

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский ядерный университет «МИФИ»
Новоуральский технологический институт
филиал федерального государственного автономного образовательного учреждения высшего
образования «Национальный исследовательский ядерный университет «МИФИ»
(НТИ НИЯУ МИФИ)
Колледж НТИ

Цикловая методическая комиссия информационных технологий

ОДОБРЕНО

Учёным Советом НТИ НИЯУ МИФИ

Протокол № 5 от 02 сентября 2025 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПО МЕЖДИСЦИПЛИНАРНОМУ КУРСУ МДК.01.03
«ТЕСТИРОВАНИЕ ИНФОРМАЦИОННЫХ РЕСУРСОВ»
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ
ПМ.01 «ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА
ИНФОРМАЦИОННЫХ РЕСУРСОВ»

для студентов колледжа НТИ НИЯУ МИФИ,
обучающихся по программе среднего профессионального образования

специальность 09.02.09

«Веб-разработка»

очная форма обучения


на базе основного общего образования

квалификация

разработчик веб-приложений

Новоуральск 2025

ОДОБРЕНО:
на заседании
цикловой методической комиссии
информационных технологий
Протокол № 8 от 01.09.2025 г.
Председатель ЦМК ИТ


_____ И.И. Горницкая

Составлен в соответствии с рабочей программой междисциплинарного курса МДК.01.03 «Тестирование информационных ресурсов» профессионального модуля ПМ.01 «Проектирование и разработка информационных ресурсов» по специальности 09.02.09 Веб-разработка

Фонд оценочных средств по междисциплинарному курсу МДК.01.03 «Тестирование информационных ресурсов» профессионального модуля ПМ.01 «Проектирование и разработка информационных ресурсов» – Новоуральск: Изд-во колледжа НТИ НИЯУ МИФИ, 2025. – 47с.

АННОТАЦИЯ

Фонд оценочных средств предназначен для текущего контроля и промежуточной аттестации обучающихся по специальности 09.02.09 Веб-разработка на соответствие их персональных достижений поэтапным требованиям программы подготовки специалистов среднего звена по междисциплинарному курсу МДК 01.03 «Тестирование информационных ресурсов» профессионального модуля ПМ.01 «Проектирование и разработка информационных ресурсов». Комплектация фонда оценочных средств: паспорт, программа оценивания, оценочные средства текущего контроля и промежуточной аттестации, критерии оценивания. В паспорте фонда оценочных средств указаны: место междисциплинарного курса в структуре программы подготовки специалистов среднего звена, требования ФГОС СПО к результатам освоения междисциплинарного курса, перечень формируемых компетенций, компоненты фонда оценочных средств

Разработчик: Лебедева А.Н., преподаватель ЦМК информационных технологий

Редактор: Горницкая И.И.

СОДЕРЖАНИЕ

ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО МЕЖДИСЦИПЛИНАРНОМУ КУРСУ МДК.01.03 «ТЕСТИРОВАНИЕ ИНФОРМАЦИОННЫХ РЕСУРСОВ» ПРОФЕССИОНАЛЬНОГО МОДУЛЯ ПМ.01 «ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ РЕСУРСОВ»	4
ПРОГРАММА ОЦЕНИВАНИЯ.....	7
ОЦЕНОЧНЫЕ СРЕДСТВА И КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ.....	9
ОЦЕНОЧНЫЕ СРЕДСТВА И КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....	43

**ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ
ПО МЕЖДИСЦИПЛИНАРНОМУ КУРСУ МДК.01.03
«ТЕСТИРОВАНИЕ ИНФОРМАЦИОННЫХ РЕСУРСОВ»
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ
ПМ.01 «ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА
ИНФОРМАЦИОННЫХ РЕСУРСОВ»**

Фонд оценочных средств является составной частью учебно-методических документов, обеспечивающих реализацию основной образовательной программы СПО по специальности 09.02.09 Веб-разработка.

Фонд оценочных средств предназначен для проверки результатов освоения междисциплинарного курса МДК 01.03 «Тестирование информационных ресурсов» профессионального модуля ПМ.01 «Проектирование и разработка информационных ресурсов».

Междисциплинарный курс МДК 01.03 «Тестирование информационных ресурсов» является обязательной частью профессионального модуля ПМ.01 «Проектирование и разработка информационных ресурсов», который входит в профессиональный цикл и направлен освоение вида деятельности ВД.1 Проектирование и разработка информационных ресурсов.

Перечень формируемых компетенций в соответствии с требованиями ФГОС СПО:

Общие компетенции (ОК):

ОК.01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК.02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК.03 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в

профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;

ОК.04 Эффективно взаимодействовать и работать в коллективе и команде;

ОК.05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК.06 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК.07 Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК.08 Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК.09 Пользоваться профессиональной документацией на государственном и иностранном языках.

Профессиональные компетенции (ПК):

ПК 1.3 Интегрировать программный код в соответствующую инфраструктуру.

ПК 1.4 Использовать систему контроля версий в процессе коллективной (параллельной) разработки.

ПК 1.5 Выполнять процедуры тестирования программного кода.

В результате освоения междисциплинарного курса студент должен:

Владеть навыками:	тестирования информационного ресурса в соответствии с планом тестирования; документирования результатов тестирования; работы с системой контроля версий, в том числе при коллективной разработке;
Уметь:	выбирать и комбинировать техники тестирования информационных ресурсов; тестировать информационный ресурс с использованием тест-планов; применять инструменты подготовки тестовых данных; работать с инструментами подготовки тестовых данных; создавать отчет по результатам тестирования; создавать, клонировать, развивать репозитории хранения кода; создавать ветки репозитория и управления изменениями кода; решать конфликты версий кода;
Знать:	современные методики тестирования информационных ресурсов; принцип устройства систем хранения версий кода; интерфейсы управления системами хранения версий кода;

Фонд оценочных средств по междисциплинарному курсу МДК 01.03 «Тестирование информационных ресурсов» профессионального модуля ПМ.01 «Проектирование и разработка информационных ресурсов» включает оценочные средства для текущего контроля и оценочные средства для проведения промежуточной аттестации.

ПРОГРАММА ОЦЕНИВАНИЯ

№ п/п	Контролируемые элементы междисциплинарного курса	Контролируемые компетенции (или их части)	Вид оценивания
1	2	3	4
1	Тема 3.1 Тестирование готового программного кода	ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9 ПК 1.3, ПК 1.4, ПК 1.5	Устный опрос Письменный опрос (тестирование)
2	Тема 3.2 Современные методики тестирования информационных ресурсов	ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9 ПК 1.3, ПК 1.4, ПК 1.5	Устный опрос Письменный опрос (проверочная работа)
3	Тема 3.3 Юзабилити, UI testing	ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9 ПК 1.3, ПК 1.4, ПК 1.5	Устный опрос Письменный опрос (проверочная работа)
4	Тема 3.4 Тестирование безопасности	ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9 ПК 1.3, ПК 1.4, ПК 1.5	Устный опрос Письменный опрос (тестирование)
5	Тема 4.1 Работа с системой контроля версий	ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9 ПК 1.3, ПК 1.4,	Устный опрос Письменный опрос (тестирование)

№ п/п	Контролируемые элементы междисциплинарного курса	Контролируемые компетенции (или их части)	Вид оценивания
		ПК 1.5	
6	Тема 4.2 Принципы устройства систем хранения версий кода	ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9 ПК 1.3, ПК 1.4, ПК 1.5	Устный опрос Письменный опрос (проверочная работа)
7	Тема 4.3 Интерфейс управления системами хранения версий кода	ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9 ПК 1.3, ПК 1.4, ПК 1.5	Устный опрос Письменный опрос (проверочная работа)
8	Тема 4.4 Веб-интерфейсы контроля версий	ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9 ПК 1.3, ПК 1.4, ПК 1.5	Устный опрос Письменный опрос (тестирование)
9	Практические занятия	ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9 ПК 1.3, ПК 1.4, ПК 1.5	Представление решения профессиональных задач, защита отчета по практическому занятию
10	Промежуточная аттестация по междисциплинарному курсу	IV семестр промежуточная аттестация в форме комплексного дифференцированного зачета (зачет с оценкой)	

ОЦЕНОЧНЫЕ СРЕДСТВА И КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ

Текущий контроль проводится на теоретических и практических занятиях и включает в себя оценку знаний и умений, компетенций обучающихся.

Формы проведения текущего контроля: устный опрос, письменный опрос (может быть проведен в форме тестирования), выполнение практических работ на практических занятиях.

Примерный перечень вопросов для проведения устного опроса

Тема 3.1 Тестирование готового программного кода

- 1 Что такое тестирование ПО? Назовите 3-4 его основные цели.
- 2 Перечислите 3 уровня тестирования и кратко поясните, что проверяет каждый.
- 3 В чём разница между методами «чёрного ящика» и «белого ящика»? Приведите по 1 примеру техники для каждого.
- 4 Что такое тест-кейс? Назовите 4 обязательных элемента его структуры.
- 5 Что включает тест-план? Перечислите 3-4 ключевых раздела.
- 6 Что такое баг-репорт? Назовите 5 обязательных полей для описания дефекта.
- 7 Что такое регрессионное тестирование? Чем оно отличается от повторного тестирования?
- 8 Приведите 2 примера техник тестирования на основе требований. Как требования влияют на тестовую документацию?
- 9 Что такое нагрузочное тестирование? Для каких систем оно особенно важно и почему?
- 10 Назовите 2–3 метрики для оценки покрытия тестами. Почему это важно?

Тема 3.2 Современные методики тестирования информационных ресурсов

- 1 Пояснить что такое Agile-тестирование? В чём его ключевые отличия от классического подхода к тестированию?
- 2 Объясните суть методики «разработка через тестирование». Перечислите 3 преимущества и 1 ограничение этого подхода.
- 3 Разъясните, что такое разработка на основе поведения пользователя?
- 4 Расскажите, что подразумевается под непрерывным тестированием в CI/CD-пайплайне? Назовите 2 инструмента, поддерживающих эту практику.
- 5 Объясните суть методики «тестирование на основе рисков»? Как определить приоритетность тестов в этом подходе?
- 6 Опишите суть исследовательского тестирования. В каких ситуациях оно наиболее эффективно?
- 7 Что такое тест-дизайн? Перечислите 3 современные техники тест-дизайна и кратко поясните каждую.
- 8 Расскажите, что такое автоматизированное тестирование? Назовите 2 фреймворка для веб-тестирования. В каких случаях автоматизация нецелесообразна?
- 9 Опишите, процесс тестирования UX/UI? Приведите 3 примера проверок, специфичных для веб-интерфейсов.
- 10 Что такое A/B-тестирование? Как оно связано с тестированием ПО?

Тема 3.3 Юзабилити, UI testing

- 1 Поясните, что такое юзабилити? Перечислите 5 ключевых характеристик юзабилити согласно стандарту ISO 9241-11.
- 2 Разъясните, в чём разница между UI-тестированием и UX-тестированием? Приведите по 2 примера проверок для каждого вида.
- 3 Назовите 3 метода оценки юзабилити. Кратко опишите

процедуру проведения одного из них.

4 Уточните, что такое эвристики Нильсена? Перечислите 3–4 эвристики и приведите пример нарушения каждой.

5 Укажите, какие элементы интерфейса проверяют при UI-тестировании? Назовите 5–6 ключевых аспектов (например, шрифты, цвета, отступы).

6 Расскажите, что такое адаптивный дизайн в контексте юзабилити? Перечислите 3 проблемы, которые могут возникнуть при неправильной адаптации интерфейса.

7 Поясните, как тестируется доступность интерфейса? Назовите 2 стандарта и 3 конкретных проверки по ним.

8 Объясните, что такое пользовательские сценарии? Как они используются в юзабилити-тестировании? Приведите краткий пример сценария для веб-формы.

9 Перечислите инструменты, которые используются для UI-тестирования? Назовите 2–3 инструмента для визуального регрессионного тестирования и 1 для проверки контрастности.

10 Опишите процесс тестирования на целевой аудитории? Опишите 3 этапа такого тестирования и 1 метод сбора обратной связи.

Тема 3.4 Тестирование безопасности

1 Раскройте понятие «тестирование безопасности»? Назовите 3 ключевые цели тестирования безопасности.

2 Перечислите 4 распространённые уязвимости веб-приложений и кратко поясните каждую.

3 Разъясните, что такое SQL-инъекция? Приведите простой пример вредоносного ввода и объясните, как она может быть предотвращена.

4 Объясните, в чём опасность межсайтового скриптинга (XSS)? Назовите 2 типа XSS и 2 способа защиты от них.

5 Расскажите, что такое «межсайтовая подделка запроса»? Как можно защититься от CSRF-атак? Приведите 1 пример реализации защиты.

6 Укажите, что включает тестирование аутентификации и авторизации? Назовите 3 проверки, которые следует выполнить для формы входа.

7 Объясните, что такое сканирование уязвимостей? Назовите 2 инструмента для автоматизированного сканирования и кратко опишите их назначение.

8 Расскажите, что такое пентест? Чем он отличается от сканирования уязвимостей? Перечислите 2 этапа пентеста.

9 Опишите, как тестируется защита передачи данных? Назовите 2 проверки для оценки настройки SSL/TLS.

10 Дайте определения понятиям «логи», «аудит безопасности»? Приведите 3 примера событий, которые должны фиксироваться в логах для анализа инцидентов.

Тема 4.1 Работа с системой контроля версий

1. Разьясните, что такое система контроля версий? Назовите 2 основные задачи, которые она решает в командной разработке.

2. Поясните ключевое отличие централизованных СКВ от распределённых? Перечислите по 1-2 преимущества каждого типа.

3. Дайте определение понятию «репозиторий»? Опишите разницу между локальным и удалённым репозиторием.

4. Объясните смысл операций commit, push и pull в Git. В какой последовательности они обычно выполняются?

5. Объясните, что такое ветка в системе контроля версий? Для чего создают отдельные ветки?

6. Расскажите, что происходит при конфликте слияния? Опишите 3 шага для его разрешения в Git.

7. Поясните, для чего используют команду git stash? Приведите пример ситуации, когда она полезна.

8. Поясните назначение gitignore-файла? Приведите 3 типа файлов/папок, которые обычно добавляют в .gitignore для веб-проекта.

9. Расскажите, как проверить историю изменений файла в Git? Назовите 2 команды для просмотра лога и кратко поясните их отличия.

10. Поясните, что такое pull request / merge request? Опишите его роль в командном workflow.

Тема 4.2 Принципы устройства систем хранения версий кода

1. Расскажите, что такое система контроля версий (СКВ)? Перечислите 3 ключевые функции, которые она обеспечивает в процессе разработки ПО.

2. Опишите принципиальное различие между централизованными и распределёнными системами контроля версий. Приведите по 1 примеру каждой.

3. Что представляет собой репозиторий в СКВ? Объясните разницу между локальным и удалённым репозиторием, для каждого укажите по 2 особенности.

4. Поясните, что такое коммит? Какие метаданные обычно сохраняются вместе с коммитом? Назовите не менее 3.

5. Объясните концепцию ветвления в СКВ. Для чего создаются отдельные ветки? Приведите 2 примера типовых веток в Git-проекте.

6. Расскажите, что происходит при слиянии веток? Опишите 2 возможных сценария слияния и укажите, в каком случае возникает конфликт.

7. Поясните назначение индекса в Git? Какова его роль в процессе подготовки коммита? Опишите последовательность действий от изменения файла до коммита.

8. Объясните, что такое хеш-идентификатор коммита? Почему в Git используются хеши вместо числовых номеров? Назовите 2 преимущества такого подхода.

9. Расскажите, как СКВ хранит изменения между версиями файлов? Объясните принцип дельта-кодирования и его преимущества перед полным копированием файлов.

10. Объясните, что такое удалённый репозиторий? Опишите 2

основных сценария взаимодействия с ним и объясните, как это обеспечивает командную работу.

Тема 4.3 Интерфейс управления системами хранения версий кода

1. Перечислите 3–4 основных интерфейса взаимодействия с системами контроля версий и кратко поясните сферу применения каждого.
2. Назовите 5 базовых команд Git в командной строке (CLI) и укажите их назначение.
3. Как в Git через CLI создать новую ветку и переключиться на неё? Приведите точную команду и объясните её синтаксис.
4. Опишите последовательность действий в CLI для добавления изменённых файлов в индекс и создания коммита. Какие команды и в каком порядке нужно выполнить?
5. Поясните, как через CLI отправить локальные изменения в удалённый репозиторий? Назовите команду и 2 возможных флага-модификатора.
6. Назовите известные вам графические интерфейсы (GUI) для работы с Git. Перечислите 2–3 программы и укажите по 1 их ключевой особенности.
7. Укажите преимущество использования GUI-клиентов для СКВ по сравнению с CLI? Назовите 2 аргумента и 1 ситуацию, где CLI остаётся предпочтительнее.
8. Расскажите, как в GUI-клиенте выполнить слияние веток и разрешить конфликт? Опишите 3 шага процесса.
9. Расскажите, что такое веб-интерфейс СКВ (на примере GitHub/GitLab). Перечислите 3 действия, которые удобно выполнять именно через веб-интерфейс, а не через CLI/GUI.
10. Поясните, как через веб-интерфейс создать pull request? Опишите 4 ключевых этапа его оформления и укажите, какую информацию важно заполнить.

Тема 4.4 Веб-интерфейсы систем контроля версий

1. Поясните, что такое веб-интерфейс системы контроля версий? Приведите 2 примера популярных платформ и укажите их ключевое отличие.
2. Перечислите 4-5 основных функций, доступных пользователю через веб-интерфейс СКВ (на примере GitHub/GitLab).
3. Опишите процесс создания нового репозитория через веб-интерфейс. Опишите 3-4 ключевых шага и укажите, какие параметры обычно можно настроить при создании.
4. Расскажите, что такое pull request в веб-интерфейсе СКВ? Опишите 4 этапа работы с PR от создания до слияния.
5. Опишите, как в веб-интерфейсе отследить историю изменений файла? Назовите 2 способа и укажите, какую информацию можно получить из истории коммитов.
6. Поясните, что такое ветки в контексте веб-интерфейса СКВ? Расскажите, как через интерфейс создать новую ветку и переключиться на неё?
7. Поясните процесс разрешения конфликта при слиянии веток через веб-интерфейс? Опишите 3 шага процесса и укажите, какие инструменты интерфейса помогают в этом.
8. Расскажите, что такое Issues в веб-интерфейсе СКВ? Перечислите 3 сценария использования Issues в командной разработке.
9. Расскажите, как настроить права доступа к репозиторию в веб-интерфейсе? Назовите 2 роли участников и опишите их разрешения.
10. Объясните, что такое Actions/CI в веб-интерфейсах СКВ? Приведите 2 примера автоматизации, которую можно настроить через этот инструмент.

Критерии оценивания ответов на вопросы для проведения устного опроса

«ОТЛИЧНО». Обучающийся демонстрирует глубокое и прочное усвоение материала темы. Дан полный, развернутый ответ на вопрос,

показана совокупность осознанных знаний по междисциплинарному курсу, доказательно раскрыты основные положения; в ответе прослеживается четкая структура, логическая последовательность, отражающая сущность раскрываемых понятий, теорий, явлений. Знание по междисциплинарному курсу демонстрируется на фоне понимания его в системе данной науки и междисциплинарных связей. Ответ изложен литературным языком с использованием методической терминологии. Могут быть допущены недочеты в определении понятий, исправленные студентом самостоятельно в процессе ответа.

«ХОРОШО». Обучающийся демонстрирует знания в объеме пройденной программы. Дан полный, развернутый ответ на поставленный вопрос, показано умение выделить существенные и несущественные признаки, причинно-следственные связи. Ответ четко структурирован, логичен, изложен литературным языком. Могут быть допущены 2-3 неточности или незначительные ошибки, исправленные студентом с помощью дополнительных вопросов и уточняющих вопросов преподавателя.

«УДОВЛЕТВОРИТЕЛЬНО». Обучающийся демонстрирует недостаточно полное усвоение знаний по пройденной программе. Дан недостаточно полный и недостаточно развернутый ответ. Логика и последовательность изложения имеют нарушения. Допущены ошибки в раскрытии понятий, употреблении терминов. Студент не способен самостоятельно выделить существенные и несущественные признаки и причинно-следственные связи. Умение раскрыть значение обобщенных знаний не показано. Речевое оформление требует поправок, коррекции.

«НЕУДОВЛЕТВОРИТЕЛЬНО». Обучающийся демонстрирует непонимание основного содержания учебного материала. Ответ представляет собой разрозненные знания с существенными ошибками по вопросу. Присутствуют фрагментарность, нелогичность изложения. Отсутствуют конкретизация и доказательность изложения. Речь неграмотная.

Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента.

Перечень заданий для проведения письменного опроса

Проверочная работа

Раздел 3 Тестирование информационных ресурсов и интеграция программного кода

Форма проведения письменного опроса – проверочная работа по вариантам.

Проверяемые понятия:

- методологии разработки и тестирования, подходы к проектированию тестов, техники тест-дизайна, инструментарий;
- основы юзабилити, методы оценки, доступность, практические артефакты.

Тема 3.2 Современные методики тестирования информационных ресурсов

Вариант 1

1. Перечислите и кратко опишите 3 современные методики тестирования. Укажите, в чём их ключевое отличие от классического waterfall-подхода.
2. Составьте краткий сценарий BDD-теста для функции «Добавление товара в корзину» в интернет-магазине.

Вариант 2

1. Что такое непрерывное тестирование в CI/CD? Назовите 2 инструмента, поддерживающих эту практику, и поясните их роль.
2. Опишите 3 тестовых сценария для проверки API сервиса заказа такси. Укажите входные данные и ожидаемые ответы.

Вариант 3

1. Объясните суть Risk-Based Testing. Как определить приоритетность тестов? Приведите 2 примера критериев приоритизации.
2. Для формы регистрации пользователя предложите 2 теста на основе

техники «эквивалентное разбиение» и 1 тест на «граничные значения». Опишите входные данные и ожидаемый результат.

Тема 3.3 Юзабилити, UI testing

Вариант 1

1. Перечислите 5 ключевых характеристик юзабилити по ISO 9241-11. Кратко поясните каждую.

2. Проведите эвристическую оценку (по Нильсену) главной страницы новостного сайта (описание/скриншот предоставляется). Выпишите 2-3 нарушения и предложите исправления.

Вариант 2

1. В чём разница между UI- и UX-тестированием? Приведите по 2 примера проверок для каждого.

2. Составьте чек-лист из 5 пунктов для UI-тестирования мобильной версии формы оплаты. Включите проверки шрифтов, цветов, отступов, интерактивных элементов.

Вариант 3

1. Что такое доступность (accessibility) интерфейса? Назовите 2 стандарта (например, WCAG) и 3 конкретных критерия проверки.

2. Разработайте краткий пользовательский сценарий (user flow) для задачи «Найти и забронировать отель» в туристическом приложении. Укажите 4-5 шагов и возможные точки затруднения.

Раздел 4 Работа с системой контроля версий

Форма проведения письменного опроса – проверочная работа по вариантам.

Проверяемые понятия:

- типы СКВ, базовые сущности, механизмы хранения, процессы;
- интерфейсы взаимодействия, функционал веб-интерфейсов, рабочие процессы, интеграция и автоматизация.

Тема 4.2 Принципы устройства систем хранения версий кода

Вариант 1

1. Объясните разницу между централизованными и распределёнными СКВ. Приведите по 1 примеру и укажите 2 преимущества каждого типа.
2. Нарисуйте схему взаимодействия локального и удалённого репозитория при выполнении команд: `commit`, `push`, `pull`. Подпишите, какие данные передаются на каждом этапе.

Вариант 2

1. Что такое коммит? Какие метаданные сохраняются в коммите? Перечислите 4-5 полей.
2. Опишите последовательность действий для создания ветки `feature/login` в Git, внесения изменений, коммита и отправки в удалённый репозиторий. Приведите команды CLI.

Вариант 3

1. Как СКВ хранит изменения между версиями файлов? Объясните принцип дельта-кодирования и его выгоду.
2. Представьте, что при слиянии веток возник конфликт. Опишите 3 шага его разрешения в Git (через CLI или GUI). Укажите, какие файлы/строки будут помечены.

Тема 4.3 Интерфейс управления системами хранения версий кода

Вариант 1

1. Перечислите 3 интерфейса взаимодействия с СКВ. Для каждого укажите 1 преимущество и 1 ограничение.
2. Через веб-интерфейс GitHub создайте pull request (PR) из ветки `feature/search` в `main`. Опишите 4 шага оформления PR (заголовок, описание, рецензенты, метки).

Вариант 2

1. Какие GUI-клиенты для Git вы знаете? Назовите 2 программы и 2 их ключевые функции.
2. В GUI-клиенте разрешите конфликт при слиянии веток. Опишите 3

действия: как найти конфликт, как выбрать версии строк, как зафиксировать разрешение.

Вариант 3

1. Что такое Issues в веб-интерфейсе СКВ? Перечислите 3 сценария их использования в команде.

2. Через веб-интерфейс GitLab настройте права доступа к репозиторию: роли Developer и Maintainer. Укажите по 2 разрешения для каждой роли (например, «создавать ветки», «сливать PR»).

Критерии оценки результатов выполнения проверочных работ

«ОТЛИЧНО». Обучающимся подготовлен полный, логичный ответ, все пункты раскрыты, примеры корректны и обоснованы;

«ХОРОШО». Обучающимся подготовлен в целом полный ответ, но есть 1–2 недочёта (нехватка примера, неполное пояснение);

«УДОВЛЕТВОРИТЕЛЬНО». Обучающимся подготовлен частично неполный ответ (пропущены 1–2 пункта), примеры поверхностны или не всегда корректны.

«НЕУДОВЛЕТВОРИТЕЛЬНО». Обучающимся подготовлен фрагментарный ответ, большинство пунктов не раскрыты, примеры отсутствуют или ошибочны.

Тестирование

Форма проведения письменного опроса – тестирование.

Проверяемые понятия:

– уровни тестирования, типы тестирования, методологии и подходы, артефакты и документация;

– основные уязвимости и атаки, методы и инструменты, механизмы защиты, стандарты и ресурсы;

– базовые сущности, основные команды git, процессы и сценарии, типы СКВ

Раздел 3 Тестирование информационных ресурсов и интеграция программного кода

Тест №1 Тестирование готового программного кода

1. Что является основной целью тестирования ПО?
 - а) Исправить все ошибки в коде.
 - б) Доказать, что программа работает без ошибок.
 - в) Выявить дефекты и снизить риск отказов в эксплуатации.
 - г) Заменить этап разработки.
2. Какой уровень тестирования проверяет отдельные модули/компоненты программы?
 - а) Системное тестирование.
 - б) Модульное тестирование.
 - в) Приёмочное тестирование.
 - г) Нагрузочное тестирование.
3. Что такое тест-кейс?
 - а) Программа для автоматического запуска тестов.
 - б) Документ с шагами тестирования и ожидаемым результатом.
 - в) Список всех найденных багов.
 - г) Отчёт о покрытии кода тестами.
4. Что проверяет интеграционное тестирование?
 - а) Работу отдельных функций.
 - б) Взаимодействие между модулями/компонентами.
 - в) Производительность системы.
 - г) Пользовательский интерфейс.
5. Что означает термин «регрессионное тестирование»?
 - а) Тестирование новой функциональности.
 - б) Повторное тестирование после исправлений, чтобы убедиться, что старые функции не сломались.
 - в) Тестирование на старых версиях ПО.

- г) Тестирование без тестовой документации.
6. Какой метод тестирования предполагает знание внутренней структуры кода?
- а) «Чёрный ящик».
 - б) «Белый ящик».
 - в) Исследовательское тестирование.
 - г) Приёмочное тестирование.
7. Что такое покрытие тестами?
- а) Количество найденных багов.
 - б) Процент протестированных требований/кода.
 - в) Время, затраченное на тестирование.
 - г) Число выполненных тест-кейсов.
8. Что включает тест-план?
- а) Только список тест-кейсов.
 - б) Стратегию, объём, ресурсы, расписание тестирования.
 - в) Отчёт о найденных багах.
 - г) Код автоматических тестов.
9. Что такое баг-репорт?
- а) Документ с описанием найденного дефекта и шагами для его воспроизведения.
 - б) Отчёт о покрытии тестами.
 - в) План тестирования.
 - г) Список требований к ПО.
10. Какой тип тестирования проверяет, соответствует ли ПО требованиям заказчика?
- а) Модульное тестирование.
 - б) Приёмочное тестирование (UAT).
 - в) Нагрузочное тестирование.
 - г) Стрессовое тестирование.

Тест №2 Тестирование безопасности

1. Что такое SQL-инъекция?
 - а) Атака через вредоносный JavaScript-код.
 - б) Внедрение вредоносного SQL-кода в запросы к БД.
 - в) Перехват сетевого трафика.
 - г) Взлом пароля методом подбора.
2. Что защищает от XSS (межсайтового скриптинга)?
 - а) Шифрование трафика (HTTPS).
 - б) Валидация и санитизация ввода, Content Security Policy (CSP).
 - в) Сильные пароли.
 - г) Файрвол.
3. Что такое CSRF (Cross-Site Request Forgery)?
 - а) Внедрение SQL-кода.
 - б) Подделка запросов от имени авторизованного пользователя.
 - в) Перехват сессии.
 - г) Атака «человек посередине».
4. Какой инструмент используется для сканирования уязвимостей?
 - а) Jira.
 - б) Nmap или Burp Suite.
 - в) Git.
 - г) Selenium.
5. Что проверяет тестирование аутентификации?
 - а) Скорость работы системы.
 - б) Корректность входа/выхода, восстановление пароля, права доступа.
 - в) Визуальное оформление форм.
 - г) Совместимость с браузерами.
6. Что такое пентест (penetration testing)?
 - а) Автоматизированное сканирование уязвимостей.
 - б) Имитация атаки злоумышленника для оценки защищённости.
 - в) Проверка кода статическим анализатором.

- г) Мониторинг логов.
7. Как защитить передачу данных в веб-приложении?
- а) Использовать HTTP.
 - б) Использовать HTTPS и TLS.
 - в) Хранить пароли в открытом виде.
 - г) Отключить аутентификацию.
8. Что такое OWASP Top 10?
- а) Список лучших практик кодирования.
 - б) Перечень наиболее критических уязвимостей веб-приложений.
 - в) Стандарт шифрования.
 - г) Инструмент тестирования.
9. Что такое логи безопасности?
- а) Файлы с записями событий (вход, ошибки, запросы) для анализа инцидентов.
 - б) Список пользователей.
 - в) Код приложения.
 - г) Настройки сервера.
10. Что такое двухфакторная аутентификация (2FA)?
- а) Ввод только пароля.
 - б) Пароль + дополнительный фактор (SMS, токен, биометрия).
 - в) Вход по отпечатку пальца.
 - г) Использование одного токена.

Раздел 4 Работа с системой контроля версий

Тест №3 Работа с системой контроля версий

1. Что такое репозиторий в системе контроля версий?
- а) Программа для написания кода.
 - б) Хранилище файлов проекта с историей изменений.
 - в) Сервер для размещения готового ПО.
 - г) Инструмент для тестирования кода.

2. Какая команда инициализирует новый локальный репозиторий Git?
 - а) git start
 - б) git new
 - в) git init
 - г) git create
3. Для чего используется команда git clone?
 - а) Для создания новой ветки.
 - б) Для копирования удалённого репозитория на локальный компьютер.
 - в) Для отправки изменений в удалённый репозиторий.
 - г) Для просмотра истории коммитов.
4. Что происходит при выполнении команды git push?
 - а) Загружаются изменения из удалённого репозитория.
 - б) Сохраняются изменения в локальном репозитории.
 - в) Отправляются локальные коммиты в удалённый репозиторий.
 - г) Создаётся новая ветка.
5. Что хранит файл .gitignore?
 - а) Список всех коммитов.
 - б) Настройки пользователя Git.
 - в) Список файлов и папок, которые Git должен игнорировать.
 - г) Историю слияния веток.
6. Как посмотреть историю коммитов в Git?
 - а) git status
 - б) git log
 - в) git show
 - г) git diff
7. Что такое ветка (branch) в Git?
 - а) Копия всего репозитория.
 - б) Независимая линия разработки внутри репозитория.
 - в) Файл с настройками проекта.
 - г) Временная папка для черновиков.

8. Какая команда добавляет файлы в индекс (staging area)?
- а) git commit
 - б) git add
 - в) git push
 - г) git merge
9. Что происходит при конфликте слияния (merge conflict)?
- а) Git автоматически разрешает конфликт.
 - б) Изменения отменяются без сохранения.
 - в) Git останавливает слияние и требует ручного разрешения.
 - г) Создаётся резервная копия репозитория.
10. Какая команда фиксирует изменения в локальном репозитории с комментарием?
- а) git save
 - б) git update
 - в) git commit
 - г) git record

Тест №4 Веб-интерфейсы систем контроля версий

1. Что такое веб-интерфейс системы контроля версий?
- а) Программа для локального редактирования кода.
 - б) Веб-приложение, позволяющее взаимодействовать с репозиторием через браузер.
 - в) Утилита командной строки для работы с Git.
 - г) Специальный формат хранения файлов в репозитории.
2. Какой из перечисленных сервисов НЕ является веб-интерфейсом для систем контроля версий?
- а) GitHub.
 - б) GitLab.
 - в) Bitbucket.
 - г) Git Bash.

3. Какая основная функция веб-интерфейса СКВ (системы контроля версий)?
 - а) Автоматическое написание кода.
 - б) Визуализация репозитория, управление ветками, просмотр истории изменений.
 - в) Компиляция исходного кода в исполняемый файл.
 - г) Тестирование производительности приложения.
4. Что обычно можно сделать через веб-интерфейс СКВ?
 - а) Только просматривать файлы репозитория.
 - б) Только скачивать архивы проекта.
 - в) Создавать ветки, делать коммиты, просматривать diff, управлять правами доступа.
 - г) Запускать серверные скрипты.
5. Для чего используется Pull Request (PR) в веб-интерфейсах СКВ?
 - а) Для удаления файлов из репозитория.
 - б) Для предложения изменений в чужой репозиторий и их обсуждения перед слиянием.
 - в) Для автоматического развёртывания кода на сервер.
 - г) Для создания резервных копий репозитория.
6. Какой сервис предоставляет веб-интерфейс для работы не только с Git, но и с Mercurial?
 - а) GitHub.
 - б) GitLab.
 - в) Bitbucket.
 - г) SourceForge.
7. Что такое «репозиторий» в контексте СКВ?
 - а) Специальный текстовый редактор для кода.
 - б) Хранилище файлов проекта с историей изменений.
 - в) Сервер для хостинга веб-сайтов.
 - г) Инструмент для отладки программ.
8. Какая функция обычно недоступна в базовом веб-интерфейсе СКВ?

- а) Просмотр истории коммитов.
 - б) Создание новых веток.
 - в) Выполнение сложных операций слияния через командную строку (например, git rebase).
 - г) Добавление комментариев к файлам.
9. Что означает статус «Merge conflict» в веб-интерфейсе СКВ?
- а) Успешное слияние веток.
 - б) Ошибка аутентификации пользователя.
 - в) Конфликт изменений в файлах при попытке слияния веток.
 - г) Отсутствие прав на запись в репозиторий.
10. Какой из этих элементов обычно отображается на главной странице репозитория в веб-интерфейсе?
- а) Список всех пользователей системы.
 - б) Дерево файлов, описание проекта, последние коммиты.
 - в) Статистика использования оперативной памяти сервера.
 - г) Журнал системных ошибок сервера.

Ключ к тестам

Тест 1

1. в 2. б 3. б 4. б 5. б 6. б 7. б 8. б 9. а 10. б

Тест 2

1. б 2. б 3. б 4. б 5. б 6. б 7. б 8. б 9. а 10. б

Тест 3

1. б 2. в 3. б 4. в 5. в 6. б 7. б 8. б 9. в 10. в

Тест 4

1. б 2. г 3. б 4. в 5. б 6. в 7. б 8. в 9. в 10. б

Критерии оценки результатов выполнения тестовых заданий

В качестве критерия освоения учебного модуля берётся выполнение 50% заданий от их общего числа в тест-задании.

Для интерпретации результатов тестирования используется переводная шкала, с помощью которой результаты теста преобразуют в оценку знаний

студентов по пятибалльной системе. Переводная шкала представлена в Таблице 1.

Таблица 1 – Переводная шкала для интерпретации результатов выполнения тест-задания

% ошибочных ответов	Интерпретация результатов тестирования по 5-ой системе
до 10%	5
от 15% до 30%	4
от 35% до 50%	3
более 50%	2

Практические занятия

Учебным планом предусмотрено проведение практических занятий:

Раздел 3 Тестирование информационных ресурсов и интеграция программного кода

Практическое занятие №1 «Функциональное тестирование»

Практическое занятие №2 «Интеграционное тестирование»

Практическое занятие №3 «Тестирование локализации и глобализации»

Практическое занятие №4 «Кросс-платформенное тестирование»

Практическое занятие №5 «Тестирование удобства использования»

Практическое занятие №6 «Тестирование безопасности»

Практическое занятие №7 «Выбрать технику тестирования и протестировать готовый веб-ресурс с использованием тест-планов. Создать отчет с результатами тестирования»

Раздел 4 Работа с системой контроля версий

Практическое занятие №1 «Зарегистрировать и создать репозиторий»

Практическое занятие №2 «Создать ветки репозитория»

Практическое занятие №3 «Копировать свой и чужой проект в свой репозиторий»

Практическое занятие №4 «Коллективная разработка»

В ходе практических занятий по Разделу 3 формируются навыки и

умения системного подхода к обеспечению качества ПО, работы с тестовой документацией, использования инструментов тестирования, анализа требований и перевода их в тестовые сценарии, коммуникации с разработчиками и заказчиками по вопросам качества.

В результате практических занятий по Разделу 4 отрабатываются навыки выполнения базовых операций Git, навыки командной работы, понимание workflow, использования платформ GitHub, GitLab и аналогов для хостинга и совместной работы, ведения репозитория.

Оценивание выполнения практических работ – защита отчета по практическому занятию.

Чек-лист оценивания выполненной работы по практическому занятию:

1 Выслать ссылку на облачное хранилище, в котором размещен файл с выполненным заданием по теме практического занятия

2 Выслать ссылку на облачное хранилище, в котором размещен отчет по практическому занятию

- структура отчета: Титульный лист, Цель работы, Оборудование, Программное обеспечение, Текст отчета (ответы на контрольные вопросы для отчета), Вывод

- текст отчета оформить в текстовом процессоре MS Word

- формат отчета: pdf

3 Подготовиться к устному собеседованию с преподавателем по теме практического занятия

4 Защитить отчет по практическому занятию в режиме взаимодействия с преподавателем

Отработка умений осуществляется по индивидуальным заданиям.

Примерный перечень тем индивидуальных заданий для проведения практических занятий

Раздел 3 Тестирование информационных ресурсов и интеграция программного кода

Практическое занятие № 1 «Функциональное тестирование»

Вариант 1

Провести функциональное тестирование простой формы регистрации на веб-сайте (поля: имя, email, пароль).

- Составить чек-лист из 8-10 пунктов, покрывающий позитивные и негативные сценарии.
- Выполнить тесты, зафиксировать результаты.
- Оформить отчёт: список пройденных тестов, найденные дефекты (если есть), скриншоты.

Вариант 2

Протестировать функционал корзины интернет-магазина (добавление/удаление товаров, расчёт итоговой суммы).

- Разработать 5–7 тест-кейсов с условиями, шагами и ожидаемыми результатами.
- Проверить граничные условия (добавление 0 товаров, отрицательное количество).
- Заполнить таблицу результатов тестирования, указать статус каждого кейса.

Вариант 3

Выполнить функциональное тестирование API метода GET /api/users (возвращает список пользователей).

- Сформировать тестовые данные (2-3 набора корректных и некорректных параметров запроса).
- Проверить коды ответов, структуру JSON, соответствие данных БД.
- Подготовить краткий отчёт: перечень тестов, фактические результаты, выводы о работоспособности метода.

Практическое занятие № 2 «Интеграционное тестирование»

Вариант 1

Проверить интеграцию формы обратной связи с SMTP-сервером.

- Отправить сообщение через форму, убедиться в получении письма

на указанный email.

- Протестировать сценарии: пустой текст, превышенный объём, некорректный email.

- Задokumentировать цепочку взаимодействия компонентов и найденные проблемы.

Вариант 2

Исследовать интеграцию платёжного шлюза с интернет-магазином.

- Провести тестовые платежи (успешный, отклонённый, тайм-аут).

- Убедиться, что статус заказа в БД обновляется корректно.

- Составить схему потоков данных между системами и отметить точки проверки.

Вариант 3

Протестировать взаимодействие мобильного приложения с бэкендом через REST API.

- Проверить синхронизацию данных при создании/изменении записи в приложении.

- Имитировать потерю сети, оценить поведение приложения.

- Оформить отчёт: описание интеграционных сценариев, логи запросов/ответов, выводы.

Практическое занятие № 3 «Тестирование локализации и глобализации»

Вариант 1

Проверить локализацию веб-сайта на английский и испанский языки.

- Убедиться в корректности перевода заголовков, кнопок, сообщений.

- Проверить отображение дат и времени в формате US/ES.

- Зафиксировать несоответствия в таблице (оригинал – перевод – комментарий).

Вариант 2

Протестировать глобализацию приложения для рынков ЕС и Японии.

- Проверить форматы чисел (разделители тысяч/дробной части), валют

(€, ¥).

- Оценить отображение длинных текстовых строк в интерфейсе.
- Подготовить список рекомендаций по улучшению локализации.

Вариант 3

Исследовать поведение приложения при смене локали во время сессии.

- Переключиться с en-US на fr-CA, проверить сохранение контекста.
- Протестировать ввод данных на разных языках (кириллица, иероглифы).
- Описать сценарии, где локализация может нарушить функциональность.

Практическое занятие № 4 «Кросс-платформенное тестирование»

Вариант 1

Проверить отображение веб-страницы на 3 браузерах (Chrome, Firefox, Safari) и 2 разрешениях экрана.

- Зафиксировать различия в вёрстке, шрифтах, интерактивных элементах.
- Сделать скриншоты для каждого случая.
- Составить таблицу совместимости с оценками «Да/Нет/Частично».

Вариант 2

Протестировать мобильное приложение на Android (2 версии ОС) и iOS (2 версии ОС).

- Проверить работу основных функций (вход, просмотр контента, отправка данных).
- Отметить различия в UX/UI между платформами.
- Подготовить отчёт: перечень устройств, найденные проблемы, приоритеты исправлений.

Вариант 3

Исследовать кросс-платформенную совместимость десктопного ПО (Windows, macOS, Linux).

– Установить приложение на каждой ОС, зафиксировать ошибки установки.

– Проверить ключевые сценарии работы (сохранение файлов, печать, интеграция с ОС).

– Описать рекомендации по унификации поведения на разных платформах.

Практическое занятие № 5 «Тестирование удобства использования»

Вариант 1

Оценить юзабилити формы поиска на сайте.

– Провести тестирование «пяти секунд»: что пользователь замечает первым?

– Замерить время выполнения задачи (найти товар по запросу).

– Составить список эргономических недочётов и предложений по улучшению.

Вариант 2

Провести экспертную оценку интерфейса мобильного приложения (5-7 критериев).

– Использовать эвристики юзабилити (видимость статуса, согласованность, помощь в ошибках).

– Для каждого нарушения привести обоснование и скриншот.

– Предложить альтернативные решения.

Вариант 3

Организовать мини-юзабилити-тест с 2–3 участниками.

– Подготовить сценарий (3-4 задачи, например, «оформить заказ»).

– Записать наблюдения и комментарии тестируемых.

– Сформировать отчёт: ключевые проблемы, процент успешности задач, рекомендации.

Практическое занятие № 6 «Тестирование безопасности»

Вариант 1

Проверить форму входа на уязвимости.

- Попробовать SQL-инъекцию в поле логина/пароля.
- Тестировать брутфорс (5 попыток неверного пароля).
- Зафиксировать ответы сервера и возможные утечки информации.

Вариант 2

Исследовать защиту API от несанкционированного доступа.

- Отправить запрос без токена авторизации, оценить ответ.
- Попробовать подменить ID пользователя в URL.
- Описать найденные уязвимости и способы их эксплуатации.

Вариант 3

Провести сканирование веб-приложения инструментом OWASP ZAP.

- Настроить автоматическое тестирование на типичные уязвимости.
- Проанализировать отчёт: выделить критические риски.
- Подготовить рекомендации по устранению выявленных проблем.

Практическое занятие № 7 «Выбрать технику тестирования и протестировать готовый веб-ресурс с использованием тест-планов»

Вариант 1

Выбрать технику (например, граничные значения) и протестировать калькулятор на сайте.

- Составить тест-план: цели, объекты, методы, ресурсы.
- Выполнить 8-10 тестов, задокументировать результаты.
- Подготовить отчёт: сводка по дефектам, уровень покрытия, выводы.

Вариант 2

Применить технику «таблиц решений» для тестирования системы скидок.

- Описать правила скидок (например, «если сумма > 1000 руб., то скидка 5%»).
- Построить таблицу с комбинациями условий и ожидаемых результатов.

- Провести тесты, оформить отчёт с анализом покрытия.

Вариант 3

Комплексно протестировать блог-платформу (публикации, комментарии, поиск).

- Выбрать 2–3 техники тестирования (например, эквивалентные классы + сценарное тестирование).
- Разработать тест-кейсы, выполнить их.
- Подготовить итоговый отчёт: метрики качества, приоритетные дефекты, рекомендации по релизу.

Раздел 4 Работа с системой контроля версий

Практическое занятие № 1 «Зарегистрировать и создать репозиторий»

Вариант 1

- Зарегистрируйтесь на GitHub (если учётной записи нет).
- Создайте публичный репозиторий с названием my-first-project.
- Добавьте файл README.md с кратким описанием проекта (2-3 предложения).
- Создайте файл .gitignore для проекта на Python (или другом языке по вашему выбору).
- Сделайте первый коммит с сообщением «Initial commit».
- Отправьте изменения на сервер (git push).
- Предоставьте ссылку на репозиторий и скриншот его главной страницы.

Вариант 2

- Зарегистрируйтесь на GitLab (если учётной записи нет).
- Создайте приватный репозиторий test-repo.
- Загрузите в репозиторий 2 файла: index.html (простая HTML-страница), style.css (базовые стили).
- Сделайте коммит с сообщением «Add basic files».
- Отправьте изменения на сервер.

- Проверьте, что файлы отображаются в репозитории.
- Сделайте скриншот страницы репозитория с загруженными файлами.

Вариант 3

- Зарегистрируйтесь на GitHub.
- Создайте репозиторий learning-git с опцией «Добавить README.md» и «Добавить .gitignore» (выберите шаблон для JavaScript).
- Откройте README.md в онлайн-редакторе GitHub, добавьте заголовок и пару абзацев текста.
- Сохраните изменения (создаст коммит автоматически).
- Скачайте репозиторий на компьютер через кнопку «Code» → «Download ZIP».
- Распакуйте архив и убедитесь, что файлы на месте.
- Отправьте ссылку на репозиторий и скриншот скачанной папки.

Практическое занятие № 2 «Создать ветки репозитория»

Вариант 1

- В существующем репозитории создайте ветку feature/header.
- В новой ветке добавьте файл header.html с простым HTML-кодом заголовка.
- Сделайте коммит с сообщением «Add header component».
- Отправьте ветку на сервер (git push -u origin feature/header).
- Переключитесь на ветку main и убедитесь, что файла header.html там нет.
- Сделайте скриншот списка веток на GitHub/GitLab.

Вариант 2

- Создайте ветку bugfix/title-color.
- В ветке измените цвет заголовка в файле style.css (например, color: blue;).
- Сделайте коммит с сообщением «Fix title color».
- Отправьте ветку на сервер.

– В интерфейсе GitHub/GitLab создайте Pull Request (PR) из `bugfix/title-color` в `main`.

- Опишите в PR, что исправлено.
- Сделайте скриншот страницы PR.

Вариант 3

- Создайте ветку `feature/footer`.
- Добавьте в неё файл `footer.html` с простым HTML-кодом подвала.
- Сделайте коммит с сообщением «Add footer component».
- Отправьте ветку на сервер.
- В интерфейсе GitHub/GitLab сравните ветку `feature/footer` с `main` (используйте вкладку «Compare»).
- Сделайте скриншот сравнения веток.

Практическое занятие № 3 «Копировать свой и чужой проект в свой репозиторий»

Вариант 1

- Найдите публичный репозиторий на GitHub (например, учебный проект с открытым кодом).
- Скопируйте ссылку для клонирования (через SSH или HTTPS).
- На компьютере выполните команду `git clone <ссылка>`.
- Перейдите в папку с клонированным проектом.
- Создайте новый репозиторий на GitHub с названием `cloned-project`.
- Свяжите локальный проект с новым репозиторием: `git remote add my-repo <ссылка-на-ваш-репозиторий>`
- Отправьте код в новый репозиторий (`git push my-repo main`).
- Сделайте скриншот страницы нового репозитория.

Вариант 2

- Клонировать репозиторий `https://github.com/octocat/Spoon-Knife.git` на свой компьютер.
- Перейдите в папку проекта.

- Создайте новый файл my-change.txt с любым текстом.
- Сделайте коммит с сообщением «My first change».
- Создайте на GitHub репозиторий forked-spoon.
- Свяжите локальный проект с новым репозиторием и отправьте изменения.
- Проверьте, что файл my-change.txt появился в новом репозитории.
- Сделайте скриншот файла в репозитории.

Вариант 3

- Клонировать любой публичный репозиторий с GitHub на свой компьютер.
- Удалите папку .git из клонированного проекта (это отвяжет его от исходного репозитория).
- Создайте новый репозиторий на GitHub с названием local-copy.
- В папке проекта выполните: git init, git add, git commit -m "Initial commit in new repo", git remote add origin <ссылка-на-ваш-репозиторий>, git push -u origin main
- Убедитесь, что все файлы загружены.
- Сделайте скриншот страницы репозитория с историей коммитов.

Практическое занятие № 4 «Коллективная разработка»

Вариант 1

- Создайте репозиторий team-work на GitHub.
- Добавьте соавтора (учётную запись преподавателя/одногруппника) с правами на запись.
- Каждый участник создаёт в репозитории свой файл: Участник 1: task1.txt, Участник 2: task2.txt
- Каждый делает коммит и отправляет изменения в ветку main.
- Проверьте, что все файлы появились в репозитории.
- Сделайте скриншот списка файлов и истории коммитов.

Вариант 2

- Создайте репозиторий group-project на GitHub.
- Добавьте 2 соавторов.
- Каждый участник создаёт отдельную ветку: Участник 1: feature/menu, Участник 2: feature/sidebar
 - В своих ветках добавьте по 1 файлу (например, menu.html, sidebar.css).
 - Сделайте коммиты и отправьте ветки на сервер.
 - Создайте Pull Request для каждой ветки в main.
 - Одобрите PR друг друга (если есть доступ) или опишите, как это сделать.
 - Сделайте скриншот страниц PR.

Вариант 3

- Создайте репозиторий collaboration-demo на GitHub.
- Добавьте 1 соавтора.
- Участник 1 создаёт ветку content и добавляет файл article.md с текстом.
- Участник 2 создаёт ветку design и добавляет файл styles.css с стилями.
- Каждый делает коммит и отправляет ветку на сервер.
- Участник 1 оставляет комментарий к файлу styles.css через интерфейс GitHub (в режиме «Files changed» в PR).
- Участник 2 отвечает на комментарий и вносит правки.
- Сделайте скриншот обсуждения в PR.

Критерии оценивания результатов выполнения практических работ и подготовки отчета по практическому занятию

Критериями оценки результативности практического занятия являются:

- степень реализации цели и задач работы;
- степень выполнения заданий;

- степень соответствия результатов работы заданным требованиям;
- качество подготовки отчета;
- степень сформированности у студентов необходимых умений и навыков.

«ОТЛИЧНО». Обучающийся выполняет профессиональные действия и демонстрирует практические умения без ошибок, в полной мере владеет учебным материалом, самостоятельно интерпретирует полученные результаты, технически грамотно формулирует выводы. Не допускает ошибок в процессе защиты отчёта. Отчёт оформлен в соответствии с установленными требованиями.

«ХОРОШО». Обучающийся выполняет профессиональные действия и демонстрирует практические умения с небольшими недочетами, в полной мере владеет учебным материалом, контрольные вопросы содержат недостатки, технически грамотно формулирует выводы. Задания для самостоятельного выполнения осуществляет с незначительными ошибками. Допускает незначительные ошибки в процессе защиты отчёта. Отчёт оформлен в соответствии с установленными требованиями;

«УДОВЛЕТВОРИТЕЛЬНО». Обучающийся выполняет профессиональные действия и демонстрирует практические умения с недочетами, слабо владеет учебным материалом, контрольные вопросы содержат ошибки, неграмотно формулирует выводы. Задания для самостоятельного выполнения осуществляет с ошибками. Допускает ошибки в процессе защиты отчёта. Отчёт оформлен в соответствии с установленными требованиями;

«НЕУДОВЛЕТВОРИТЕЛЬНО». Обучающийся выполняет профессиональные действия и демонстрирует практические умения со значительными ошибками, не владеет учебным материалом, контрольные вопросы содержат ошибки, неграмотно формулирует выводы. Задания для самостоятельного выполнения осуществляет неправильно. Допускает

ошибки в процессе защиты отчёта. Отчёт оформлен не в соответствии с установленными требованиями.

ОЦЕНОЧНЫЕ СРЕДСТВА И КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Дифференцированный зачет

Промежуточная аттестация по междисциплинарному курсу МДК 01.03 «Тестирование информационных ресурсов» в форме комплексного дифференцированного зачета (зачет с оценкой).

До даты проведения комплексного дифференцированного зачета необходимо пройти оценивание всех работ, выполненных в течение IV семестра (2 курс обучения): теоретическое обучение, практические занятия.

Оценки комплексного дифференцированного зачета – отлично, хорошо, удовлетворительно, неудовлетворительно.

Примерный перечень вопросов к комплексному дифференцированному зачету в части МДК.01.03

1 Поясните, что такое тестирование ПО? Назовите его цели и основные задачи.

2 Перечислите и кратко поясните основные принципы тестирования.

3 Выделите ключевую разницу между верификацией и валидацией?

Приведите примеры.

4 Опишите жизненный цикл дефекта, перечислите основные статусы.

5 Перечислите, что входит в тестовую документацию? Назовите 3-4 ключевых документа и их назначение.

6 Разъясните, чем отличается позитивное тестирование от негативного? Приведите примеры.

7 Перечислите основные виды тестирования по объектам (не менее 4), дайте краткую характеристику.

8 Опишите уровни тестирования (модульное, интеграционное, системное, приёмочное), выделите их отличия.

9 Разъясните, что такое регрессионное тестирование? Когда и зачем оно проводится?

- 10 Поясните, в чём суть дымового и санитарного тестирования?
- 11 Перечислите составляющие тестирования безопасности? Назовите 2-3 типичные угрозы и способы проверки.
- 12 Сравните подходы «чёрного», «белого» и «серого» ящика: области применения.
- 13 Расскажите, что такое статическое и динамическое тестирование? Приведите по одному примеру метода.
- 14 Объясните технику эквивалентного разбиения на примере.
- 15 Поясните, что такое анализ граничных значений? Приведите пример.
- 16 Назовите 3-4 инструмента для ручного и автоматизированного тестирования веб-приложений.
- 17 Расскажите, как использовать браузерные DevTools для тестирования? Перечислите 2-3 ключевые вкладки.
- 18 Расскажите, что такое лог-файлы и как они помогают в тестировании?
- 19 Опишите краткий процесс тестирования API. Перечислите необходимые инструменты?
- 20 Опишите процесс кросс-браузерного тестирования? Назовите 1-2 инструмента для его проведения.
- 21 Перечислите приёмы, которые помогают проверить адаптивность веб-страницы? Расскажите о методах и инструментах.
- 22 Перечислите, из каких этапов состоит процесс тестирования (от планирования до отчётности)?
- 23 Расскажите, что включает тест-план? Назовите 3–4 обязательных раздела.
- 24 Поясните, как составляется чек-лист? Приведите краткий пример для простого функционала.
- 25 Разъясните, что такое тест-кейс? Перечислите 3-4 обязательных поля и приведите пример.

26 Объясните, что такое серьёзность и приоритет дефекта? Приведите примеры сочетаний.

27 Объясните, как правильно описать дефект в системе учёта? Назовите 3-4 ключевых поля.

28 Дайте определение понятию «доступность веб-контента»? Назовите 1-2 стандарта и способ проверки.

29 Перечислите 3-4 ключевых аспекта тестирования веб-интерфейса.

30 Поясните роль тестировщика в Agile/Scrum-команде? Как тестирование интегрируется в итерации?

Примерный перечень практических заданий к комплексному дифференцированному зачету в части МДК.01.03

1 Составьте тест-план для веб-приложения «Онлайн-магазин» (укажите цели, объекты, риски, критерии выхода).

2 Разработайте чек-лист для проверки функционала корзины покупок (не менее 10 пунктов).

3 Напишите тест-кейс для сценария «Авторизация пользователя» (обязательно включите: предусловия, шаги, ожидаемый результат, приоритет).

4 Проанализируйте требования к форме обратной связи и составьте 5 тест-кейсов (включая негативные сценарии).

5 Оформите отчёт о тестировании для модуля поиска товаров (включите статистику по дефектам и выводы).

6 Проверьте адаптивность главной страницы сайта на 3 разрешениях (например, 1920×1080, 1280×720, 375×667). Опишите найденные несоответствия.

7 Протестируйте форму регистрации на соответствие требованиям (обязательные поля, валидация email, ограничения по длине).

8 Выполните юзабилити-тестирование навигации сайта (логичность меню, доступность ссылок, понятность кнопок).

9 Проверьте доступность (accessibility) страницы товара:

контрастность текста, подписи для изображений, навигация с клавиатуры.

10 Протестируйте интерактивные элементы (карусель товаров, всплывающие подсказки) на корректность работы.

11 Найдите и задокументируйте 3 дефекта на демо-версии сайта. Оформите баг-репорты с указанием приоритета.

12 Воспроизведите дефект из описания: «При нажатии кнопки «Оформить заказ» страница зависает» и составьте пошаговый отчёт.

13 Определите, является ли ситуация дефектом: «Кнопка «Купить» неактивна, пока не выбран размер товара». Обоснуйте ответ.

14 Классифицируйте предложенные дефекты по строгости приоритета. Приведите по 1 примеру.

15 Исправьте некорректно оформленный баг-репорт (дополните недостающие данные).

16 Протестируйте поиск по сайту: проверьте обработку пустых запросов, спецсимволов, неточностей в написании.

17 Проверьте фильтрацию товаров по категориям и цене. Уделить внимание граничным значениям, сбросу фильтров.

18 Протестируйте сценарий «Оформление заказа»: выбор доставки, применение промокода, подтверждение оплаты.

19 Проверьте работу пагинации в каталоге товаров (переход между страницами, отображение количества элементов).

20 Протестируйте функцию «Сравнить товары»: добавление/удаление позиций, отображение различий.

21 Используя DevTools, проверьте сетевые запросы при загрузке главной страницы. Зафиксируйте время ответа, коды статусов.

22 Проанализируйте лог-файлы сервера и найдите ошибки, связанные с 404-страницами.

23 С помощью DevTools имитируйте медленное соединение и оцените поведение сайта.

24 Проверьте кросс-браузерность формы входа в Chrome и Firefox

Опишите различия.

25 Используйте DevTools для проверки валидности HTML/CSS. Найдите и зафиксируйте 2-3 предупреждения.

26 Составьте сценарий нагрузочного тестирования для страницы с товарами. Укажите метрики: время отклика, количество пользователей.

27 Протестируйте локализацию сайта: проверьте перевод кнопок, формат даты/валюты для региона RU.

28 Проверьте безопасность формы входа: ввод SQL-инъекций, XSS-кода, перебор паролей.

29 Протестируйте работу сайта без JavaScript: оцените доступность основных функций.

30 Проверьте SEO-параметры страницы (наличие meta-тегов, alt-описаний для изображений, заголовков H1-H3).

Критерии оценивания знаний обучающихся на дифференцированном зачете

«ОТЛИЧНО». Полное освоение теоретического материала, безупречное выполнение практических заданий, креативные и обоснованные решения, чёткая защита работ.

«ХОРОШО». Хорошее знание теории, выполнение заданий с единичными негрубыми ошибками, умение обосновывать решения, но возможны небольшие недочёты в оформлении.

«УДОВЛЕТВОРИТЕЛЬНО». Базовое понимание материала, выполнение заданий с ошибками, затруднение при ответе на дополнительные вопросы, неполнота в изложении.

«НЕУДОВЛЕТВОРИТЕЛЬНО». Незнание ключевых понятий, грубые ошибки в выполнении заданий, неспособность обосновать решения, существенные пробелы в знаниях.