

РАССМОТРЕНО
на заседании методической
комиссии гуманитарных и программно-
вычислительных дисциплин
Председатель МК Строде Т.Н.
Протокол №10 « 14 » 05 2025 г.

УТВЕРЖДАЮ
Зам. директора по учебной работе
И.А. Овчинникова
« 14 » 05 2025 г.

СОГЛАСОВАНО
Генеральный директор ООО «Айти Грэйд»

М.А. Тапцов
« 14 » 05 2025 г.

КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ
для промежуточной аттестации (комплексный дифференцированный зачет)
по УП.01.01 и ПП.01.01 в составе

ПМ.01 Разработка кода для обучения искусственного интеллекта
для специальности

09.02. 13 Интеграция решений с применением технологий искусственного интеллекта
Квалификация: Специалист по работе с искусственным интеллектом

Составители: Котяткина А.Н. – преподаватель СКТ(ф)СПбГУТ
Драницина М.Д. – зав.практикой СКТ(ф)СПбГУТ

Комплексный дифференцированный зачет является промежуточной формой контроля, подводит итог освоения УП.01.01, ПП.01.01 в 4 семестре.

В результате освоения практик студент должен освоить следующие компетенции:

Код	Наименование видов деятельности и профессиональных компетенций
ПК 1.1.	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.
ПК 1.2.	Разрабатывать программные модули в соответствии с техническим заданием.
ПК 1.3.	Оформлять программный код в соответствии с техническим заданием.
ПК 1.4.	Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки.
ПК 1.5.	Выполнять отладку программных модулей с использованием специализированных программных средств.
ПК 1.6.	Выполнять тестирование программного кода.
ПК 1.7.	Составлять тестовые сценарии.

С целью овладения профессиональными компетенциями студент в ходе освоения учебной и производственной практик должен:

Владеть навыками:	Уметь:	Знать:
Н.1. Разработки,	У.1. Анализировать	З.1. Основные методы и

<p>оптимизации и тестирования алгоритмов для ИИ программ.</p> <p>Н.2. Использование библиотек и инструментов для работы с алгоритмами и данными (Pandas, NumPy, ScikitУ. 1.learn);</p> <p>Н.3. Применения структур данных (деревья, графы, списки) для реализации алгоритмов;</p> <p>Н.4. Разработки модульных ИИ-систем, соответствующих требованиям производительности и безопасности;</p> <p>Н.5. Внедрения разработанных ИИ-модулей в комплексные программные системы;</p> <p>Н.6. Оптимизации кода и работы с интерфейсами для взаимодействия между модулями;</p> <p>Н.7. Оформления, документирования и структурирования кода для последующей поддержки;</p> <p>Н.8. Использования инструментов статического анализа кода для выявления ошибок и улучшения качества;</p> <p>Н.9. Работы с системами документирования кода (например,Doxygen, Sphinx) с использованием Git для организации командной работы;</p> <p>Н.10. Управления проектами с использованием Git для организации командной работы</p> <p>Н.11. Разрешения конфликтов при слиянии веток и использования</p>	<p>технические задания и выявлять требования к алгоритмам;</p> <p>У.2. Применять методы алгоритмизации для решения задач программирования;</p> <p>У.3. Разрабатывать оптимальные алгоритмы для решения задач в области ИИ;</p> <p>У.4. Реализовывать программные модули на основе требований технического задания;</p> <p>У.5. Писать чистый, понятный и поддерживаемый код;</p> <p>У.6. Использовать стандартные библиотеки и фреймворки для ускорения разработки;</p> <p>У.7. Оформлять код в соответствии с принятыми стандартами и требованиями;</p> <p>У.8. Документировать разработанный программный код;</p> <p>У.9. Применять соглашения о наименованиях переменных, функций и классов (например, PEP8 для Python);</p> <p>У.10. Работать с системами контроля версий для управления проектами (Git, GitLab);</p> <p>У.11. Организовывать совместную работу над проектом через ветки разработки и слияние изменений;</p> <p>У.12. Разрешать конфликты при слиянии кода;</p> <p>У.13. Использовать инструменты для отладки программного кода;</p> <p>У.14. Идентифицировать и исправлять ошибки в</p>	<p>подходы к построению алгоритмов (жадные алгоритмы, динамическое программирование, рекурсивные подходы);</p> <p>3.2. Принципы эффективной обработки данных;</p> <p>3.3. Языки программирования, применяемые для разработки алгоритмов (Python, C#, Java);</p> <p>3.4. Принципы модульного программирования;</p> <p>3.5. Языки программирования для разработки модулей (Python, C#, Java);</p> <p>3.6. Стандартные фреймворки и библиотеки для работы с ИИ (TensorFlow, PyTorch, Keras);</p> <p>3.7. Основные принципы чистого кода (CleanCode);</p> <p>3.8. Стандарты и практики документирования программного обеспечения;</p> <p>3.9. Инструменты для автоматической проверки качества кода (например, PyLint, ESLint);</p> <p>3.10. Принципы работы распределенных систем контроля версий;</p> <p>3.11. Основные команды и операции в Git (commit, pull, push, merge);</p> <p>3.12. Методы разрешения конфликтов в ходе групповой разработки;</p> <p>3.13. Принципы работы отладчиков и логирования;</p> <p>3.14. Способы выявления ошибок в программе (отладка по шагам, точки останова);</p> <p>3.15. Инструменты для</p>
--	--	--

<p>pullrequest для рецензирования кода;</p> <p>Н.12. Настройки процессов CI/CD для автоматического тестирования и развертывания кода;</p> <p>Н.13. Отладки программных модулей с использованием пошаговой проверки;</p> <p>Н.14. Применения методов логирования и профилирования производительности;</p> <p>Н.15. Использование специальных средств для отладки многопоточных программ;</p> <p>Н.16. Написания юнит-тестов для проверок отдельных функций и модулей;</p> <p>Н.17. Создания автоматизированных тестов для интеграционных проверок;</p> <p>Н.18. Работы с CI/CD пайплайнами для автоматизации тестирования;</p> <p>Н.19. Проектирования тестовых сценариев, включая пограничные и негативные сценарии;</p> <p>Н.20. Использования шаблонов для написания тест-кейсов;</p> <p>Н.21. Автоматизации создания и выполнения тестовых сценариев.</p>	<p>программе;</p> <p>У.15. Применять методы логирования для анализа выполнения программ;</p> <p>У.16. Проводить различные виды тестирования (юнит тестирование, интеграционное тестирование);</p> <p>У.17. Разрабатывать тестовые сценарии для проверки корректности работы программных модулей;</p> <p>У.18. Автоматизировать тестирование программного обеспечения;</p> <p>У.19. Определять критические сценарии работы системы, которые необходимо протестировать;</p> <p>У.20. Разрабатывать пошаговые тестовые сценарии на основе требований;</p> <p>У.21. Оценивать покрытие тестов и их соответствие техническому заданию.</p>	<p>отладки кода (например, PyCharm, Visual Studio Debugger);</p> <p>3.16. Принципы тестирования программного обеспечения;</p> <p>3.17. Методы и подходы к написанию тестов (Test-Driven Development, Behavior-Driven Development);</p> <p>3.18. Инструменты для тестирования программного кода (PyTest, JUnit, Selenium);</p> <p>3.19. Основы тест-дизайна и методы разработки тестовых сценариев;</p> <p>3.20. Принципы проектирования сценариев для функционального и нефункционального тестирования;</p> <p>3.21. Методы составления тест-кейсов для разных типов тестирования.</p>
--	---	--

Комплексный дифференцированный зачет по УП.01.01 и ПП.01.01 проводится на основе тестирования по учебной практике, а также предоставленных документов по производственной практике: отчета по производственной практике в соответствии с требованиями оформления, дневника по практике, положительной характеристики работодателя и заполненного аттестационного листа.

Шкала перевода баллов в оценки:

Оценка результатов КДЗ	Количество баллов		
	УП.01.01 (тест)	ПП.01.01 (аттестационный лист)	По отчету
«5» (отлично)	5	7	1
«4» (хорошо)	4	7	1
«3» (удовлетворительно)	3	7	1
«2» (неудовлетворительно)	2 5 4 3	Менее 7	0 или 1

К тестированию допускается студент, выполнивший и защитивший все практические задания по программе УП.

Тест по учебной практике содержит 25 из 80 вопросов (суммарно вопросов с выбором ответов и теоретических вопросов с кратким ответом), выбираемых случайным образом программой из блоков заданий. ПК.1.1- ПК1.7 - 50 вопросов (25 с выбором ответа, 25 с кратким ответом).

Время тестирования – 60 минут (по 2 минуты на вопрос с выбором ответа, 3 минуты на вопрос с ответом).

Шкала оценивания образовательных результатов тестирования:

критерии	Кол-во баллов по тестированию
получают студенты, справившиеся с работой 100-90%	5 баллов
получают студенты, справившиеся с работой 89-76%	4 балла
получают студенты, справившиеся с работой 60-75%	3 балла
менее 60% правильных ответов	От 0 до 2 баллов

Первый блок заданий – вопросы с выбором ответа:

Проверяемые ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4

- 1) Прочитайте текст и выберите один правильный ответ.
Какой документ определяет перечень функций разрабатываемого модуля?
 1. Устав компании
 2. Техническое задание
 3. Фирменный стиль
 4. Слоган

- 2) Прочитайте текст и выберите один правильный ответ.
Что важно сделать первым при проектировании алгоритма?
 1. Выбрать стиль оформления кода
 2. Определить основные действия и их последовательность
 3. Нарисовать фирменный логотип
 4. Выполнить проверку орфографии в тексте задачи

- 3) Прочитайте текст и выберите один правильный ответ.
Какой вид диаграмм часто используют для визуализации алгоритмов?

1. Диаграммы Венна
2. Блок-схемы
3. Сетевые графы
4. Инфографика

4) Прочитайте текст и выберите один правильный ответ.

Что характеризует ветвящийся алгоритм?

1. Ветвление
2. Только циклы
3. Перепутанные шаги
4. Наличие условий, влияющих на путь выполнения

5) Прочитайте текст и выберите один правильный ответ.

Базовой конструкцией для повторяющихся операций в алгоритме является:

1. Переменная
2. Ветвление
3. Кнопка
4. Цикл

6) Прочитайте текст и выберите один правильный ответ.

Почему важно прописывать алгоритм до написания кода?

1. Чтобы заранее определить структуру решения
2. Чтобы согласовать подход с командой
3. Чтобы упростить поиск ошибок в будущем
4. Для понимания логики решения задачи

7) Прочитайте текст и выберите один правильный ответ.

Какой шаг выполняется после анализа задачи?

1. Финальное тестирование
2. Реализация модуля
3. Проектирование алгоритма
4. Сборка приложения

8) Прочитайте текст и выберите один правильный ответ.

Как обеспечить понятность алгоритма?

1. Делить на отдельные простые шаги
2. Использовать длинные предложения
3. Оформлять одним абзацем
4. Прячь детали

9) Прочитайте текст и выберите один правильный ответ.

В каком случае алгоритм можно считать детализированным?

1. Каждый шаг легко реализовать в коде
2. Если он занимает больше страницы
3. Написан без пробелов
4. Содержит сложные термины

10) Прочитайте текст и выберите один правильный ответ.

Алгоритм должен приводить к:

1. Четкому, заранее определенному результату
2. Неоднозначным результатам
3. Непредсказуемому поведению

4. Бесконечным циклам

11) Прочитайте текст и выберите один правильный ответ.

Какие требования считаются ключевыми при разработке программного модуля?

1. Особенности корпоративного дизайна
2. Использование технического задания
3. Мода на современные технологии
4. Региональные климатические условия

12) Прочитайте текст и выберите один правильный ответ.

Как рационально организовать командную работу над проектом?

1. Разделить проект на отдельные модули
2. Основываться на цветовой схеме интерфейса
3. Вести работу по принципу территориального расположения
4. Распределять по страницам документации

13) Прочитайте текст и выберите один правильный ответ.

Что следует сделать, если найдено неоднозначное требование в техническом задании?

1. Принять самостоятельное решение
2. Согласовать детали с заказчиком или аналитиком
3. Оставить пункт без внимания
4. Отклонить задачу

14) Прочитайте текст и выберите один правильный ответ.

Где обычно указывается описание интерфейса программного модуля?

1. В разделах технического задания или спецификации
2. В аппаратной документации
3. В личных предпочтениях разработчика
4. В отчетах по тестированию

15) Прочитайте текст и выберите один правильный ответ.

На что следует обратить особое внимание при написании программных модулей?

1. Разнообразие используемых технологий
2. Исключительное отсутствие комментариев
3. Постоянное изменение архитектуры проекта
4. Соблюдение стандартов кодирования

16) Прочитайте текст и выберите один правильный ответ.

Какое главное преимущество даёт модульная структура программы?

1. Увеличение сложности системы
2. Упрощение поиска и исправления ошибок, а также поддержка кода
3. Снятие ограничения на архитектуру
4. Повышение визуальной привлекательности

17) Прочитайте текст и выберите один правильный ответ.

Что должно входить в конечный результат работы над модулем?

1. Программный код, тесты и сопроводительная документация
2. Только финальная презентация
3. Изображение пользовательского интерфейса
4. Внутренняя справка

18) Прочитайте текст и выберите один правильный ответ.

По какому критерию оценивается успешная реализация программного модуля?

1. Соответствует ли он требованиям технического задания и заявленной функциональности
2. По объему исходного кода
3. По числу современных фреймворков
4. По уникальности дизайнерских решений

19) Прочитайте текст и выберите один правильный ответ.

Что важно проверять при разработке программного модуля?

1. Исключительно длину переменных
2. Корректность структуры кода относительно архитектуры проекта
3. Только быстрое действие отдельных операций
4. Размер итогового файла

20) Прочитайте текст и выберите один правильный ответ.

Какой способ наиболее эффективен для организации совместной работы над модулем?

1. Использование системы контроля версий
2. Разработка каждым сотрудником в отдельном файле
3. Ведение записей на бумаге
4. Проведение устных обсуждений задач

21) Прочитайте текст и выберите один правильный ответ.

Какое условие способствует выполнению стандартов оформления программного кода?

1. Игнорирование утвержденных соглашений
2. Случайный выбор стиля написания
3. Наличие единых правил кодирования для всей команды
4. Безразличие к оформлению кода

22) Прочитайте текст и выберите один правильный ответ.

Каким должно быть имя переменной для обеспечения читаемости кода?

1. Случайным набором символов
2. Смысловым и отражающим назначение
3. Взятым из нерелевантного источника
4. Состоящим из малопонятных сокращений

23) Прочитайте текст и выберите один правильный ответ.

Какова основная функция комментариев в программном коде?

1. Объяснять логику работы и условия выполнения кода
2. Использоваться только для украшения проекта
3. Исключительно для увлечения объема файла
4. Для подготовки к печати

24) Прочитайте текст и выберите один правильный ответ.

Как минимизировать конфликты оформления в работе большой команды?

1. Применять согласованный стиль написания кода
2. Опирайтесь на индивидуальные предпочтения без осуждения
3. Игнорировать внедрение автоматизированных инструментов
4. Преднамеренно избегать стандартов

25) Прочитайте текст и выберите один правильный ответ.

Какой из ниже перечисленных элементов не относится к оформлению исходного кода?

1. Выравнивание и отступы

2. Комментарии к функциям
3. Процесс сборки и компиляции проекта
4. Принципы именования переменных

Проверяемые ПК 1.5, ПК 1.6, ПК 1.7

1) Прочитайте текст и выберите один правильный ответ.

Как называется специальная точка, в которой выполнение программы прерывается для анализа?

1. Точка останова (breakpoint)
2. Маркер
3. Ограничение
4. Принципы именования переменных

2) Прочитайте текст и выберите один правильный ответ.

Какой инструмент предназначен для обнаружения ошибок в процессе исполнения кода?

1. Веб-браузер
2. Отладчик
3. Редактор электронных таблиц
4. Программа для обработки изображений

3) Прочитайте текст и выберите один правильный ответ.

Как называется процесс последовательного выполнения кода по одной строке?

1. Быстрый запуск
2. Автоматическое выполнение
3. Пошаговое выполнение
4. Пакетная обработка

4) Прочитайте текст и выберите один правильный ответ.

Какие ошибки выявляются только при запуске программы?

1. Исполнительные (run-time)
2. Административные
3. Синтаксические
4. Орфографические

5) Прочитайте текст и выберите один правильный ответ.

Какой метод помогает анализировать значения переменных в процессе отладки?

1. Использование функций вывода (print, console.log)
2. Сравнение с документацией продукта
3. Активация внешних носителей
4. Чтение аппаратных спецификаций

6) Прочитайте текст и выберите один правильный ответ.

Как называется процесс вывода отладочной информации в специальном окне или терминале?

1. Архивирование
2. Логирование
3. Импорт данных
4. Ввод логов вручную

7) Прочитайте текст и выберите один правильный ответ.

Почему крайне важно устранять все выявленные ошибки при отладке?

1. Для увеличения количества задач в проекте

2. Исключительно ради интереса
3. Для расширения доступа к проекту
4. Для корректной и стабильной работы приложения

8) Прочитайте текст и выберите один правильный ответ.

Какой способ помогает понять причину возникновения программного исключения?

1. Анализ стека вызовов
2. Просмотр свойств файловой системы
3. Изменение темы оболочки
4. Создание снимков экрана

9) Прочитайте текст и выберите один правильный ответ.

В каких случаях используются условные точки останова?

1. Всегда при запуске программы
2. Исключительно при работе на устаревшем оборудовании
3. Для проверки выполнения кода при определённых значениях переменных
4. Только при внешних ошибках

10) Прочитайте текст и выберите один правильный ответ.

Что отладчик позволяет эффективно выявлять в процессе работы над программой?

1. Проблемы с дизайном интерфейса
2. Логические ошибки в работе программного кода
3. Новых членов команды
4. Недостатки аппаратной платформы

11) Прочитайте текст и выберите один правильный ответ.

Как называется процесс проверки работоспособности отдельных компонентов программы?

1. Модульное (юнит-) тестирование
2. Интеграционное тестирование
3. Логический анализ
4. Системное тестирование

12) Прочитайте текст и выберите один правильный ответ.

Какой подход охватывает взаимодействие нескольких программных модулей в одном тесте?

1. Приемочное тестирование
2. Интеграционное тестирование
3. Модульное (юнит-) тестирование
4. Системное тестирование

13) Прочитайте текст и выберите один правильный ответ.

Что такое тест-кейс в тестировании программного продукта?

1. Описание условия проведения теста и ожидаемый результат
2. Уникальное имя файла теста
3. Общий список всех модулей системы
4. Документ, описывающий визуальный дизайн программы

14) Прочитайте текст и выберите один правильный ответ.

Какой из перечисленных признаков НЕ характеризует качественный тестовый сценарий?

1. Четкость формулировок
2. Однозначность требований
3. Объем не менее 10 страниц

4. Полное покрытие тестируемой функциональности

15) Прочитайте текст и выберите один правильный ответ.

Какой основной элемент обязательно должен содержаться в тестовом сценарии?

1. Имя ответственного тестировщика
2. Последовательность шагов и ожидаемый результат
3. Иллюстрации или схемы
4. Звуковое сопровождение сценария

Второй блок заданий – вопросы открытого типа:

Проверяемые ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4

1) Прочитайте текст. Дайте краткий ответ.

Объясните, почему важно сначала анализировать техническое задание перед построением алгоритма.

2) Прочитайте текст. Дайте краткий ответ.

Опишите процесс разбивки задачи на шаги при формировании алгоритма.

3) Прочитайте текст. Дайте краткий ответ.

Приведите пример составления блок-схемы для простой задачи.

4) Прочитайте текст. Дайте краткий ответ.

Какие основные ошибки допускают при формировании алгоритмов? Как их избежать?

5) Прочитайте текст. Дайте краткий ответ.

Что следует делать, если в процессе построения алгоритма обнаружены неоднозначности в требованиях заказчика?

6) Прочитайте текст. Дайте краткий ответ.

Расскажите о важности выбора оптимальных структур данных при формировании алгоритма.

7) Прочитайте текст. Дайте краткий ответ.

Опишите, как документировать алгоритм для команды разработки.

8) Прочитайте текст. Дайте краткий ответ.

Как учесть возможные ветвления и повторения (циклы) в алгоритме?

9) Прочитайте текст. Дайте краткий ответ.

Опишите последовательность разработки программного модуля на примере реальной задачи.

10) Прочитайте текст. Дайте краткий ответ.

Какие критерии указывают на то, что модуль полностью соответствует техническому заданию?

11) Прочитайте текст. Дайте краткий ответ.

Как происходит взаимодействие между модулями в сложной программе? Приведите пример.

12) Прочитайте текст. Дайте краткий ответ.

Расскажите о процессе интеграции разработанных модулей в общий проект.

13) Прочитайте текст. Дайте краткий ответ.

Как вы тестируете работоспособность вновь написанного модуля?

14) Прочитайте текст. Дайте краткий ответ.

Какие методы помогают находить и устранять ошибки в разработанных модулях?

15) Прочитайте текст. Дайте краткий ответ.

Опишите случаи, когда требуется доработка модуля после тестирования.

- 16) Прочитайте текст. Дайте краткий ответ.
Почему важно соблюдать стандарты оформления кода в командной разработке?
- 17) Прочитайте текст. Дайте краткий ответ.
Расскажите о правилах именования переменных и функций в современном программировании.
- 18) Прочитайте текст. Дайте краткий ответ.
Какие инструменты помогают поддерживать оформление кода в едином стиле?
- 19) Прочитайте текст. Дайте краткий ответ.
Как использование комментариев влияет на поддержку и развитие проекта?
- 20) Прочитайте текст. Дайте краткий ответ.
Опишите преимущества использования линтеров и автоформатеров в работе над кодом.
- 21) Прочитайте текст. Дайте краткий ответ.
Какие негативные последствия могут возникнуть при пренебрежении оформлением кода?
- 22) Прочитайте текст. Дайте краткий ответ.
Для чего необходима система контроля версий в современных ИТ-проектах?
- 23) Прочитайте текст. Дайте краткий ответ.
Опишите процедуру добавления нового файла в репозиторий с помощью системы контроля версий.
- 24) Прочитайте текст. Дайте краткий ответ.
Какие трудности могут возникнуть при одновременной работе нескольких участников над одним и тем же файлом? Как их решать?
- 25) Прочитайте текст. Дайте краткий ответ.
Как с помощью системы контроля версий отслеживать историю изменений и возвращаться к прошлым версиям кода?

Проверяемые ПК 1.5, ПК 1.6, ПК 1.7

- 1) Прочитайте текст. Дайте краткий ответ.
Что такое отладка программного модуля и зачем она нужна?
- 2) Прочитайте текст. Дайте краткий ответ.
Какие этапы включает процесс отладки программного кода?
- 3) Прочитайте текст. Дайте краткий ответ.
Какие виды ошибок чаще всего выявляются с помощью отладчика?
- 4) Прочитайте текст. Дайте краткий ответ.
Чем отличается пошаговое выполнение программы от установки точек останова (breakpoints)?
- 5) Прочитайте текст. Дайте краткий ответ.
Для каких задач подойдет встроенный отладчик в IDE?
- 6) Прочитайте текст. Дайте краткий ответ.
Что такое модульное тестирование и как оно проводится?
- 7) Прочитайте текст. Дайте краткий ответ.
Какие виды тестирования программного продукта бывают?
- 8) Прочитайте текст. Дайте краткий ответ.
Какие инструменты чаще всего используют для автоматизированного тестирования кода?
- 9) Прочитайте текст. Дайте краткий ответ.
Зачем нужны негативные тесты и приведите пример?

- 10) Прочитайте текст. Дайте краткий ответ.
Какие критерии определяют, что тестирование прошло успешно?
- 11) Прочитайте текст. Дайте краткий ответ.
Что содержит типичный тестовый сценарий?
- 12) Прочитайте текст. Дайте краткий ответ.
Как определить набор функций, для которых нужно создавать тестовые сценарии?
- 13) Прочитайте текст. Дайте краткий ответ.
Приведите пример тестового сценария для функции авторизации пользователя.
- 14) Прочитайте текст. Дайте краткий ответ.
Как учитывать граничные значения в тестовых сценариях?
- 15) Прочитайте текст. Дайте краткий ответ.
Как регулярно обновлять тестовые сценарии?

АТТЕСТАЦИОННЫЙ ЛИСТ ПО УЧЕБНОЙ ПРАКТИКЕ

ФПО

Обучающийся(аяся) на 2 курсе в группе ИИ1026 по специальности СПО

09.02.13 Интеграция решений с применением технологий искусственного интеллекта

код

наименование

успешно прошел(ла) **учебную** практику по профессиональному модулю

ПМ.01 Разработка кода для обучения искусственного интеллекта

наименование профессионального модуля

в объеме 72 часов с «__» _____ 202__ г. по «__» _____ 202__ г. в организации

Смоленский колледж телекоммуникаций (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича»

наименование организации

г. Смоленск, ул. Коммунистическая , д.21

юридический адрес

Виды и качество выполнения работ

<i>Виды и объем работ, выполненных студентом во время практики</i>	<i>Отметка о выполнении</i>
Сбор и предобработка данных из открытых источников для задач машинного обучения. (6 часов)	
Разработка простых программных модулей для анализа данных с использованием библиотек Python(Pandas,NumPy). (6 часов)	
Разработка базовых моделей машинного обучения(линейная регрессия, дерево решений)для реальных задач. (6 часов)	
Визуализация данных и результатов работы моделей ИИ с использованием Matplotlib. (6 часов)	
Интеграция предобученной модели машинного обучения в простое мобильное приложение(AndroidStudio). (6 часов)	
Разработка прототипа мобильного приложения с элементами ИИ (например, распознавание объектов). (6 часов)	
Написание и отладка юнит-тестов для программных модулей, реализованных в ИИ-системах. (12 часов)	
Работа с системами контроля версий(Git,GitHub)для управления проектами. (12 часов)	
Контейнеризация простых ИИ-приложений с использованием Docker. (6 часов)	
Внедрение и отладка CI/CD процессов для автоматизированного тестирования. (6 часов)	
Количество баллов по тестированию	

Характеристика учебной и профессиональной деятельности студента во время учебной практики.

Аттестуемый(ая) продемонстрировал(а) / не продемонстрировал(а) владение профессиональными компетенциями:

ПК 1.1.

Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2.

Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3.

Оформлять программный код в соответствии с техническим заданием.

ПК 1.4.

Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки.

ПК 1.5.

Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.6.

Выполнять тестирование программного кода.

ПК 1.7.

Составлять тестовые сценарии.

Дата «___» _____ 202__ г.

Подпись(и) руководителя(ей) практики

Преподаватель _____

подпись

расшифровка подписи

Заведующий практикой _____

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

СМОЛЕНСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ (ФИЛИАЛ) СПбГУТ
(СКТ(ф)СПбГУТ)

ТЕХНИЧЕСКИЙ ОТЧЕТ

по производственной практике

студента

ФИО

ПМ. 01 Разработка кода для обучения искусственного
интеллекта

по специальности 09.02.13 Интеграция решений с применением
технологий искусственного
интеллекта

г. Смоленск
202_

ТРЕБОВАНИЯ ПО СОСТАВЛЕНИЮ ТЕХНИЧЕСКОГО ОТЧЕТА ПО ПРАКТИКЕ

1. Технический отчет по производственной практике студенты пишут во время прохождения практики в соответствии с графиком учебного процесса.
2. Технический отчет должен быть выполнен на стандартных листах писчей бумаги (ф. А 4), в объеме 10-12 страниц.
3. Перечень вопросов технического отчета следующий:
 - титульный лист
 - программа практики
 - введение
 - 1. Общие сведения о функциях и структуре предприятия (схема структуры предприятия)
 - 2. Описание производственного процесса участка, на котором проходит основной период производственной практики.
 - 3. Индивидуальное задание по ПМ.
 - 4. Организация и состояние охраны труда на предприятии.
 - Список литературы.
 - Приложение (фото, аудио-файлы при их наличии).
4. Технический отчет должен быть оформлен в соответствии с требованиями ГОСТР 2.105-2019 ЕСКД (ред. от 08.11.2023) общие требования к текстовым документам.
 - Текст отчета должен быть выполнен на компьютере с одинаковым межстрочным интервалом (1,5).
 - Отчет выполняется на листах с одной стороны, разборчиво, аккуратно, четко.
 - Текст набирается нежирным шрифтом Times New Roman на стандартных листах 14 шрифтом с соответствующей рамкой, границы которой располагаются следующим образом:
 - расстояние слева от границы листа до рамки – 20мм
 - расстояние сверху, справа и снизу от границы листа до рамки 5 мм
 - Текст каждого листа записи должен иметь следующие поля
 - расстояние слева от текста до рамки 5 мм, справа – 3 мм,
 - расстояние от заголовка, верхней и нижней строки текста до рамки 10 мм,
 - абзацы в тексте начинаются отступом 15 мм.
 - В отчет обязательно должны входить структурные, функциональные схемы.
 - Нумерация страниц обязательна.
5. Технический отчет должен быть проверен и подписан руководителем практики от предприятия и заверен печатью.
6. Технический отчет сдается заведующему практикой от колледжа для получения дифференциального зачета.

Заведующий практикой

Драницина М.Д.

Утверждаю
Зам. директора по учебной работе

Овчинникова И.А.

ПРОГРАММА ПРАКТИКИ

Название МДК	Виды работ в соответствии с рабочими программами МДК	Количество часов
МДК.01.02 Разработка мобильных приложений с поддержкой искусственного интеллекта. МДК.01.03. Тестирование программных модулей.	Состав служб и участков предприятия. Правила внутреннего распорядка. Организация мероприятий по охране труда. Инструктаж по ТБ и охране труда	6
	- Инструктаж по технике безопасности и охране труда. - Проектирование и реализация моделей машинного и глубокого обучения для решения производственных задач (например, классификация изображений или прогнозирование данных). - Оптимизация моделей ИИ для повышения производительности на реальных задачах предприятия. - Разработка и внедрение сложных ИИ-приложений для мобильных платформ с использованием TensorFlowLite или CoreML. - Интеграция разработанных ИИ-модулей в существующие информационные системы предприятия. - Разработка и публикация мобильных приложений с поддержкой ИИ для Android и iOS. - Автоматизация тестирования программных продуктов предприятия с использованием Jenkins и GitLabCI. - Проведение интеграционного тестирования для сложных систем ИИ и их взаимодействие с другими модулями. - Мониторинг производительности ИИ-приложений в реальных условиях эксплуатации. - Разработка и внедрение систем автоматизированного развертывания ИИ-приложений с использованием Docker и Kubernetes.	60
	Оформление дневника, отчета и других документов по практике. Предоставление дневника по практике и других необходимых документов. Публичная защита отчета по практике. Сдача КДЗ	6
	ВСЕГО:	72

Индивидуальное задание (1-2 вопроса практического характера, составляются преподавателями данного ПМ):

- 1.
- 2.

Председатель методической
комиссии

АТТЕСТАЦИОННЫЙ ЛИСТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

ФИО

Обучающийся (аяся) на ____ курсе в группе ИИ1026 по специальности СПО

09.02.13. Интеграция решений с применением технологий искусственного интеллекта

успешно прошел(ла) **производственную** практику по профессиональному модулю

ПМ.01 Разработка кода для обучения искусственного интеллекта

в объеме 72 часа с « » 202 г. по « » 202 г.

в организации

юридический адрес

Виды работ, выполненных студентом во время практики :

Выполнял работу по проектированию и реализации моделей машинного и глубокого обучения для решения производственных задач

Выполнял работу по оптимизации моделей ИИ для повышения производительности на реальных задачах предприятия.

Выполнял работу по разработке и внедрению сложных ИИ-приложений для мобильных платформ с использованием TensorFlow Lite или CoreML.

Выполнял работу по интеграции разработанных ИИ-модулей в существующие информационные системы предприятия

Выполнял работу по разработке и публикации мобильных приложений с поддержкой ИИ для Android и iOS

Выполнял работу по автоматизации тестирования программных продуктов предприятия с использованием Jenkins и GitLabCI

Выполнял работу по проведению интеграционного тестирования для сложных систем ИИ и их взаимодействие с другими модулями

Выполнял работу по мониторингу производительности ИИ-приложений в реальных условиях эксплуатации

Выполнял работу по разработке и внедрению систем автоматизированного развертывания ИИ-приложений с использованием Docker и Kubernetes

Работал с документами отраслевой направленности

Выполнял: оформление дневника, отчета и других документов по практике.
Предоставление дневника по практике и других необходимых документов.
Публичная защита отчета по практике.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

СМОЛЕНСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ (ФИЛИАЛ) СПбГУТ
(СКТ(ф)СПбГУТ)

В Е Д О М О С Т Ь
20__/20__ учебный год

УП.01 Учебная практика

ПП.01 Производственная практика (по профилю специальности)

ПМ.01 Разработка кода для обучения искусственного интеллекта

Курс _____ группа _____

Специальность **09.02.13 Интеграция решений с применением технологий искусственного интеллекта**

Преподаватель _____
(фамилия, имя, отчество)

№№ пп	ФИО студента	Кол-во баллов по УП.01	Кол-во баллов по ПП.01 (аттестационный лист, дневник, положительное заключение работодателя)	Кол-во баллов по отчету по ПП.01	Оценка результата КДЗ
1.					
2.					
3.					
4.					
5.					
6.					
.....					

Преподаватель

подпись

ФИО

Зав.практикой

подпись

М.Д.Драницина

ФИО

« ____ » _____ 202__ г.

Шкала оценивания образовательных результатов тестирования:

критерии	Кол-во баллов по тестированию
получают студенты, справившиеся с работой 100-90%;	5 баллов
получают студенты, справившиеся с работой 89-76%	4 балла
получают студенты, справившиеся с работой 60-75%	3 балла
менее 60% правильных ответов	От 0 до 2 баллов

Шкала перевода баллов в оценки:

Оценка результатов КДЗ	Количество баллов		
	УП.01	ПП.01 <i>(аттестационный лист, дневник, положительное заключение работодателя)</i>	ПП01 (отчет по практике)
«5» (отлично)	5	7	1
«4» (хорошо)	4	7	1
«3» (удовлетворительно)	3	7	1
«2» (неудовлетворительно)	5 4 3 2 1	Менее 7	0 или 1

**Эталоны ответов к заданиям
Блок заданий № 1 закрытого типа
Проверяемые ПК.1.1 – ПК.1.4**

Вопрос № пп	Правильный ответ
1.	2
2.	2
3.	2
4.	4
5.	4
6.	4
7.	3
8.	1
9.	1
10.	1
11.	2
12.	1
13.	2
14.	1
15.	4
16.	2
17.	1
18.	1
19.	2
20.	1
21.	3
22.	2
23.	1
24.	1
25.	3

Проверяемые ПК.1.5 – ПК.1.7

Вопрос № пп	Правильный ответ
1.	1
2.	2
3.	3
4.	1
5.	1
6.	2
7.	4
8.	1
9.	3
10.	2
11.	1
12.	2
13.	1
14.	3
15.	2

Ответы на вопросы блока заданий № 2 открытого типа

Проверяемые ПК.1.1 – ПК.1.4

1. Ответ: Анализ ТЗ перед построением алгоритма необходим для точного понимания требований, целей и ограничений проекта. Это позволяет грамотно спланировать логику, минимизировать ошибки и избежать неопределённости на следующих этапах разработки.
2. Ответ: Процесс декомпозиции задачи начинается с выделения главных этапов решения, затем каждый этап разбивается на последовательные действия до достижения минимально простых операций, реализуемых в коде. Это визуализируется с помощью текстовых описаний или блок-схем.
3. Ответ: Пример — для задачи “нахождение большего из двух чисел”:
— Ввод чисел A и B;
— Сравнение: если $A > B$, вывести A, иначе вывести B;
— Завершить.
На блок-схеме это отразится как: `старт` → `ввод` → `проверка` → `условный переход` → `вывод` → `конец`.
4. Ответ: Распространённые ошибки: некорректные условия ветвления, отсутствие учёта крайних случаев, невозможность завершения (бесконечные циклы) и неоднозначные шаги. Профилактика — последовательная детализация, тестирование на крайних сценариях и привлечение коллег к ревью алгоритма.
5. Ответ: Неоднозначности следует сразу выявлять и уточнять у заказчика или бизнес-аналитика, фиксировать дополнительные требования в документации и в дальнейшем строго им следовать.
6. Ответ: Структуры данных определяют организацию и скорость обработки информации. Грамотный выбор структур повышает эффективность алгоритма и облегчает его дальнейшую поддержку и расширение.
7. Ответ: Документирование происходит в виде подробных текстовых описаний, приложений блок-схем и указания выбранных структур данных. Оформление согласуется с внутренними стандартами команды, что облегчает понимание алгоритма всеми участниками.
8. Ответ: В алгоритме ветвления прописываются условия переходов для всех сценариев, а для циклов чётко указывается начальное условие, параметры повторения и критерии выхода. Это гарантирует корректную обработку всех возможных ситуаций.
9. Ответ: Разработка модуля начинается с изучения ТЗ, далее проектируется структура, реализуются функции, затем проводится модульное тестирование. После исправления ошибок создаётся документация, и модуль интегрируется в общий проект.
10. Ответ: Модуль полностью соответствует ТЗ, если реализует все функции и интерфейсы, соблюдает форматы входных/выходных данных и успешно проходит проверки и тест-кейсы, указанные в требованиях.
11. Ответ: Взаимодействие между модулями реализуется через четко описанные интерфейсы: например, один модуль предоставляет API, а другой взаимодействует с ним через определённые методы для передачи или получения данных.
12. Ответ: Интеграция модулей начинается после их индивидуального тестирования: модули объединяют через интерфейсы, проводят интеграционное тестирование и при необходимости — доработку для обеспечения совместимости.
13. Ответ: Модуль тестируется с помощью юнит-тестов для всех функций, далее — ручная проверка на граничных и типичных данных, после чего результат анализируется на соответствие ожидаемому поведению.
14. Ответ: Поиск и исправление ошибок производится благодаря юнит-тестам, логированию, анализу журналов выполнения и отладчику.
15. Ответ: Доработка требуется, если после тестирования обнаружены недостатки реализации, невыполнение требований ТЗ, выявлены баги при интеграции или изменились функциональные запросы заказчика.
16. Ответ: Соблюдение стандартов оформления в коллективной разработке важно для унификации стиля, быстрого понимания чужого кода, облегчения поддержки и снижения количества ошибок, связанных с неоднозначной реализацией.
17. Ответ: Имена переменных и функций должны быть осмысленными, отображать назначение, писаться в едином стиле (например, `camelCase` или `snake_case`), не содержать лишних сокращений и придерживаться выбранного языка, чаще всего — английского.
18. Ответ: Поддержанию единого стиля оформления способствуют линтеры, автоформатеры (например, ESLint, Prettier, PyLint), а также шаблоны и плагины IDE, которые автоматически исправляют отступы, рекомендации и стиль кода.

19. Ответ: Комментарии делают код более прозрачным для коллег, помогают быстрее разбираться в логике и алгоритмах, существенно снижают временные издержки при поддержке и доработках программного продукта.
20. Ответ: Линтеры и автоформатеры позволяют автоматически выявлять нарушения стиля, предотвращать распространённые ошибки и сокращать время на ручную правку оформления, что особенно важно при командной работе и CI/CD.
21. Ответ: Игнорирование оформления приводит к ухудшению читаемости, увеличению вероятности ошибок, росту времени на ревью и доработку, а также к затруднениям при внедрении новых специалистов в проект.
22. Ответ: Система контроля версий (например, Git) используются для отслеживания всех изменений в коде, организации совместной работы, обеспечения возможности возврата к прошлым версиям и быстрого обнаружения причин возникновения ошибок.
23. Ответ: Файл сначала создаётся в проекте, затем добавляется в индекс командой (`git add <имя_файла>`), фиксируется коммитом (`git commit -m "Комментарий"`), после чего передаётся в общий репозиторий через (`git push`).
24. Ответ: При параллельной работе возможны конфликты изменений в одних и тех же строках файлов. Решают их через ручное слияние, обсуждение изменений на этапе pull request, настройку политики блокировки веток и регулярную синхронизацию с основной веткой.
25. Ответ: Контроль версий позволяет просматривать историю изменений, разницу между коммитами, восстанавливать прежние состояния с помощью команд (`git log`, `git diff`, `git checkout`, `git revert`), тем самым возвращаться к работоспособным версиям проекта.

Проверяемые ПК.1.5 – ПК.1.7

1. Ответ: Отладка — это процесс выявления и устранения ошибок в программном коде с помощью специальных инструментов (отладчиков). Она нужна для повышения надежности и корректной работы программы.
2. Ответ: Выделяют поиск ошибки, анализ причин, локализацию участка сбоя, исправление и повторную проверку.
3. Ответ: Логические, синтаксические, ошибки времени исполнения, неверные значения переменных.
4. Ответ: Пошаговое выполнение позволяет исполнять код по одной команде, а точки останова останавливают выполнение при достижении определенной строки кода.
5. Ответ: Для наблюдения за переменными, анализа стека вызовов, трассировки выполнения и локализации ошибок.
6. Ответ: Это проверка отдельных частей программы (модулей/функций) отдельно друг от друга с помощью специальных тестов.
7. Ответ: Модульное, интеграционное, системное, приемочное.
8. Ответ: JUnit, PyTest, Selenium и аналогичные зависимости для разных языков программирования.
9. Ответ: Для проверки, как программа обрабатывает неверные данные. Например, попытка деления на ноль или ввод некорректного email.
10. Ответ: Все запланированные тесты выполнены, обнаруженные ошибки исправлены, тестовые сценарии больше не выявляют новых проблем.
11. Ответ: Цель теста, шаги выполнения, необходимые исходные данные, ожидаемый результат, фактический результат.
12. Ответ: По требованиям к ПО, документации и анализу бизнес-логики системы.
13. Ответ: Вход: правильные логин и пароль. Шаги: ввести данные, нажать “Войти”. Ожидание: пользователь попадает в личный кабинет.
14. Ответ: Нужно создавать сценарии с минимальными, максимальными и средними значениями, а также за пределами допустимого диапазона.
15. Ответ: Внести пункт проверки сценариев при любом изменении требований или кодовой базы, а также проводить ревизию перед крупными релизами.