Внедрение
- В) Проектирование
- Г) Тестирование
- Д) Анализ требований

Правильный ответ: Д, В, А, Г, Б

Компетенции: ОК 1, ОК 2, ПК 1.4.

3. Установите правильную последовательность шагов в процессе управления дефектами

- А) Классификация дефекта
- Б) Выявление дефекта
- В) Исправление дефекта
- Г) Подтверждение исправления

Правильный ответ: Б, А, В, Г,

Компетенции: ОК 2, ОК 9, ПК 1.4.

4. Установите правильную последовательность этапов тестирования приложения. Этапы:

- А) Запуск бета-тестирования
- Б) Сбор отзывов и исправление ошибок

- В) Подготовка тестового окружения
 - Г) Проведение функционального тестирования
 - Д) Проведение тестирования производительности
 - Е) Проведение тестирования пользовательского интерфейса
- Правильный ответ: В, Г, Д, Е, А, Б
- Компетенции: ОК 1, ОК 2, ОК 9, ПК 1.4

5. Тестирование программного обеспечения включает в себя целый комплекс действий, аналогичных последовательности процессов разработки программного обеспечения. Установите правильную последовательность тестирования. В него входят:

- А) выполнение тестов
- Б) изучение результатов тестирования
- В) проектирование теста
- Г) написание тестов
- Д) тестирование тестов
- Е) постановка задачи для теста

Правильный ответ: Е, В, Г, Д, А, Б

Компетенции: ОК 1, ОК 2, ПК 1.4.

6. Расположите следующие действия в правильной последовательности, когда будет использован отладчик для анализа и исправления ошибки в программе

- А) просмотреть значения переменных в текущем контексте.
- Б) запустить программу в отладчике.
- В) установить точку останова в подозрительном месте кода.
- Г) пошагово выполнить код, отслеживая изменение значений переменных.
- Д) проанализировать стек вызовов для определения пути выполнения программы.

Е) определить причину ошибки на основе полученной информации.

Правильный ответ: Б, В, Г, А, Д, Е

Компетенции: ОК 1, ОК 9, ПК 1.3

7. Укажите правильную последовательность шагов при отладке программного модуля с использованием логгирования

- А) проанализировать лог-файлы для выявления аномалий или ошибок.
- Б) вставить операторы логгирования в ключевые места кода.
- В) запустить программу с включенным логгированием.

Г) удалить или закомментировать операторы логгирования после исправления ошибки.

Д) воспроизвести сценарий, приводящий к ошибке.

Правильный ответ: Б, В, Д, А, Г

Компетенции: ОК 1, ОК 2, ОК 9, ПК 1.3

8. Установите правильную последовательность шагов при проведении рефакторинга кода:

А) Проанализировать покрытие тестами и при необходимости добавить тесты

Б) Выделить участок кода, требующий улучшения

В) Внести минимальные изменения и убедиться в прохождении тестов

Г) Выбрать конкретный прием рефакторинга

Д) Зафиксировать изменения в системе контроля версий

Правильный ответ: А, Б, Г, В, Д

Компетенции: ПК 1.5

9. Установите правильную последовательность этапов оптимизации программного кода:

А) Провести профилирование для выявления "узких мест"

Б) Выбрать целевые метрики производительности

В) Реализовать оптимизацию в наиболее критичном участке

Г) Измерить результат и сравнить с исходными показателями

Д) Протестировать функциональность после оптимизации

Правильный ответ: Б, А, В, Г, Д

Компетенции: ПК 1.5

10. Установите правильную последовательность работы с "запахам кода" (code smells):

А) Составить приоритетный список найденных "запахов"

Б) Выполнить рефакторинг для устранения наиболее критичных проблем

В) Провести статический анализ кода

Г) Определить категории "запахов": архитектурные, модульные, внутриметодные

Д) Протестировать работоспособность после изменений

Правильный ответ: В, Г, А, Б, Д

Компетенции: ПК 1.5

Задания открытого типа

Задания открытого типа на дополнение

Напишите пропущенное слово (словосочетание).

1. Процесс, посредством которого проверяется правильность программы называется _____.

Правильный ответ: тестирование

Компетенции: ОК 1, ОК 2, ПК 1.4

2. _____ — совокупность действий по обеспечению работоспособности и актуальности программного обеспечения.

Правильный ответ: поддержка

Компетенции: ОК 2, ПК 1.4.

3. _____ — процесс поиска дефектов в программе.

Правильный ответ: отладка

Компетенции: ОК 2, ПК 1.3.

4. _____ — конкретный набор условий, входных данных и ожидаемых результатов, используемый для проверки определенной функциональности программы (ПО).

Правильный ответ: тест-кейс

Компетенции: ОК 2, ПК 1.4.

5. _____ — документ, описывающий все этапы тестирования, цели и задачи, а также критерии оценки для конкретного проекта.

Правильный ответ: План тестирования

Компетенции: ОК 10, ПК 1.4.

6. _____ — метод отладки, при котором программа выполняется шаг за шагом для отслеживания значений переменных.

Правильный ответ: Трассировка

Компетенции: ОК 9, ПК 1.3.

7. Процесс улучшения структуры существующего кода без изменения его внешнего поведения называется _____.

Правильный ответ: рефакторинг

Компетенции: ПК 1.5

8. Метка в коде, которая приостанавливает выполнение программы в определенной точке для анализа состояния переменных и потока выполнения, называется _____.

Правильный ответ: точка останова (breakpoint)

Компетенции: ПК 1.3

9. Метод рефакторинга, при котором часть кода выделяется в отдельный метод с понятным названием, называется _____.

Правильный ответ: извлечение метода (extract method)

Компетенции: ПК 1.5

10. Процесс изменения кода с целью повышения скорости выполнения или уменьшения потребления ресурсов называется _____.

Правильный ответ: оптимизация

Компетенции: ПК 1.5

Задания открытого типа с кратким свободным ответом

1. Какой документ описывает функциональные требования к программному обеспечению?

Правильный ответ: Техническое задание

Компетенции: ОК 10, ПК 1.4.

2. Какой из типов тестирования фокусируется на производительности приложения?

Правильный ответ: Нагрузочное тестирование

Компетенции: ОК 2, ПК 1.4.

3. Что такое валидация (validation) в контексте тестирования программного обеспечения?

Правильный ответ: подтверждение соответствия требованиям пользователя

Компетенции: ОК 2, ПК 1.4.

4. Что такое проверка (verification) в контексте тестирования программного обеспечения?

Правильный ответ: подтверждение соответствия спецификациям
Компетенции: ОК 2, ПК 1.4.

5. Как называется вид тестирования, направленный на выявление уязвимостей и обеспечение защиты данных?

Правильный ответ: тестирование безопасности
Компетенции: ОК 2, ПК 1.4.

6. Как называется тестирование, проводимое конечными пользователями для оценки готовности продукта к выпуску?

Правильный ответ: приёмочное тестирование (или пользовательское тестирование)
Компетенции: ОК 2, ПК 1.4.

7. Как называется инструмент, который позволяет выполнять программу пошагово и отслеживать значения переменных?

Правильный ответ: отладчик (debugger)
Компетенции: ПК 1.3

8. Какой метод отладки предполагает вывод промежуточных значений переменных в консоль или файл?

Правильный ответ: логгирование (logging)
Компетенции: ПК 1.3

9. Какой прием рефакторинга следует применить, если в коде многократно повторяется один и тот же фрагмент?

Правильный ответ: вынести повторяющийся код в отдельный метод
Компетенции: ПК 1.5

10. Что такое "запахи кода" (code smells)?

Правильный ответ: признаки потенциальных проблем в коде, указывающие на необходимость рефакторинга
Компетенции: ПК 1.5

11. Какой инструмент используется для измерения производительности кода и выявления "узких мест"?

Правильный ответ: профайлер (profiler)
Компетенции: ПК 1.5

Задания открытого типа с развернутым ответом

1. Отладка критической ошибки в модуле работы с массивами

Задачи:

- найти причину ошибки сегментации в функции обработки массива
- использовать отладчик GDB для анализа падения программы
- предложить исправление кода

Исходный код для анализа:

```
#include <iostream>

void processArray(int *arr, int size) {
    for (int i = 0; i <= size; ++i)
    {
        arr[i] *= 2;    } // Ошибка здесь
    }

int main() {
    int size = 5;
    int *array = new int[size];
    // Инициализация массива
    for (int i = 0; i < size; ++i) {
        array[i] = i + 1;
    }
    processArray(array, size);
    // Вывод результата (проверка)
    for (int i = 0; i < size; ++i) {
        std::cout << array[i] << " ";
    }
    std::cout << std::endl;
    delete[] array;
    array = nullptr;
    return 0; }
```

Время выполнения – 5 мин.

Ожидаемый ответ (один из возможных вариантов):

Диагностика в GDB: Запуск: `gdb ./program`; Установка точки останова: `break process_array`; Запуск: `run`; Анализ падения: `backtrace`

Найденная ошибка:

Выход за границы массива в условии цикла `i <= size` вместо `i < size`

Исправление кода:

```
void processArray(int *arr, int size) {
    for (int i = 0; i < size; ++i) // Ошибка здесь
    {
        arr[i] *= 2;    }
```

}

Критерии оценивания:

- правильное использование команд GDB для диагностики
- точное определение причины ошибки
- корректное исправление кода
- понимание работы с динамической памятью

Компетенции: ОК 1, ОК 2, ПК 1.3

2. Отладка бесконечного цикла в модуле математических вычислений

Задачи:

- использовать отладчик для анализа бесконечного цикла
- найти логическую ошибку в условии выхода из цикла
- предложить корректный алгоритм

Исходный код для анализа:

```
#include <iostream>
int calFactorial(int n) {
    int result = 1;
    int i = 1;

    while (i <= n) { // Ошибка в условии
        result *= i;
    }
    return result;
}

int main() {
    int n = 5; // Число, для которого вычисляем факториал
    int fact = calcFactorial(n);
    std::cout << "Факториал " << n << " равен: " << fact << std::endl;
    return 0; }
```

Время выполнения – 5 мин.

Ожидаемый ответ (один из возможных вариантов):

Диагностика в отладчике: Запуск программы в отладчике; Установка точки останова в цикле; Наблюдение за изменением переменной *i*; Обнаружение, что *i* не изменяется.

Найденная ошибка:

Отсутствует инкремент переменной *i* в цикле

Исправление кода:

```
while (i <= n) {    result *= i;    i++; } // Добавить инкремент
```

Критерии оценивания:

- умение использовать отладчик для анализа циклов
- правильное определение причины бесконечного цикла
- точное исправление логической ошибки
- понимание работы циклов и условий

Компетенции: ОК 1, ОК 2, ПК 1.3

3. Разработка стратегии тестирования сложного модуля. Необходимо протестировать модуль расчета страховых премий, который учитывает множество факторов: возраст клиента, историю вождения, тип автомобиля, регион проживания. Модуль содержит сложную логику вычислений с множеством условий и исключений.

Задачи:

- разработайте комплексную стратегию тестирования данного модуля.
- перечислить виды тестирования, указать причину выбора
- разработать тестовые случаи (3-4 ключевых тест-кейсов)
- представить тестовые данные для покрытия различных сценариев
- проверить граничные условия и исключительные ситуации

Время выполнения – 20 мин.

Ожидаемый ответ (один из возможных вариантов):

1. Виды тестирования и их обоснование

Для комплексного тестирования модуля можно применить следующие виды тестирования:

Функциональное тестирование - проверка корректности расчетов согласно бизнес-требованиям

Тестирование граничных значений - анализ поведения системы на крайних допустимых значениях

Эквивалентное разделение - группировка входных данных на классы с ожидаемо одинаковым поведением

Табличное тестирование принятия решений - проверка сложной логики с множеством условий

Негативное тестирование - проверка обработки некорректных данных и исключительных ситуаций

2. Ключевые тест-кейсы

Тест-кейс 1: Стандартный расчет для надежного водителя

Входные данные: возраст 35 лет, стаж вождения 10 лет, 0 аварий, автомобиль - седан, регион - Москва

Ожидаемый результат: стандартная ставка с минимальной надбавкой

Тест-кейс 2: Молодой водитель с рисками

Входные данные: возраст 19 лет, стаж 1 год, 2 аварии, автомобиль - спортивный, регион - крупный город

Ожидаемый результат: повышенная ставка с максимальными надбавками за возраст и риск

Тест-кейс 3: Водитель преклонного возраста

Входные данные: возраст 70 лет, стаж 40 лет, 0 аварий, автомобиль - семейный, регион - сельская местность

Ожидаемый результат: средняя ставка с надбавкой за возраст, но скидкой за стаж и регион

Тест-кейс 4: Пограничные значения стажа

Входные данные: возраст 25 лет, стаж 3 года (граница для скидки), 0 аварий, стандартный автомобиль

Ожидаемый результат: переход на льготный тариф

3. Тестовые данные для различных сценариев

– Данные по возрасту:

Минимальный возраст (18 лет)

Граничные значения (21, 25, 65 лет)

Максимальный возраст (75 лет)

Некорректные значения (17, 100 лет)

– Данные по стажу вождения:

Нулевой стаж

1-2 года (начальный период)

3-5 лет (переходный период)

10+ лет (опытный водитель)

Стаж больше возраста (невозможная ситуация)

– Данные по авариям:

0 аварий (идеальная история)

1-2 аварии (повышенный риск)

3+ аварий (критический риск)

Отрицательное количество аварий

4. Проверка граничных условий и исключительных ситуаций

Граничные условия:

Возраст: 17/18/21/24/25/64/65/66/74/75 лет

Стаж: -1/0/1/2/3/4/5/49/50/51 лет

Количество аварий: -1/0/1/2/3/10/100

Исключительные ситуации:

Пустые или нулевые входные данные

Несуществующие регионы или типы автомобилей

Стаж вождения больше возраста минус 16 лет

Одновременное достижение нескольких граничных условий

Очень большие числа (переполнение)

Методы проверки:

Валидация входных параметров перед расчетом

Проверка обработки исключений и вывода понятных сообщений об ошибках

Тестирование производительности при больших объемах данных

Проверка согласованности расчетов (например, при одинаковых входных данных - одинаковый результат)

Данная стратегия обеспечит полное покрытие функциональности модуля и выявление потенциальных ошибок в сложной логике расчетов.

Критерии оценивания:

- полнота стратегии тестирования
- релевантность тест-кейсов и тестовых данных
- учет различных видов тестирования
- проработанность проверки граничных условий

Компетенции: ОК 1, ОК 2, ПК 1.4