ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

КАФЕДРА «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
«Методологии разработки решений на основе ИИ»

Направление подготовки

09.03.01 Информатика и вычислительная техника

ОПОП

«Программно-аппаратное обеспечение вычислительных комплексов и систем искусственного интеллекта»

Квалификация (степень) выпускника — бакалавр

Форма обучения — очная

Рязань

1 ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (практических заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности компетенций, закрепленных за дисциплиной.

Контроль знаний обучающихся проводится в форме промежуточной аттестации. Промежуточная аттестация проводится в форме зачета.

Форма проведения зачета – тестирование, письменный опрос по теоретическим вопросам.

2 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции (или ее части) в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Уровень освоения компетенций, формируемых

дисциплиной: Описание критериев и шкалы оценивания

тестирования:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 70 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 50 до 69%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 49%

Описание критериев и шкалы оценивания теоретического вопроса:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов

1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя
0 баллов	выставляется студенту, который не смог ответить на вопрос

На промежуточную аттестацию (зачет) выносится тест, два теоретических вопроса. Максимально студент может набрать 6 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «зачтено», «не зачтено».

Оценка «зачтено» выставляется студенту, который набрал в сумме не менее 4 баллов (выполнил одно задание на эталонном уровне, другое – не ниже порогового, либо оба задания выполнит на продвинутом ур овне). Обязательным условием является выполнение всех предусмотренных в течение семестр а практических и лабораторных работ заданий.

Оценка «не зачтено» выставляется студенту, который набрал в сумме менее 4 баллов, либо имеет к моменту проведения промежуточной аттестации несданные практические, либо лабораторные работы.

3 ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

Контролируемые разделы (темы) дисциплины	**Код контролируемой компетенции (или её части)**	**Вид, метод, форма оценочного мероприятия**
Раздел 1. Основные понятия и цели дисциплины. Среды разработки ПО и их основные функции.	ПК-2.1	**Зачет**
Раздел 2. Системы контроля версий. Общие принципы работы. Централизованные и распределенные системы.	ПК-2.2	**Зачет**
Раздел 3. Основы коммуникационной культуры в команде разработчиков ИИ. Методы совместного планирования. Понимание процессов Agile и Scrum. Система контроля версий Subversion.	УК-12.1, УК-12.2, ПК-2.2	**Зачет**
Раздел 4. Система контроля версий GIT. Основы работы, ветвление, работа с удаленными репозиториями.	ПК-2.2	**Зачет**
Раздел 5. Системы отслеживания ошибок, средства автоматизации тестирования.	ПК-2.2	**Зачет**
Раздел 6. Виртуализация. Принципы, преимущества, типы виртуализации.	ПК-2.1	**Зачет**
Раздел 7. Контейнеризация. Применение Docker для разработки решений на основе ИИ.	УК-12.1, УК-12.2, ПК-2.1	**Зачет**

4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

Промежуточная аттестация в форме зачета

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ПК-2	Способен проектировать и разрабатывать программное обеспечение

ПК-2.1. Проектирует и разрабатывает программное обеспечение
Знать Специфику проектирования и разработки программного обеспечения с применением современных инструментальных средств **Уметь** Выбирать адекватные ставящимся задачам инструменты и технологии реализации программного обеспечения **Владеть** Навыками участия в разработке программного обеспечения с применением современных инструментальных средств

1. Что такое система контроля версий? (выберите один вариант ответа)

а) Программное обеспечение, призванное избавить программиста от необходимости загружать проект к себе на машину

б) Программное обеспечение, призванное автоматизировать работу с историей файла, и организовать защищенное хранилище проекта

в) Программное обеспечение, позволяющее версионировать проект

2. Что такое репозиторий? (выберите один вариант ответа)

а) Место, где система контроля версий хранит все документы вместе с их историей

б) Директория для вашего проекта

в) Рабочая копия документов

3. Соедините описание системы контроля версий с её названием (соединить)

Локальные	СКВ, в которых вся работа производится с центральным хранилищем (все действия, так или иначе, зависят от него)
Централизованные	СКВ, главной парадигмой которых является локализация данных на машине каждого разработчика проекта
Децентрализованные	СКВ, хранящиеся только на локальном компьютере

4. Объедините СКВ с её типом (соединить)

GIT	Локальная

SVN	Распределённая и централизованная
Mercurial	Централизованная
CVS	Децентрализованная
Bazaar	Распределённая и локальная

5. К какому виду систем контроля версий относится Subversion? (выберите один вариант ответа)

а) Централизованному

б) Распределенному

в) Локальному

г) Всем вышеперечисленным

6. Соотнесите состояние хранилища и то, какие действия предпримет Subversion во время коммита (соединить)

Хранилище изменялось локально и не устарело	Система не сделает ничего, вам необходимо внести изменения и сделать коммит
Хранилище не изменялось и устарело	Никаких
Хранилище изменялось локально и устарело	Заберет изменения с сервера
Хранилище не изменялось и не устарело	Система забирает к себе изменения с сервера и пользователю нужно разрешить конфликты, если необходимо

7. Соотнесите модель с её описанием (соединить)

Блокирование Изменение Разблокирование	Оба пользователя копируют файл и редактируют его на своём компьютере. Затем один из пользователей отправляет изменения на сервер. Второй пользователь заканчивает свои изменения и тоже хочет отправить файлы на сервер, но не может, т.к. там уже находится обновленная версия. Он должен сначала взять изменения с сервера, объединить изменения, разрешить конфликты, если они есть, и затем отправить свои изменения на сервер.
Копирование — Изменение Слияние	Пользователь, решивший редактировать файл, блокирует его, в то время другой пользователь не может его редактировать. При разблокировке файла следующим участником другой пользователь может забрать изменения предыдущего пользователя и редактировать файл дальше.

8. Отметьте, какие команды вы можете использовать для получения подсказок в Subversion (выберите несколько вариантов ответа)

а) svn help

б) svn resolve

в) svn commit

г) svn update

д) svn log

е) svn info

ж) svn revert

з) svn delete

9. Какие команды вы можете использовать для создания/получения репозитория? (выберите несколько вариантов ответа)

а) svnadmin create /path/to/rep

б) svn import mytree file:///usr/local/svn/newrepos/some/project \
-m "Initial import"

в) svn checkout http://svn.example.com/repos/calc

10. Отметьте команды для внесения изменений в рабочую копию (выберите несколько вариантов ответа)

а) svn add

б) svn delete

в) svn copy

г) svn move

д) svn mkdir

е) svn update

11. Объедините способы разрешения конфликтов с их описанием (соединить)

1) Объединить конфликтный файл: Конфликтный файл будет выглядеть так: <<<<<<< имя файла ваши изменения ======= результат автоматического слияния с репозиторием >>>>>>> ревизия 2) Выполнить "svn resolved"	Отмена ваших изменений

1) Выполнить "svn revert file"	Объединение вручную
Скопировать filename.rOLDREV или filename.rNEWREV	Копирование одного из файлов (своего или чужого)

12. Вопрос: верно ли, что в Git и Subversion используются принципиально разные подходы к хранению файлов репозитория? (выберите один вариант ответа)

а) да

б) нет

13. Вопрос: каковы были основные цели, преследуемые при создании Git? (выберите один вариант ответа)

а)

1. Простая архитектура.
2. Полная децентрализация.
3. Хорошая поддержка нелинейной разработки.

б)

1. Высокая скорость работы.
2. Применение подхода CVS.
3. Поддержка нелинейной разработки.

в)

1. Простая архитектура.
2. Централизованный подход.
3. Поддержка нелинейной разработки.

14. Под какой лицензией выпущена системы контроля версий Git? (краткий ответ)

---

15. Какой алгоритм использует Git для вычисления хэш-сумм? (выберите один вариант ответа)

а) CRC32

б) MD6

в) SHA-1

г) SHA-2

16. Что такое ветка в Subversion? (выберите один вариант ответа)

а) Направление разработки, которое существует независимо от другого направления

б) Направление разработки, которое существует независимо от другого направления, однако имеющие с ним общую историю

в) Копия репозитория у любого из разработчиков

17. В каких случаях есть необходимость создать ветку? (выберите несколько вариантов ответа)

а) Изменения, которые вы хотите внести, могут повредить работающему коду

б) Вы хотите написать улучшение/оптимизировать существующий код

в) Вы хотите зафиксировать изменения

18. Выберите верные сведения о создании ветвей в Subversion (выберите несколько вариантов ответа)

а) При использовании команды svn copy <удаленный url> <удаленный url> -m “Сообщение коммита”, вы создаёте ветку на удалённом сервере, а не на локальной машине

б) При использовании команды svn copy <удаленный url> <удаленный url> -m “Сообщение коммита”, вы создаёте ветку на локальной машине

в) При использовании команды svn copy <удаленный url> <удаленный url> -m “Сообщение коммита”, вам не обязательно иметь рабочую копию

г) При использовании команды svn copy <локальная директория> <локальная директория> вы создаёте ветку на локальной машине

д) При использовании команды svn copy <локальная директория> <локальная директория> вы создаёте ветку на удаленном сервере, а не на локальной машине

19. При создании ветки в Git командой git branch <branchname>, происходит ли автоматический переход на эту ветку? (выберите один вариант ответа)

а) да

б) нет

20. Что происходит при команде git checkout <branchname> (выберите один вариант ответа)

а) Мы переходим в директорию branchname

б) Указатель HEAD перемещается на ветку branchname

в) Мы создаём ветку branchname и перемещаемся на неё

21. По каким протоколам можно настроить работу Git на сервере? (выберите несколько вариантов ответа)

а) HTTP

б) FTP

в) SSH

г) Telnet

д) Git

е) SMTP

22. Что вам нужно сделать с проектом в случае «вы хотите внести изменения в репозиторий, в который у вас нет доступа»? (выберите один вариант ответа)

а) fork

б) pull request

23. Выберите случаи, когда вам может потребоваться использовать GitHub (выберите несколько вариантов ответа)

а) создать свой проект с открытым исходным кодом / использовать как хранилище кода, который не хотелось бы потерять

б) воспользоваться сторонней библиотекой (не входящей в список стандартных)

в) внести вклад в уже существующий проект на GitHub

Типовые вопросы открытого типа

1. Запишите команду SVN, позволяющую создать новый репозиторий (svnadmin create <путь>)
2. Запишите команду SVN, позволяющую создать рабочую копию (svn checkout <путь до репозитория>)
3. Запишите команду SVN, позволяющую просмотреть информацию о репозитории (svn info)
4. Запишите команду SVN, позволяющую просмотреть историю изменений (svn log)
5. Запишите команду SVN, позволяющую добавить файл под версионный контроль (svn add)
6. Запишите команду SVN, позволяющую обновить рабочую копию (svn update)
7. Запишите команду SVN, позволяющую зафиксировать изменения в репозиторий (svn commit)
8. Запишите команду SVN, позволяющую убрать файл из-под версионного контроля (svn delete)
9. Запишите команду SVN, позволяющую привести рабочую копию в согласованное состояние и очистить лог незавершенных операций (svn cleanup)
10. Запишите команду SVN, позволяющую объединить изменение из двух ветвей (svn merge)
11. Запишите команду GIT, позволяющую создать пустой репозиторий (git init)
12. Запишите команду GIT, позволяющую клонировать существующий репозиторий (git clone <путь>)
13. Запишите команду GIT, позволяющую выполнить первоначальную настройку репозитория (git config)
14. Запишите команду GIT, позволяющую проиндексировать изменения в рабочей директории (git add)
15. Запишите команду GIT, позволяющую зафиксировать изменения в локальный репозиторий (git commit)
16. Запишите команду GIT, позволяющую зафиксировать изменения в удаленный репозиторий (git push)
17. Запишите команду GIT, позволяющую получить данные из удаленного репозитория (git pull)
18. Запишите команду GIT, позволяющую создать ветку (git branch <имя ветки>)
19. Запишите команду GIT, позволяющую переключиться на созданную ветку (git checkout <имя ветки>)
20. Запишите команду GIT, позволяющую присоединить метку к текущему коммиту (git tag)
21. Запишите команду GIT, позволяющую выполнить слияние двух веток (git merge)
22. Запишите команду GIT, позволяющую просмотреть историю коммитов(git log)
23. Запишите команду GIT, позволяющую просмотреть проиндексированные изменения (git diff staged)
24. Запишите команду GIT, позволяющую просмотреть выполненные локальные изменения (git diff)
25. Запишите команду GIT, позволяющую просмотреть справку по интересующей команде (git help <команда>)

ПК-2.2. Применяет современные инструментальные средства при разработке программного обеспечения
Знать Современные инструментальные средства, применяемые при коллективной разработке программного обеспечения **Уметь** Выбирать инструментальные средства исходя из фактический потребностей проекта **Владеть** Навыками работы в команде, с использованием современных программных средств коллективной разработки, по проектированию информационных систем и программного обеспечения

24. Что такое система отслеживания ошибок? (выберите один вариант ответа)

а) Прикладная программа, которая ищет ошибки в вашем коде, и следит за этими участками

б) Прикладная программа, разработанная с целью помочь разработчикам программного обеспечения учитывать и контролировать ошибки, найденные в программах, пожелания пользователей, а также следить за процессом устранения этих ошибок и выполнением или невыполнения пожеланий

25. Что такое «баг» (в программировании)? (выберите один вариант ответа)

а) Жук, который забрался в системный блок компьютера

б) Ошибка в программе, которая выдаёт неожиданный или неправильный результат

в) Программа, которая пишется, чтобы найти ошибки в программе

26. На каком этапе программы (или кем) могут быть обнаружены «баги»? (выберите несколько вариантов ответа)

а) В процессе тестирования программы

б) В процессе отладки программы

в) На этапе проектирования программы

г) Сторонними пользователями приложения

27. Расставьте в нужном порядке последовательность действий при обнаружении ошибки технического характера (порядок)

а) Сообщение об обнаруженной ошибке технической поддержке приложения

б) Взятие ошибки в обработку членом команды разработки

в) Обнаружение ошибки пользователем

г) Исправление ошибки (закончено)

д) Выпуск патча (исправления) к существующему приложению

е) Занесение ошибки в систему отслеживания ошибок

ж) Работа над исправлением ошибки (члена команды разработки)

з) Смена статуса ошибки в системе отслеживания ошибок на «в работе»

и) Смена статуса ошибки в системе отслеживания ошибок на «закрыта»

28. Какие сведения об дефекте могут хранится в системе отслеживания ошибок? (выберите несколько вариантов ответа)

а) Номер (идентификатор) дефекта

б) Кто сообщил о дефекте

в) Версия продукта, в которой обнаружен дефект

г) Обсуждение того, кто возьмет задачу по устранению

д) Серьёзность (критичность) дефекта и приоритет решения

е) Описание шагов для выявления дефекта (воспроизведения неправильного поведения программы)

ж) Кто ответственен за устранение дефекта

з) Обсуждение возможных решений и их последствий

и) Текущее состояние (статус) дефекта

к) Версия продукта, в которой дефект исправлен

29. В каком состоянии может быть дефект? (выберите несколько вариантов ответа)

а) Открыт

б) Назначен

в) Некому назначить

г) Проверен

д) Отклонен

е) Закрыт

30. Что такое тестирование? (выберите несколько вариантов ответа)

а) Одна из техник контроля качества, включающая в себя активности по планированию работ, проектированию тестов, выполнению тестирования и анализу полученных результатов

б) Работа тестировщика

в) Проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом

31. Выберите виды тестов, входящие в пирамиду тестирования (Майк Кон) (выберите несколько вариантов ответа)

а) юнит-тесты

б) интеграционные тесты

в) тесты корректности работы операционной системы, на которой будет установлено приложение

г) тесты пользовательского интерфейса

д) тесты правильности сборки аппаратной платформы компьютера

32. Выберите положения, правильные для юнит тестирования (выберите несколько вариантов ответа)

а) Должны не зависеть от окружения, на котором они выполняются

б) Запускаться регулярно в автоматическом режиме

в) Должны выполняться под специально настроенным окружением

г) Должны запускаться вручную для контроля за ними

33. Соедините методологию тестирования и ее описание (соединить)

Тестирование черного ящика	метод тестирования программного обеспечения, который предполагает, что внутренняя устройство системы известны тестировщику
Тестирование методом серого ящика	метод тестирования, базируется только лишь на тестировании по функциональной спецификации и требованиям, при этом не имея доступа во внутреннюю структуру кода и базу данных.
Тестирование методом белого ящика	Метод тестирования ПО, который предполагает, что внутреннее устройство программы нам известно лишь частично

34. Расставьте в нужном порядке последовательность действий при разработке через тестирование (порядок)

а) Пишеться тест, покрывающие желаемое изменение

б) Проводится рефакторинг нового кода к соответствующим стандартам

в) Пишется код, который позволит пройти тест

35. Соедините вид теста и то, что он проверяет (соединить)

Интеграционные тесты	Тестирование каждой нетривиальной функции или метода
Юнит-тесты	Тестирование корректного взаимодействия с другими приложениями
Тесты пользовательского интерфейса	Тестирование правильности работы пользовательского интерфейса приложения

36. Какие функции должна выполнять служба CI? (выберите несколько вариантов ответа)

а) получение исходного кода из репозитория
б) сборка проекта (компиляция)
в) выполнение тестов
г) развёртывание готового проекта
д) отправка отчетов (на электронную почту)
е) последовательное выведение релиза в промышленную среду

37. Как вы помните, CD включает в себя и CI. Напишите кратко, какую главную функцию выполняет CD, но не выполняет CI (краткий ответ)

38. Какие задачи входят в обязанности Dev-Ops инженера? (выберите несколько вариантов ответа)

а) Системное администрирование

б) Тестирование кода разработчиков

в) Программирование

г) Использование облачных технологий

д) Автоматизация крупной инфраструктуры

39. Соедините подкатегорию виртуализации с ее описанием (соединить)

виртуализация платформ	Данный вид виртуализации преследует своей целью комбинирование или упрощение представления аппаратных ресурсов для пользователя и получение неких пользовательских абстракций оборудования, пространств имен, сетей и т.п
виртуализация ресурсов	Продуктом этого вида виртуализации являются виртуальные машины некие программные абстракции, запускаемые на платформе реальных аппаратнопрограммных систем

40. Какие подвиды включает виртуализация платформ? (выберите несколько вариантов ответа)

а) Разделение ресурсов

б) Инкапсуляция

в) Частичная виртуализация

г) Паравиртуализация

д) Виртуализация уровня операционной системы

е) Полная эмуляция (симуляция).

ж) Частичная эмуляция (нативная виртуализация).

з) Виртуализация уровня приложений

41. Какие подвиды включает виртуализация ресурсов? (выберите несколько вариантов ответа)

а) Разделение ресурсов

б) Инкапсуляция

в) Частичная виртуализация

г) Паравиртуализация

д) Виртуализация уровня операционной системы

е) Кластеризация компьютеров и распределенные вычисления (grid computing)

ж) Частичная эмуляция (нативная виртуализация).

з) Виртуализация уровня приложений

и) Объединение, агрегация и концентрация компонентов

42. Что такое контейнеризация? (выберите один вариант ответа)

а) Подход к разработке программного обеспечения, при котором приложение или служба, их зависимости и конфигурация упаковываются вместе в образ контейнера

б) Абстракция вычислительных ресурсов и предоставление пользователю системы, которая «инкапсулирует» (скрывает в себе) собственную реализацию

в) Набор изолированных приложений, не взаимодействующих друг с другом

43. Преимущества контейнеризации перед виртуализацией (выберите несколько вариантов ответа)

а) Более простая настройка контейнеров

б) Контейнеры требуют гораздо меньше ресурсов

в) Контейнеры легко развертывать

г) Контейнеры быстрее запускаются

д) Приложение в контейнере выполняется в любой среде

44. Соедините термин с его определением (соединить)

Контейнер	пакет со всеми зависимостями и сведениями, необходимыми для создания контейнера
Образ контейнера	действие по созданию образа контейнера на основе сведений и контекста, предоставленных файлом Dockerfile, а также дополнительных файлов в папке, где создается образ
Сборка	текстовый файл, содержащий инструкции по сборке образа Docker
Dockerfile	экземпляр образа Docker

45. Какие компоненты входят в каждый контейнер Docker? (выберите несколько вариантов ответа)

а) Выбранная операционная система (например, дистрибутив Linux, Windows Nano Server или Windows Server Core)

б) Файлы, добавленные разработчиком (двоичные файлы приложения и т. п.)

в) Сведения о конфигурации (параметры среды и зависимости)

г) Тесты для приложения

Ответы:

№ вопроса	Ответ	№ вопроса	Ответ
1	б	34	а, в, б
2	а	35	1-2, 2-1, 3-3
3	1-3, 2-1, 3-2	36	а, б, в, г, д
4	1-4 , 2-3 , 3-4 , 4-3 , 5-2	37	Ведение релиза к промышленной среде
5	а	38	а, в, г, д
6	1-2 , 2-3 , 3-4 , 4-1	39	1-2, 2-1
7	1-2, 2-1	40	в, г, д, е, ж, з
8	а, е	41	а, е, и
9	а, б, в	42	а
10	а, б, в, г, д, е	43	б, в, г, д
11	1-2, 2-1, 3-3	44	1-4, 2-1, 3-2, 4-3
12	а	45	а, б, в
13	а		
14	GNU GPL 2		
15	в		
16	б		
17	а, б		
18	а, в, г		
19	б		
20	б		
21	а, в, д		
22	б		
23	а, б, в		
24	б		
25	б		
26	а, б, г		
27	в, а, е, б, з, ж, г, и, д		
28	Все кроме «г»		
29	а, б, г, д, е		
30	а, в		
31	а, б, г		
32	а, б		
33	1-2 , 2-3, 3-1		

Код компетенции	Результаты освоения ОПОП Содержание компетенций
УК-12	Способен осуществлять свою трудовую деятельность с учётом необходимости эффективной коммуникации и взаимодействия в рамках коллективной проектной работы в сфере ИИ

УК-12.1. Эффективно коммуницирует с участниками проектной команды при планировании, реализации и анализе результатов работы
Знать Основные принципы командной работы в проектах, связанных с ИИ (роли, процессы, зоны ответственности). Методы эффективной коммуникации (активное слушание, конструктивная обратная связь, управление конфликтами). Инструменты коллективной работы (Jira, YouTrack, Git,) и их применение в ИИ-проектах. **Уметь** Четко формулировать цели, задачи и требования к проекту с учетом особенностей ИИ-разработки. Распределять роли и задачи в команде с учетом компетенций участников. **Владеть** Навыками использования инструментальных средств коммуникации с участниками проектной команды

Типовые тестовые вопросы

1. *Какой из перечисленных инструментов НАИБОЛЕЕ эффективен для ежедневной синхронной коммуникации в распределенной команде над проектом ИИ?* (Выберите один вариант) а) Электронная почта
б) Jira
в) **Slack/Telegram**
г) Хранилище Git

2. *Основная цель ежедневного стендапа (daily scrum) в рамках Agile — это:* (Выберите один вариант)
а) Провести детальный разбор архитектуры системы
б) Обсудить зарплаты участников команды
в) **Синхронизироваться по текущим задачам, помехам и планам на день**
г) Утвердить итоговый отчет по проекту

3. *Что из перечисленного является ПРАВИЛЬНЫМ примером постановки задачи по методу SMART для этапа сбора данных в проекте ИИ?* (Выберите один вариант)
а) "Собрать побольше данных"
б) **"К 25.10.2023 ответственный Иванов А.И. должен собрать и провести первичную разметку не менее 10 000 изображений дорожных сцен с точностью разметки не менее 95% для обучения модели компьютерного зрения"**
в) "Сделать датасет"
г) "Найти данные для обучения нейросети"

4. *Какая информация является КЛЮЧЕВОЙ для включения в отчет о выполненной задаче по исправлению ошибки (бага) в системе отслеживания?* (Выберите несколько вариантов)
а) **Идентификатор коммита, в котором было внесено исправление**
б) Рассуждения о том, кто виноват в появлении ошибки
в) **Краткое описание выполненного решения**
г) **Ссылка на тест-кейс, проверяющий исправление**

5. *Ключевой принцип Agile "Работающий продукт важнее исчерпывающей документации" означает, что:* (Выберите один вариант)
а) Документацию можно не писать никогда
б) **Команда должна регулярно демонстрировать заказчику работающие инкременты продукта**
в) Весь проект должен быть полностью описан до начала разработки
г) Документация не важна для проектов ИИ

6. *При возникновении конфликта между разработчиком и тестировщиком по поводу критичности найденной ошибки, ПЕРВООЧЕРЕДНОЕ действие — это:* (Выберите один

вариант)
а) Эскалация конфликта на руководителя проекта
б) **Совместное обращение к формальным требованиям (спецификации) продукта**
в) Игнорирование ошибки тестировщиком
г) Отказ разработчика от дальнейшей работы

7. *Для эффективного планирования задач на спринт команде следует использовать:* (Выберите один вариант)

а) Мозговой штурм без фиксации результатов
б) **Backlog продукта и Backlog спринта**
в) Только устные договоренности
г) Указания только тимлида

8. *Какая из перечисленных практик НЕ способствует прозрачности и улучшению коммуникации в команде?* (Выберите один вариант)

а) Ведение актуального Backlog'а
б) **Сокрытие трудностей и срывов дедлайнов до последнего момента**
в) Использование общих досок задач (Kanban, Scrum)
г) Проведение ретроспектив по итогам спринта

9. *Что такое "Definition of Done" (DoD) в Scrum?* (Выберите один вариант)

а) Список всех возможных требований к продукту
б) **Четкий критерий, определяющий, когда задача считается выполненной**
в) Мнение продукт-оунера о готовности продукта
г) Отчет о затраченном времени

10. *Какой инструмент из перечисленных является системой отслеживания ошибок и управления проектами, а не средством коммуникации?* (Выберите один вариант)

а) Slack
б) **Jira**
в) Microsoft Teams
г) Telegram

Типовые вопросы открытого типа

1. Как называется методология разработки, ориентированная на итерации? (Agile)
2. Какой основной инструмент визуализации прогресса команды используется в Agile? (Доска задач)
3. Какой артефакт в Scrum содержит приоритизированный список всей работы над продуктом? (Бэклог продукта)
4. Как называется встреча по итогам спринта для обсуждения улучшений процесса? (Ретроспектива)
5. Какая система отслеживания ошибок имеет встроенную Wiki и тесно интегрирована с Git? (GitLab)
6. Какой статус задачи в системе отслеживания ошибок означает, что она назначена исполнителю? (Назначен)
7. Какая информация является главным компонентом системы отслеживания ошибок? (База данных дефектов)
8. Какой процесс в системах отслеживания ошибок (например, Jira) formalizes обсуждение решений между участниками команды? (Workflow / Рабочий процесс)
9.
10. Какой этап жизненного цикла дефекта следует после его разрешения разработчиком? (Проверка)
11. Какая практика в Git позволяет коллегам обсудить изменения перед их интеграцией в основную ветку? (Мерж-реквест / Пулл-реквест)

УК-12.2. Учитывает профессиональные и ролевые особенности коллег при совместной разработке технических решений и представлении результатов
Знать Ролевые особенности участников проектной команды **Уметь** Выстраивать деятельность и выбирать инструменты, исходя из ролевых особенностей коллег **Владеть** Навыками использования инструментальных средств, обеспечивающих взаимодействие между различными ролями в проектной команде

Типовые тестовые вопросы

11. *Data Scientist в команде ИИ-проекта передает модель разработчику для интеграции. Какая информация КРИТИЧЕСКИ ВАЖНА для разработчика?* (Выберите несколько вариантов)

а) **Требования к среде выполнения (версии Python, библиотеки)**
б) Мыслительные процессы Data Scientist при выборе архитектуры модели
в) **Формат входных данных и ожидаемый вывод модели**
г) Исходные сырые данные для обучения

12. *Разработчик создает новую ветку в Git для реализации функции. Чтобы другие члены команды понимали контекст, в сообщении к коммиту следует указать:* (Выберите несколько вариантов)

а) **Номер задачи в Jira/YouTrack**
б) Список всех измененных файлов
в) **Краткое содержательное описание внесенных изменений**
г) Имена всех коллег, которые могут быть заинтересованы

13. *Тестировщик обнаружил баг в работе ML-модели. При создании отчета в системе отслеживания ошибок, помимо описания шагов, ему следует обязательно приложить:* (Выберите несколько вариантов)

а) **Входные данные, на которых проявляется ошибка**
б) Свои предположения о том, какой код нужно исправить
в) **Логи или скриншоты с ошибочным поведением**
г) Требование об увольнении ответственного разработчика

14. *Какой инструмент позволяет разным ролям в проекте (менеджер, разработчик, тестировщик) иметь единое представление о статусе задач?* (Выберите один вариант)

а) Локальные текстовые файлы на каждом компьютере
б) **Общая доска задач в GitLab / Jira**
в) Электронная почта
г) Устные ежедневные поручения

15. *При создании Merge Request (MR) в GitLab, чтобы облегчить код-ревью коллеге, автор должен:* (Выберите несколько вариантов)

а) Создать MR без описания
б) **Указать понятное описание изменений и их цели**
в) **Убедиться, что пайплайн CI/CD успешно выполнился**
г) Попросить коллегу разобраться самостоятельно

16. *Менеджер проекта просит предоставить оценку времени на выполнение задачи. Как разработчик должен поступить, чтобы оценка была реалистичной и учитывала работу других членов команды?* (Выберите один вариант)

а) Назвать произвольный срок
б) **Разбить задачу на подзадачи и оценить их, заложив время на тестирование и код-ревью**
в) Сослаться на занятость и отказаться давать оценку
г) Дублировать оценку, которую дал коллега на похожую задачу

17. *Data Engineer подготовил новый набор данных. Чтобы эффективно передать его Data Scientist для работы, ему следует:* (Выберите несколько вариантов)

а) Передать данные по электронной почте отдельными файлами

б) **Задокументировать схему данных, источник и потенциальные проблемы качества**
в) **Обеспечить доступ к данным через версионное хранилище (например, DVC) или базу данных**
г) Устно описать содержимое данных во время обеда

18. *DevOps-инженер настраивает пайплайн CI/CD для проекта ИИ. Какие ролевые особенности он должен учесть при его проектировании?* (Выберите несколько вариантов)

а) **Для Data Scientist: наличие этапа тестирования и валидации данных и модели**
б) **Для разработчика: автоматический запуск юнит-тестов и сборки проекта**
в) Для менеджера: этап формирования отчета о заработной плате
г) **Для тестировщика: возможность запуска интеграционных тестов и тестов окружения**

19. *При проведении ретроспективы спринта, чтобы учесть мнение всех ролей в команде (разработчик, тестировщик, аналитик), рекомендуется:* (Выберите один вариант)

а) Выслушать только мнение тимлида
б) **Использовать технику, где каждый участник анонимно пишет свои плюсы и минусы прошедшего спринта**
в) Обсуждать только технические проблемы
г) Отменить ретроспективу, если все задачи выполнены

20. *Какой инструмент из перечисленных НАИБОЛЕЕ гибко позволяет настраивать рабочие процессы, поля и права доступа для разных ролей в проекте (например, чтобы тестировщик не мог менять статус задачи на "исправлено")?* (Выберите один вариант)

а) **Jira / YouTrack**
б) Git
в) Электронная почта
г) Общий чат в Telegram

Типовые вопросы открытого типа

1. Какая роль в команде обычно отвечает за заполнение поля «Шаги воспроизведения» в отчёте об ошибке? (Тестировщик)
2. Какая роль в команде изменяет статус задачи на «Исправлено» в системе отслеживания ошибок? (Разработчик)
3. С помощью какой функциональности GitLab менеджер проекта может контролировать процесс слияния кода? (Ветвления и мерж-реквесты)
4. Какая система из лекций (Jira, Redmine) известна сложной настройкой рабочих процессов для разных ролей? (Jira)
5. Кто в команде, работая с Subversion, должен разрешать конфликты слияния? (Разработчик)
6. Какая роль чаще всего использует систему CI/CD для развертывания готового проекта? (DevOps-инженер)
7. Для какой роли в проекте критически важно ведение актуальной технической документации в Wiki? (Всех участников / Разработчик)
8. Кто принимает решение о закрытии дефекта после его проверки? (Тестировщик)
9. Какая система из рассмотренных (MantisBT, Redmine) написана на Ruby и часто ассоциируется с гибкостью для разных ролей? (Redmine)
10. Какой протокол, поддерживаемый Git, обеспечивает безопасный доступ для разработчиков? (SSH)

Типовые теоретические вопросы:

Раздел 1. Основные понятия и инструменты

1. Современные инструментальные средства коллективной разработки программного обеспечения: назначение и классификация.

2. Специфика проектирования и разработки программного обеспечения с применением современных инструментальных средств.

Раздел 2. Системы контроля версий

3. Системы управления версиями. Сущность, назначение, преимущества.

4. Централизованные и распределенные системы контроля версий: сравнительный анализ.

5. Модели версионирования: «Копирование-Изменение-Слияние» и «Блокирование-Изменение-Разблокирование». Достоинства и недостатки.

6. Основные представители систем контроля версий на рынке (Git, Mercurial).

Раздел 3. Командная работа и инструменты (Agile, Scrum)

7. Принципы гибкой методологии разработки (Agile) и их значение для проектов в сфере ИИ.

8. Роли и артефакты в Scrum: продукт-оунер, скрам-мастер, разработчики, бэклоги.

9. Практики эффективной коммуникации в междисциплинарной команде разработчиков ИИ.

10. Инструменты коллективной работы (Jira, YouTrack, GitLab) и их применение в ИИ-проектах.

Раздел 4. Система контроля версий Git

11. Git: философия, история создания, преимущества перед другими системами.

12. Внутреннее устройство Git: объектная модель, хэширование, целостность данных.

13. Жизненный цикл файла и коммита в Git: отслеживаемые, измененные, подготовленные файлы.

14. Ветвление и слияние в Git: механизм, стратегии, разрешение конфликтов.

15. Модели ветвления: Git Flow и иные подходы, их применение в проектах.

16. Организация совместной серверной работы с Git (GitLab, GitHub): основные функции.

Раздел 5. Системы отслеживания ошибок и автоматизация тестирования

17. Системы отслеживания ошибок: назначение, преимущества, жизненный цикл дефекта.

18. Критерии качества отчета об ошибке: состав информации и шаги воспроизведения.

19. Принципы тестирования программного обеспечения: пирамида тестирования, TDD.

20. Методологии тестирования: черного, белого и серого ящика.

21. Интеграция систем отслеживания ошибок (на примере GitLab) в процесс разработки.

Раздел 6. Виртуализация

22. Виртуализация: понятие, назначение, преимущества и области применения.

23. Виды виртуализации платформ: полная и частичная эмуляция, паравиртуализация.

24. Виртуализация на уровне операционной системы и приложений.

25. Виртуализация ресурсов: объединение, агрегация, кластеризация.

Раздел 7. Контейнеризация и Docker

26. Контейнеризация: концепция, отличия от виртуализации, преимущества для проектов ИИ.

27. Docker: архитектура, основные понятия (образ, контейнер, Dockerfile, реестр).

28. Рабочий процесс разработки приложений с использованием Docker.

29. Применение Docker для создания воспроизводимых сред выполнения моделей машинного обучения.

30. Особенности применения Docker для инкапсуляции и развертывания моделей машинного обучения и конвейеров данных.

Раздел 8. Непрерывная интеграция и доставка (CI/CD), DevOps

31. Непрерывная интеграция (CI): понятие, назначение, требования к проекту.

32. Непрерывная доставка (CD) и непрерывное развертывание: общая схема и принципы.

33. Преимущества и недостатки внедрения CI/CD в процессе разработки.

34. DevOps: концепция, цели, области применения и преимущества.

35. Взаимодействие практик CI/CD и DevOps в жизненном цикле разработки ПО.

36. Принципы DevOps: автоматизация, мониторинг, обратная связь.