

**Министерство науки и высшего образования
Российской Федерации
Министерство науки, высшего образования и инноваций
Кыргызской Республики**

**Государственное образовательное учреждение
высшего профессионального образования
Кыргызско-Российский Славянский университет имени
первого президента Российской Федерации Б.Н. Ельцина**

Естественно-технический факультет

Кафедра Информационных и вычислительных технологий

**Фонд
оценочных средств**

по дисциплине «Экспериментальные методы исследования качества ПО»

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

09.03.04 - РФ, 710400 - КР Программная инженерия
(код и наименование направления подготовки)

Квалификация

бакалавр

Фонд оценочных средств предназначен для контроля знаний обучающихся по направлению подготовки 09.03.04 – РФ, 710400 - КР «Программная инженерия» по дисциплине «Экспериментальные методы исследования качества ПО»

Фонд оценочных средств рассмотрен и утвержден на заседании кафедры Информационных и вычислительных технологий

Заведующий кафедрой
д.т.н., проф.



Лыченко Н.М.

Исполнители (разработчики):

К.т.н., доцент каф. ИВТ Мусина И.Р.



К.т.н., доцент каф. ИВТ Манжикова С.С.

СОГЛАСОВАНО:

и.о. декана



Комарцов Н.М.

Раздел 1. Перечень компетенций, с указанием этапов их формирования в процессе освоения дисциплины/практики

Формируемые компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций	Виды оценочных средств/ шифр раздела в данном документе
ПК-2: Способность оценивать временную и емкостную сложность программного обеспечения	<u>Знать:</u> - методы и уровни тестирования; - измерения, связанные с тестированием.	Блок А – задания репродуктивного уровня - Устный опрос
	<u>Уметь:</u> - планировать процесс тестирования; - разрабатывать тесты.	Блок В – задания реконструктивного уровня - Практические задания;
	<u>Владеть:</u> - навыками оценки качества программ; - навыками автоматизации модульного тестирования; - навыками проведения интеграционного и системного тестирования.	Блок С – задания практико-ориентированного и/или исследовательского уровня - Практические задания - Самостоятельная работа

Раздел 2. Технологическая карта дисциплины «Экспериментальные методы исследования качества ПО»

Курс 4, семестр 7, Количество ЗЕ -4, Отчетность – зачет с оценкой

Название модулей дисциплины согласно РПД	Контроль	Форма контроля	зачетный минимум	зачетный максимум	График контроля
1. Введение в дисциплину	текущий	Сдача практических работ №1 -№2	7	12	9
	рубежный	Сдача самостоятельных работ №1-№2	3	5	9
2. Модульное тестирование. Инспектирование кода.	текущий	Сдача практических работ №3- №4	7	12	18
	рубежный	Сдача самостоятельных работ №3-№4	3	5	18
3. Интеграционное тестирование.	текущий	Сдача практических работ №5-№6	7	13	29
	рубежный	Сдача самостоятельных работ №5-№6	3	5	29

4. Системное тестирование. Организация процесса тестирования	текущий	Сдача практических работ №7-№8	7	13	36
	рубежный	Сдача самостоятельных работ №7-№8	3	5	36
ВСЕГО за семестр			40	70	
Промежуточный контроль (Зачет с оценкой)			20	30	
Семестровый рейтинг по дисциплине			60	100	

Раздел 3. Типовые контрольные задания и иные материалы, необходимые для оценки планируемых результатов обучения по дисциплине (оценочные средства)

Блок А

Вопросы для опроса

Модуль 1

1. В чем отличие верификации ПО от валидации ПО?
2. Какие бывают виды верификации ПО?
3. Что такое инспектирование ПО?
4. Что такое тестирование ПО?
5. Что такое тестирование дефектов?
6. Что такое статистическое тестирование?
7. Что такое исчерпывающее тестирование?
8. Какие бывают уровни тестирования?
9. Что такое регрессионное тестирование
10. Что такое отладка? Как проводить процесс отладки?
11. . Планирование верификации и аттестации: Основные компоненты плана.
12. V – модель планирования разработки ПО.
13. Различия между объектно-ориентированным и функционально-ориентированными системами тестирования.
14. В чем отличие между методиками тестирования «черного ящика», «белого ящика», «серого ящика»?

Модуль 2

1. Что такое модульное тестирование?
2. Что должно быть охвачено при модульном тестировании?
3. Кто должен выполнять модульное тестирование?
4. Что такое тестовый вариант? Как его создать?
5. Изложите способа разбиения по областям эквивалентности.
6. В чем заключается метод граничных значений?
7. Тестовые сценарии. Изложите методику отбора тестовых сценариев.
8. В чем преимущества автоматизированного тестирования модуля?
9. Какие ошибки нельзя определить при модульном тестировании?
10. В чем заключается TDD (тестово-управляемая разработка ПО)?
11. Какие условия проведения инспектирования?
12. Что содержат технологические карты?
13. Какие ошибки выявляются при инспектировании?
14. Цель автоматического статического анализа и как его проводить?
15. Метод CLEAN ROOM («Чистая комната»). В чем его особенность?

Модуль 3

1. Что такое тестирование интеграции (сборки)? Зачем его проводить?
2. В чем заключается методика проведения тестирования сборки?
3. Какие имеются способы тестирования интеграции?
4. В чем особенности интеграционного тестирования способом восходящего тестирования (Down-Top)?
5. Что такое тестовый драйвер? Когда и зачем он применяется?
6. В чем особенности интеграционного тестирования способом нисходящего тестирования (Top – Down)?
7. Что такое «заглушка»? Когда и зачем она применяется?
8. Что такое «Сэндвич- тестирование» и как его проводить?
9. Что такое «Модифицированное Сэндвич- тестирование» и как его проводить?
10. Ошибки при интеграционном тестировании.
11. Какие бывают ошибки при тестировании интерфейсов.

Модуль 4

1. В чем заключается системное тестирование и чем оно отличается от интеграционного?
2. В чем заключается тестирование производительности?
3. Что такое профилирование программы?
4. Что такое стрессовое тестирование и как его проводить?
5. Что такое нагрузочное тестирование и как его проводить?
6. . Какие бывают метрики кода?
7. Что такое цикломатическая сложность программы?
8. Какие бывают способы расчета цикломатической сложности программы?
9. Зачем необходимо знать цикломатическую сложность программы?
10. Как построить потоковый граф для программы?
11. Что такое независимые пути работы программы, как определить их количество?
12. Изложите способы проведения тестирования графического интерфейса пользователя.
13. Что такое юзабилити-тестирование?
14. Для чего используется программа SELENIUM?

Блоки В и С

Практические задания

Практическая работа 1. Тестирование с использованием технологии «черного ящика». Разработка тестовых вариантов на основе технического задания

1. . Разработать ТЗ на программу обучения решению квадратного уравнения.
2. Разработать тестовые варианты.
3. Провести тестирование методом "черного ящика"
4. Составить отчет в виде таблицы по результатам тестирования.

Практическая работа 2. . Разработка тестовых вариантов на основе способов разбиения по областям эквивалентности и анализа граничных значений.

1. Подготовить ТЗ на разработку программы, реализующую бинарного поиска заданного элемента в последовательности.
2. Разработать тестовые варианты на основе способов разбиения по областям эквивалентности и анализа граничных значений.
3. Провести тестирование с использованием разработанных тестовых вариантов с использованием методики «черного ящика».
4. Подготовить отчет по результатам в виде таблицы

Практическая работа 3. Проведение модульного тестирования

1.1.Подготовить ТЗ на разработку программы, реализующей нахождение числа Фибаначчи по заданному порядковому номеру.

1.2 Создать тестовый проект для нахождения числа Фибаначчи.

Подготовить тестовые варианты.

1.3. Написать Unit- тесты для проверки значений скалярной величины с использованием методов класса Assert.

1.4.Провести тестирование. Подготовить таблицу с результатами тестирования

2.1. Создать тестовый проект для нахождения корней квадратного уравнения. Результаты работы программы сформировать в виде списка (коллекции).

2.2. Написать Unit- тесты для для проверки значений коллекции с использованием методов класса CollectionAssert.

2.3. Провести тестирование. Сформировать таблицу с результатами.

3.1 Подготовить Unit- тесты для проверки тестируемых значений с заданной точностью ожидаемых результатов.

3.2. Провести тестирование. Подготовить отчет с результатами.

3.3. Исследовать процесс тестирования с различными значениями заданной точности ожидаемых результатов.

Практическая работа 4. Автоматический статический анализ программ. Настройка автоматического анализатора (в MS Visual Studio).

1. Для тестового проекта по реализации бинарного поиска провести автоматический статический анализ программы (можно использовать любую IDE)
2. Проанализировать полученные результаты.
3. Подготовить отчет с анализом результатов инспектирования.
4. Скорректировать код согласно результатам анализа.

5. Подготовить отчет по результатам анализа.

Практическая работа 5. Тестирование графического интерфейса.

1. Разработать Web-приложение для реализации калькулятора арифметических операций.
2. провести тестирование интерфейса с использованием SELENIUM IDE с использованием режимов **verify** и **assert**.
3. Составить отчет по результатам тестирования.

Практическая работа 6. Расчет метрик кода. Расчет цикломатической сложности кода.

1. Построить потоковый граф для алгоритма бинарного поиска.
2. Рассчитать цикломатическую сложность тремя способами вручную по потоковому графу.
3. Показать независимые пути.
4. Подготовить тестовые варианты, покрывающие независимые пути.
5. Получить метрики кода (поддерживаемость, цикломатическую сложность, сцепление классов, количество строк кода и т.д.) в среде разработки (IDE).
6. Сравнить результат расчета цикломатической сложности, полученные вручную, с результатом, полученным автоматически.
7. Дать оценку полученным метрикам кода.
8. Подготовить отчет по результатам анализа.

Практическая работа 7. Нагрузочное тестирование

1. Выбрать среду (IDE), в которой будут проводиться нагрузочное тестирование.
2. Провести нагрузочное тестирование web-приложения, реализующее бинарный поиск, с использованием инструментария выбранной IDE.
3. Исследовать работу приложения с наращиванием нагрузки.
4. Подготовить отчет по проведенному тестированию: как падает производительность приложения с увеличением нагрузки.

Практическая работа 8. Разработка плана тестирования для заданной программы.

1. Выбрать программу для тестирования и согласовать выбор с преподавателем.
2. Разработать план тестирования для заданной программы.
3. Разработать график работ по тестированию программы.
4. Провести оценку рисков по процессу тестирования.
5. Подготовить отчет по проделанной работе.

Контрольные работы

Контрольная работа 1. Тестирование с использованием технологии «черного ящика». Разработка тестовых вариантов на основе технического задания.

1. Для программы с заданным ТЗ разработать тестовые варианты.

Контрольная работа 2. Разработка тестовых вариантов на основе способов разбиения по областям эквивалентности и анализа граничных значений.

2. Разработать тестовые варианты для заданных индивидуально задач на основе способов разбиения по областям эквивалентности и анализа граничных значений.
3. Провести тестирование.
4. Задokumentировать результаты.

Контрольная работа 3. Проведение модульного тестирования в MS Visual Studio. Создание тестового проекта

1. Провести модульного тестирование в MS Visual Studio (можно в другой интегрированной среде) для заданного класса. При тестировании использовать методы классов Assert и Collection Assert.
2. Показать процент покрытия кода тестами.

Контрольная работа 4. Автоматический статический анализ программ. Настройка автоматического анализатора

1. Провести автоматический статический анализ заданной программы (в любой IDE).
2. Проанализировать полученные результаты.

Контрольная работа 5. Тестирование графического интерфейса.

1. Провести тестирование интерфейса для заданного приложения с использованием SELENIUM IDE.
2. Продемонстрировать процесс тестирования с использованием режима assert и verify.

Контрольная работа 6. Расчет метрик кода. Расчет цикломатической сложности кода.

1. Рассчитать для программы, используемой в контр. работе 2, метрики кода и коэффициент цикломатической сложности в интегрированной среде программирования.
2. Просчитать коэффициент цикломатической сложности и сравнить с результатом, полученным автоматически.

Контрольная работа 7. Нагрузочное тестирование.

1. Провести стрессовое и нагрузочное тестирование для заданного web-приложения.
2. Подготовить отчет по проведенному тестированию: как падает производительность приложения с увеличением нагрузки.

Контрольная работа 8. Разработка плана тестирования для заданной программы.

1. Разработать план тестирования для программы, заданной индивидуально.
2. Провести тестирование согласно разработанному плану.

Блок D (промежуточный контроль)

Вопросы для проверки уровня обученности ЗНАТЬ

1. Виды верификация, аттестация, инспектирование и их цели.
2. Тестирование ПО (тестирование дефектов, статистическое тестирование, исчерпывающее тестирование).
3. Уровни тестирования.
4. Критерии тестирования.
5. Измерение тестирования.

6. Планирование верификации и аттестации: V – модель планирования разработки ПО, основные компоненты плана.
7. Роли в процессе инспектирования.
8. Создание технологических карт.
9. Ошибки, выявляемые при инспектировании.
10. Автоматический статический анализ программ.
11. Ошибки, обнаруживаемые статическим анализом.
12. Этапы статического анализа.
13. Метод CLEAN ROOM («Чистая комната»).
14. Модульное тестирование. Что должно быть охвачено? Кто должен его выполнять?
15. Преимущества автоматизированного тестирования модуля.
16. Тестово-управляемая разработка и модульное тестирование.
17. Тестирование интеграции.
18. Способы тестирования интеграции.
19. Драйвера. Заглушки
20. Сэндвич- тестирование.
21. Ошибки при интеграционном тестировании
22. Регрессионное тестирование.
23. Различия между объектно-ориентированным и функционально-ориентированными системами тестирования.
24. Тестирование дефектов. Цель тестирования дефектов. Общая модель процесса тестирования дефектов.
25. Тестовые сценарии. Методика отбора тестовых сценариев.
26. Ошибки при тестировании интерфейсов.
27. Коэффициент цикломатической сложности и его значение.
28. Тестирование производительности.
29. Стрессовое тестирование.
30. Нагрузочное тестирование.
31. Методик тестирования «черного ящика», «белого ящика», «серого ящика».

Задачи/здания для проверки уровня обученности УМЕТЬ и ВЛАДЕТЬ

1. Разработать тестовые варианты с использованием способа разбиения по областям эквивалентности.
2. Разработать тестовые сценарии.
3. Разработать план тестирования, оценить риски
4. Оценить разработанный код программы с использованием метрик.
5. Построить потоковый граф программы. Определить независимые пути тестирования.
6. Рассчитать цикломатическую сложность программы.
7. Провести модульное тестирование с использованием методов класса ASSERT (для скаляра).
8. Провести модульное тестирование с использованием методов класса Collection ASSERT (для коллекции).
9. Протестировать графический интерфейс (вручную и автоматически).
10. Провести интеграционное тестирование.
11. Провести системное тестирование.
12. Провести профилирование программы.

Пример

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ ПРОМЕЖУТОЧНОГО КОНТРОЛЯ № 3

1. Вопросы для проверки уровня обученности ЗНАТЬ
 - 1.1 Что входит в понятие модульное тестирование?
 - 1.2 Что означает тестирование по стратегии «белого ящика» и когда оно применяется?

1.3 Опишите метод разбиения по классам эквивалентности. Для чего он применяется?

2. Задания для проверки уровня обученности УМЕТЬ и ВЛАДЕТЬ

2.1. Постройте тестовые варианты с использованием метода разбиения по классам эквивалентности для задачи определения суммы любых целых чисел, лежащих в диапазоне от -10000 до +10000.

2.2. Создайте тестовый проект и проведите Unit-тестирование по разработанным тестовым вариантам.

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

Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Применяемые оценочные средства:

- Сдача практических работ на практических занятиях в соответствии с технологической картой дисциплины (текущая и рубежная аттестация),
- Прохождение тестирования,
- Устный опрос.
- Письменный опрос по экзаменационным билетам (промежуточная аттестация - зачет с оценкой),
- Сдача самостоятельных работ.

Все виды оценочных средств оцениваются в соответствии со шкалами оценивания.

Устный опрос на практических занятиях по отдельным темам проводится в течение всего периода обучения дисциплине. Результаты опроса учитываются при оценивании практических работ.

ШКАЛА ОЦЕНИВАНИЯ ПРАКТИЧЕСКИХ (ЛАБОРАТОРНЫХ) РАБОТ (текущий контроль)

- 85-100 % - Демонстрирует полное понимание проблемы. Все требования, предъявляемые к заданию, выполнены.
- 70-84 % - Демонстрирует значительное понимание проблемы. Все требования, предъявляемые к заданию, выполнены.
- 60-69 % - Демонстрирует частичное понимание проблемы. Большинство требований, предъявляемых к заданию выполнены.
- 31-60 % - Демонстрирует небольшое понимание проблемы. Многие требования, предъявляемые к заданию не выполнены.
- 0-30 % - Демонстрирует непонимание проблемы и даже не было попытки решить задачу.

ШКАЛА ОЦЕНИВАНИЯ КОНТРОЛЬНЫХ РАБОТ (рубежный контроль)

- 85-100 % - Демонстрирует полное понимание проблемы. Все задания выполнены.
- 70-84 % - Демонстрирует значительное понимание проблемы. Все задания выполнены, но содержат некоторые неточности.
- 60-69 % - Демонстрирует частичное понимание проблемы. Большинство требований, предъявляемых к заданию, выполнены.
- 31-60 % - Демонстрирует небольшое понимание проблемы. Многие требования, предъявляемые к заданию, не выполнены.
- 0-30 % - Демонстрирует непонимание проблемы или нет ответа и даже не было попытки решить задачу.

ШКАЛА ОЦЕНИВАНИЯ ПИСЬМЕННОГО ОПРОСА

(промежуточный контроль – «ЗНАТЬ»)

Отметкой (7-10- баллов) оценивается ответ, который показывает прочные знания теоретических основ дисциплины, понимание и правильное применение терминологии, правильные ответы на 75-100% вопросов

Отметкой (5-6 баллов) оценивается ответ, который показывает знание теоретических основ дисциплины, но неполное понимание и не всегда правильное применение терминологии, даны правильные ответы на 50-74% вопросов, в ответах допущено некоторое количество неточностей.

Отметкой (3-4 баллов) оценивается ответ, свидетельствующий о знакомстве с некоторыми теоретическими основами дисциплины. Даны правильные ответы на 25-49% вопросов, допущены неточности и ошибки.

Отметкой (2 балла) оценивается ответ, обнаруживающий незнание теоретических основ дисциплины. Отмечается отсутствие логичности и последовательности в ответе. Менее 25% правильных ответов. Допущены серьезные ошибки в содержании ответа.

Отметкой (0-1 балл) оценивается ответ, при котором студент демонстрирует непонимание поставленных вопросов, или нет ответа.

ШКАЛА ОЦЕНИВАНИЯ ПРАКТИЧЕСКИХ ЗАДАНИЙ

(промежуточный контроль – «УМЕТЬ и ВЛАДЕТЬ»)

Отметкой (8-10 баллов) оценивается ответ, при котором студент правильно отвечает на поставленные вопросы, демонстрирует полное понимание проблемы. Все требования, предъявляемые к заданию, выполнены.

Отметкой (5-7 баллов) оценивается ответ, при котором студент в основном правильно отвечает на поставленные вопросы. Демонстрирует значительное понимание проблемы. Большинство требований, предъявляемых к заданию, выполнены.

Отметкой (2-4 баллов) оценивается ответ, при котором студент в основном неправильно отвечает на поставленные вопросы. Демонстрирует частичное или небольшое понимание проблемы. Многие требования, предъявляемые к заданию, не выполнены.

Отметкой (0 -1 балл) оценивается ответ, при котором студент демонстрирует непонимание проблемы или нет ответа и даже не было попытки решить задачи.

В экзаменационный билет включены один теоретический вопрос и 2 практических задания, соответствующие содержанию формируемых компетенций. Зачет с оценкой проводится в письменной форме. На ответ и решение задачи студенту отводится 60 минут. За ответ на теоретический вопрос студент может получить максимально 10 баллов, за выполнение практических заданий - 20 баллов.

По итогам прохождения дисциплины и с учетом шкал оценивания все набранные в результате текущей, рубежной и промежуточной аттестаций баллы суммируются и выставляется оценка .

Перевод баллов в оценку:

85 - 100 баллов – «отлично»

70 - 84 баллов – «хорошо»

60 - 69 баллов – «удовлетворительно»

менее 60 баллов – «неудовлетворительно»

Раздел 5. Методические указания для обучающегося по освоению дисциплины и выполнению контрольных заданий

5.1. Общие рекомендации к организации самостоятельной работы

Текущий контроль осуществляется в течение семестра в виде защиты практических работ.

Для обучающихся разработаны учебно-методические указания, которые содержат теоретические сведения по тестированию и оценке программного обеспечения, задания для практических работ и руководство к их выполнению. Обучающие материалы представлены в электронной папке преподавателя (локальная сеть кафедры Информационных и вычислительных технологий КРСУ). Выполнение всех практических работ завершается оформлением отчета, в

котором приводится теория по теме работы, отчеты в виде скриншотов результатов, таблиц и графиков.

Рубежный контроль осуществляется в виде сдачи контрольных работ. Вопросы по контрольным работам представлены в разделе 3 (блок А). Контрольные работы оформляются в письменном виде. Для допуска к экзамену студент в течение семестра должен сдать все практические и контрольные работы.

5.2. Подготовка к практическим занятиям

Перед посещением практического занятия необходимо проработать конспект лекций и методические указания по теме занятия. Оформление отчётов должно производиться по представленному образцу после окончания работы. Для подготовки к защите отчёта следует проанализировать экспериментальные результаты, обобщить результаты исследований в виде выводов, подготовить ответы на вопросы.