

УТВЕРЖДАЮ  
Проректор по учебной работе и  
качеству образования

\_\_\_\_\_ И. А. Долгова

30 сентября 2026

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

**Б1.О.10**

**Программирование в  
гуманитарной сфере**

---

|                         |                                                        |
|-------------------------|--------------------------------------------------------|
| Направление подготовки: | 45.04.04 Интеллектуальные системы в гуманитарной среде |
| Профиль подготовки:     | Цифровая лингвистика                                   |
| Квалификация:           | магистр                                                |
| Форма обучения:         | очная                                                  |
| Год начала подготовки:  | 2027                                                   |

Рабочая программа составлена в соответствии с:

- федеральным государственным образовательным стандартом высшего образования – магистратура по направлению подготовки 45.04.04 Интеллектуальные системы в гуманитарной среде, утверждённым приказом Министерства науки и высшего образования Российской Федерации от 13 августа 2020 года № 1003.

Разработчики программы: Макаров А. А., доктор технических наук, профессор; Кириллов А. Г., кандидат филологических наук, доцент

Рабочая программа согласована с руководителем образовательной программы 45.04.04 Интеллектуальные системы в гуманитарной среде. Рабочая программа рассмотрена и одобрена на заседании кафедры информационных систем и компьютерных технологий 10 сентября 2026, протокол № 2.

# 1. ОБЩАЯ ХАРАКТЕРИСТИКА

## 1.1. Цель и задачи дисциплины

**Цель** изучения дисциплины — формирование у обучающихся углубленной системы знаний и практических навыков проектирования, реализации и документирования прикладного программного обеспечения, предназначенного для автоматизированного сбора, извлечения и структурирования гуманитарных и текстовых данных в цифровой среде.

**Задачи** изучения дисциплины:

- изучение архитектуры клиент-серверного взаимодействия, принципов работы сетевых протоколов и открытых веб-интерфейсов (API) применительно к задачам цифровой лингвистики;
- формирование практических навыков разработки прикладных программных скриптов (парсеров, краулеров) для автоматического сбора неструктурированных текстовых массивов с веб-страниц и платформ социальных медиа;
- освоение технологий коллективной разработки программного обеспечения и контроля версий (Git/GitHub), правил составления сопутствующей технической документации и спецификаций проекта;
- развитие умений проектировать простейшие базы данных и автоматизированные модули предварительной обработки языкового материала для последующего интеллектуального анализа.

## 1.2. Требования подготовленности обучающегося к освоению содержания дисциплины

Освоение дисциплины «Программирование в гуманитарной сфере» в 3 семестре базируется на устойчивом алгоритмическом фундаменте, заложенном во 2 семестре в рамках курса «Язык программирования Python», а также на базовых математических знаниях («Математика в гуманитарных науках»).

Обучающийся перед началом освоения дисциплины должен:

**Знать:**

- базовый синтаксис языка Python, встроенные типы и структуры данных (списки, строки, словари, множества);
- правила создания пользовательских функций, области видимости переменных и стандарт оформления кода PEP 8;
- синтаксис регулярных выражений для поиска и фильтрации текстовых паттернов.

**Уметь:**

- разрабатывать, тестировать и отлаживать локальные скрипты на Python в среде Jupyter Notebook или VS Code;
- выполнять чтение и запись текстовых файлов распространенных форматов (TXT, CSV, JSON);
- использовать базовые логические операции для построения ветвящихся алгоритмов.

**Владеть:**

- навыками работы в операционных системах на уровне уверенного пользователя и администратора локального ПК;
- навыками документирования написанного программного кода с использованием docstrings и комментариев.

## 1.3. Планируемые результаты обучения по дисциплине

| Код и наименование компетенции                                                                                                              | Код и наименование индикатора достижения компетенции                                                                                           | Результаты обучения по дисциплине                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ОПК-6. Способен осваивать, применять и разрабатывать документацию к программным системам в области программирования и информационных систем | ОПК-6.1. Осваивает техническую документацию, стандарты кодирования и требования к архитектуре современных информационных и программных систем  | ОПК-6.1.2. Знает архитектуру клиент-серверного взаимодействия, принципы работы веб-сервисов (API) и правила составления технических заданий на разработку лингвистического ПО |
|                                                                                                                                             | ОПК-6.2. Применяет языки программирования и инструменты веб-технологий для разработки элементов информационных систем в профессиональной сфере | ОПК-6.2.2. Умеет разрабатывать прикладные программные скрипты (парсеры сайтов, краулеры, краблеры) для автоматического сбора и обработки гуманитарных данных                  |
|                                                                                                                                             | ОПК-6.3. Разрабатывает сопутствующую техническую документацию, спецификации, руководства пользователя и комментарии к программному коду        | ОПК-6.3.2. Владеет навыками ведения репозитория проекта в системах контроля версий (Git/GitHub) и составления проектной технической документации                              |

## 2. ОБЪЁМ, СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 2.1. Объём дисциплины

| Семестр | Контактная работа |    |    | СР  | Форма ПА     | Итоговый объём, часов/з. е. |
|---------|-------------------|----|----|-----|--------------|-----------------------------|
|         | Л                 | П  | ПА |     |              |                             |
| 3       | –                 | 64 | 9  | 215 | зачёт/оценка | 288/8                       |
| Итого:  | –                 | 64 | 9  | 215 |              | 288/8                       |

Л — лекции, П — все виды занятий семинарского типа, ПА — промежуточная аттестация, СР — самостоятельная работа обучающегося.

### 2.2. Структура дисциплины

| Наименование тематического раздела дисциплины                                     | Количество часов |    |     |
|-----------------------------------------------------------------------------------|------------------|----|-----|
|                                                                                   | Л                | П  | СР  |
| Архитектура веб-технологий и основы клиент-серверного взаимодействия              |                  | 8  | 26  |
| Разработка веб-парсеров и автоматизированный сбор текстовых данных                |                  | 20 | 68  |
| Работа с открытыми веб-интерфейсами (API) и извлечение данных из социальных медиа |                  | 16 | 54  |
| Базовые технологии манипулирования данными и хранение лингвистических ресурсов    |                  | 10 | 34  |
| Системы контроля версий и техническое документирование ИТ-проектов                |                  | 10 | 33  |
| Всего:                                                                            | –                | 64 | 215 |

### 2.3. Содержание тематических разделов дисциплины

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Раздел 1. Архитектура веб-технологий и основы клиент-серверного взаимодействия</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Принципы функционирования глобальной сети Интернет. Клиент-серверная архитектура. Сетевой протокол HTTP/HTTPS, структура HTTP-запросов и ответов (заголовки, методы GET и POST, коды состояний сервера). Понятие о DOM-модели (Document Object Model) веб-страницы. Основы языков разметки гипертекста (HTML) и каскадных таблиц стилей (CSS): теги, атрибуты, классы, идентификаторы. Инструменты разработчика в современных веб-браузерах (DevTools) как средство анализа структуры веб-страниц для последующего извлечения данных. |
| <b>Раздел 2. Разработка веб-парсеров и автоматизированный сбор текстовых данных</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Понятие о веб-скрейпинге (Web Scraping) и краулинге (Web Crawling). Этические и правовые аспекты автоматического сбора данных в сети Интернет (анализ файлов robots.txt, ограничения по частоте запросов). Использование базовых библиотек Python для отправки сетевых запросов (requests, urllib). Проектирование парсеров веб-сайтов на основе открытой библиотеки BeautifulSoup. Методы поиска элементов по HTML-тегам и CSS-                                                                                                      |

классам. Извлечение текстового контента, ссылок, метаданных. Обработка исключений и сброс сетевого соединения. Архитектура пайплайна пакетного сохранения собранных данных в структурированные файлы (CSV, JSON).

### **Раздел 3. Работа с открытыми веб-интерфейсами (API) и извлечение данных из социальных медиа**

Понятие об архитектурном стиле REST и интерфейсах прикладного программирования (Web API). Принципы авторизации (API-ключи, токены). Отправка запросов к API и обработка ответов в формате JSON. Автоматизированный сбор текстового контента, комментариев и постов из открытых источников и социальных медиа (на примере API доступных платформ, мессенджеров или открытых баз данных). Использование специализированных Python-библиотек (оберток для API). Проектирование скриптов для мониторинга динамического интернет-контента.

### **Раздел 4. Базовые технологии манипулирования данными и хранение лингвистических ресурсов**

Проблема хранения больших объемов собранных текстовых данных. Ограничения текстовых файлов (длинные строки, дублирование). Понятие о реляционных базах данных. Структурированный язык запросов SQL (базовые операторы SELECT, INSERT, WHERE, JOIN). Использование легковесной встроенной СУБД SQLite в Python (модуль `sqlite3`). Создание таблиц, организация связей между ними. Проектирование скриптов для автоматического экспорта собранных парсером веб-данных в таблицы локальной базы данных SQLite.

### **Раздел 5. Системы контроля версий и техническое документирование ИТ-проектов**

Технологии коллективной и индивидуальной разработки программного обеспечения. Системы контроля версий: архитектура и базовые команды распределенной системы Git (`init`, `clone`, `add`, `commit`, `push`, `pull`, `branch`). Работа с удаленными репозиториями на платформе GitHub. Правила ведения репозитория проекта, публикация открытого исходного кода лингвистических скриптов. Разработка сопутствующей технической документации проекта: составление спецификаций, правил развертывания и файлов описания (`README.md`) на языке разметки Markdown.

## **2.4. Организация учебных занятий**

Дисциплина может реализовываться с применением электронного обучения и дистанционных образовательных технологий.

### 3. ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

#### 3.1. Учебная литература

##### 3.1.1. Основная

1. Колесникова Т. Г. Языки программирования : учебное пособие : [16+] / Т. Г. Колесникова; Кемеровский государственный университет. – Кемерово : Кемеровский государственный университет, 2019. – 182 с. : табл., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=573802>.

##### 3.1.2. Дополнительная

1. Воробьев, Г. А. Основы программирования на Python : учебно-методическое пособие : [16+] / Г. А. Воробьев ; Липецкий государственный педагогический университет им. П. П. Семенова-Тян-Шанского. – Липецк : Липецкий государственный педагогический университет им. П.П. Семенова-Тян-Шанского, 2022. – 89 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=700515>.
2. Карякин, М. И. Технологии программирования и компьютерный практикум на языке Python : учебное пособие : [16+] / М. И. Карякин, К. А. Ватульян, Р. М. Мнухин ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2022. – 244 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=698687>.
3. Литвиненко, В. А. Основы объектно-ориентированного программирования задач на графах : учебное пособие : [16+] / В. А. Литвиненко ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2019. – 133 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=612227>.
4. Северенс, Ч. Введение в программирование на Python : учебное пособие / Ч. Северенс. – 2-е изд., испр. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 231 с. : схем., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=429184>.

#### 3.2. Профессиональные базы данных и информационные справочные системы

| Наименование ресурса                                             | Адрес                                                                                                                       | Тип доступа                                            |
|------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| Электронно-библиотечные системы                                  |                                                                                                                             |                                                        |
| ЭБС «Университетская библиотека онлайн»                          | <a href="https://biblioclub.ru/">https://biblioclub.ru/</a>                                                                 | Индивидуальный неограниченный доступ после регистрации |
| КиберЛенинка, российская научная электронная библиотека          | <a href="https://cyberleninka.ru/">https://cyberleninka.ru/</a>                                                             | Открытый ресурс                                        |
| Профессиональные базы данных и информационные справочные системы |                                                                                                                             |                                                        |
| Документация BeautifulSoup4                                      | <a href="https://www.crummy.com/software/BeautifulSoup/bs4/doc/">https://www.crummy.com/software/BeautifulSoup/bs4/doc/</a> | Открытый ресурс                                        |
| MDN Web Docs (Раздел HTTP)                                       | <a href="https://developer.mozilla.org/">https://developer.mozilla.org/</a>                                                 | Открытый ресурс                                        |

| Наименование ресурса                      | Адрес                                                     | Тип доступа     |
|-------------------------------------------|-----------------------------------------------------------|-----------------|
|                                           | ru/docs/Web/HTTP                                          |                 |
| Национальный корпус русского языка (НКРЯ) | <a href="https://ruscorpora.ru">https://ruscorpora.ru</a> | Открытый ресурс |

### 3.3. Сетевые ресурсы

| Наименование ресурса                    | Адрес                                                                                                                                           |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Основы Python                           | <a href="https://education.yandex.ru/handbook/python">https://education.yandex.ru/handbook/python</a>                                           |
| A Byte of Python                        | <a href="https://python.swaroopch.com/">https://python.swaroopch.com/</a>                                                                       |
| A Byte of Python (Russian), Версия 2.01 | <a href="https://myrobot.ru/downloads/books/A_Byte_of_Python_Rus_2.01.pdf">https://myrobot.ru/downloads/books/A_Byte_of_Python_Rus_2.01.pdf</a> |
| Python                                  | <a href="https://www.python.org/">https://www.python.org/</a>                                                                                   |
| The Python Tutorial                     | <a href="https://python.land/python-tutorial">https://python.land/python-tutorial</a>                                                           |
| GitHub                                  | <a href="https://github.com">https://github.com</a>                                                                                             |
| PyPI (Python Package Index)             | <a href="https://pypi.org">https://pypi.org</a>                                                                                                 |

### 3.4. Методическое обеспечение дисциплины

#### 3.4.1. Методические указания для преподавателя

Изучение дисциплины проводится в форме лекций, практических занятий, организации самостоятельной работы студентов, консультаций.

Лекции реализуются через изложение учебного материала с возможным мультимедийным сопровождением. Основными целями лекции являются системное освещение ключевых понятий и положений по соответствующей теме, обзор и оценка существующей проблематики, её методологических и социокультурных оснований, возможных вариантов решения, предложение методических рекомендаций для дальнейшего изучения курса, в том числе литературы и источников. Лектор должен стимулировать обучающихся к участию в обсуждении вопросов лекционного занятия, к высказыванию собственной точки зрения по обсуждаемой проблеме. Главное назначение лекции — обеспечить теоретическую основу обучения, развить интерес к учебной деятельности и конкретной учебной дисциплине, сформировать у обучающихся ориентиры для самостоятельной работы над курсом.

Практические занятия направлены на развитие самостоятельности обучающихся в исследовании изучаемых вопросов и приобретение умений и навыков. Практические занятия традиционно проводятся в форме обсуждения проблемных вопросов в группе при активном участии обучающихся. Они способствуют углубленному изучению наиболее фундаментальных и сложных проблем курса, служат важной формой анализа и синтеза исследуемого материала, а также подведения итогов самостоятельной работы обучающихся, стимулируя развитие профессиональной компетентности, навыков и умений. На практических занятиях обучающиеся учатся работать с научной литературой, чётко и доходчиво излагать проблемы и предлагать варианты их решения, аргументировать свою позицию, оценивать и критиковать позиции других, свободно публично высказывать свои мысли и суждения, грамотно вести полемику и представлять результаты собственных исследований. Практические занятия проводятся в форме устных и письменных опросов, диспута, тестирования, выполнения заданий, обсуждения докладов, выполнения контрольных заданий и пр.

Результаты работы на практических занятиях учитываются преподавателем при выставлении итоговой оценки по данной дисциплине. На усмотрение преподавателя обучающиеся, активно отвечающие на занятиях и выполняющие рекомендации преподавателя при подготовке к ним, могут получить повышающий балл к своей оценке в рамках промежуточной аттестации.

### **3.4.2. Методические рекомендации по самостоятельной работе обучающихся**

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Самостоятельная работа включается в общую трудоёмкость учебной нагрузки обучающегося.

Реализация поставленной цели предполагает решение следующих задач:

- освоение и расширение теоретических знаний по изучаемой дисциплине;
- систематизация и закрепление полученных теоретических знаний и практических навыков;
- формирование умений по поиску и использованию нормативной, справочной и специальной литературы, а также других источников информации;
- развитие познавательных способностей и активности, творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самообразованию, самосовершенствованию и самореализации;
- развитие научно-исследовательских навыков;
- формирование умения решать практические задачи (в профессиональной деятельности), используя приобретённые знания, способности и навыки.

Самостоятельная работа должна быть выполнена индивидуально или являться частью коллективной работы (в случае выполнения группового задания в работе делается соответствующая оговорка).

## **3.5. Материально-техническое обеспечение дисциплины**

### **3.5.1. Аудитории для проведения занятий**

Специальные помещения представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, выполнения курсовых работ, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования.

Специальные помещения укомплектованы учебной мебелью, в том числе мебелью для преподавателя дисциплины, учебной доской.

Обучающиеся из числа лиц с ограниченными возможностями здоровья обеспечиваются электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

Материально-техническое оснащение учебных аудиторий конкретизировано на официальном сайте Университета в подразделе «Материально-техническое обеспечение и

оснащённость образовательного процесса. Доступная среда» раздела «Сведения об образовательной организации».

### 3.5.2. Оборудование и технические средства обучения

Специальные помещения укомплектованы демонстрационным оборудованием (мультимедийный проектор, экран, компьютер, звуковые колонки).

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети Интернет и обеспечением доступа в электронную информационно-образовательную среду Университета.

Электронно-библиотечная система (электронная библиотека) и электронная информационно-образовательная среда обеспечивают одновременный доступ не менее 25 процентов обучающихся.

### 3.5.3. Программное обеспечение

| Наименование                                             | Сведения о лицензии                                          |
|----------------------------------------------------------|--------------------------------------------------------------|
| Moodle, среда дистанционного обучения                    | GNU GPL, свободно распространяемое с открытым исходным кодом |
| Visual Studio Code (VS Code)                             | Свободная (лицензия MIT / Microsoft Software License)        |
| Python (включая дистрибутив Anaconda с Jupyter Notebook) | Свободная (Python Software Foundation License — PSFL)        |
| SQLite (встроенный модуль sqlite3 в Python)              | Общественное достояние (Public Domain)                       |
| Git (локальный клиент)                                   | Свободная (GNU General Public License v2 — GPLv2)            |

## 4. ОЦЕНОЧНЫЕ СРЕДСТВА

### 4.1. Методика проведения текущего контроля успеваемости и промежуточной аттестации

Оценивание уровня учебных достижений обучающихся по дисциплине осуществляется в виде текущего контроля успеваемости и промежуточной аттестации. Фонд оценочных средств по дисциплине включает оценочные материалы, направленные на проверку освоения компетенций, в том числе знаний, умений и навыков. Фонд оценочных средств включает оценочные средства текущего контроля и оценочные средства промежуточной аттестации.

#### 4.1.1. Балльно-рейтинговая карта дисциплины

| Виды работы                             | Виды контроля, критерии оценки и количество баллов                                                                                                                                                                                              |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Текущий контроль</b>                 |                                                                                                                                                                                                                                                 |
| Аудиторная работа<br>(0-20 баллов)      | Посещение занятий и участие в работе:<br>10 баллов — посещение не менее 50% занятий<br>15 баллов — посещение 60-80% занятий, участие в обсуждениях материала<br>20 баллов — посещение 90-100% занятий, активное участие в обсуждениях материала |
| Самостоятельная работа<br>(0-40 баллов) |                                                                                                                                                                                                                                                 |
| <b>Промежуточная аттестация</b>         |                                                                                                                                                                                                                                                 |
| Контрольное задание<br>(0-40 баллов)    | Выполнение письменного контрольного задания в СДО Moodle                                                                                                                                                                                        |

#### 4.1.2. Шкала перевода рейтинговых баллов в пятибалльную систему оценивания

| Общее количество баллов                                                                                                                                                                                                                                              | Оценка сформированности компетенций                                                                   | Оценка результатов обучения по дисциплине | Оценка ECTS |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|-------------------------------------------|-------------|
| 0–34                                                                                                                                                                                                                                                                 | Компетенции не сформированы.                                                                          | неудовлетворительно<br>(не зачтено)       | F           |
| Теоретическое содержание не освоено, практические навыки не сформированы, большинство учебных заданий не выполнено, качество их выполнения минимальное, все задания содержат грубые ошибки. Обучающийся не готов решать типовые профессиональные задачи.             |                                                                                                       |                                           |             |
| 35–49                                                                                                                                                                                                                                                                | Уровень владения компетенциями недостаточный для их формирования в результате обучения по дисциплине. | неудовлетворительно<br>(не зачтено)       | FX          |
| Теоретическое содержание освоено частично, практические навыки не сформированы, большинство учебных заданий не выполнено, качество их выполнения минимальное, большинство заданий выполнено с ошибками. Обучающийся не готов решать типовые профессиональные задачи. |                                                                                                       |                                           |             |
| 50–59                                                                                                                                                                                                                                                                | Уровень владения компетенциями                                                                        | удовлетворительно                         | E           |

| Общее количество баллов                                                                                                                                                                                                                                                                                                               | Оценка сформированности компетенций                                                                             | Оценка результатов обучения по дисциплине | Оценка ECTS |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|-------------------------------------------|-------------|
|                                                                                                                                                                                                                                                                                                                                       | посредственный для их формирования в результате обучения по дисциплине.                                         | (зачтено)                                 |             |
| Теоретическое содержание освоено частично, практические навыки сформированы фрагментарно, многие учебные задания не выполнены, качество их выполнения минимальное, многие задания выполнены с ошибками. Обучающийся готов решать типовые профессиональные задачи.                                                                     |                                                                                                                 |                                           |             |
| 60–69                                                                                                                                                                                                                                                                                                                                 | Уровень владения компетенциями удовлетворительный для их формирования в результате обучения по дисциплине.      | удовлетворительно (зачтено)               | D           |
| Теоретическое содержание освоено частично, пробелы не носят существенного характера, практические навыки в основном сформированы, большинство учебных заданий выполнено, качество их выполнения удовлетворительное, некоторые задания выполнены с ошибками. Обучающийся готов решать типовые профессиональные задачи.                 |                                                                                                                 |                                           |             |
| 70–89                                                                                                                                                                                                                                                                                                                                 | Уровень владения компетенциями преимущественно высокий для их формирования в результате обучения по дисциплине. | хорошо (зачтено)                          | C           |
| Теоретическое содержание освоено полностью, некоторые практические навыки сформированы недостаточно, все учебные задания выполнены, качество их выполнения высокое, некоторые задания выполнены с ошибками. Обучающийся готов решать типовые и ситуативные профессиональные задачи.                                                   |                                                                                                                 |                                           |             |
| 90–94                                                                                                                                                                                                                                                                                                                                 | Уровень владения компетенциями высокий для их формирования в результате обучения по дисциплине.                 | отлично (зачтено)                         | B           |
| Теоретическое содержание освоено полностью, практические навыки сформированы, все учебные задания выполнены, качество их выполнения близко к максимальному, однако есть несколько незначительных ошибок. Обучающийся готов эффективно решать типовые и ситуативные профессиональные задачи, в том числе повышенного уровня сложности. |                                                                                                                 |                                           |             |
| 95–100                                                                                                                                                                                                                                                                                                                                | Уровень владения компетенциями превосходный для их формирования в результате обучения по дисциплине.            | отлично (зачтено)                         | A           |
| Теоретическое содержание освоено полностью, практические навыки сформированы, все учебные задания выполнены, качество их выполнения близко к максимальному. Обучающийся готов эффективно решать типовые и ситуативные профессиональные задачи, в том числе повышенного уровня сложности, и способен разрабатывать новые решения.      |                                                                                                                 |                                           |             |

## 4.2. Оценочные средства текущего контроля

### 4.2.1. Вопросы для подготовки к семинарским занятиям

#### Раздел 1. Архитектура веб-технологий и основы клиент-серверного взаимодействия

1. Опишите фундаментальные принципы функционирования глобальной сети Интернет. В чём заключается суть клиент-серверной архитектуры взаимодействия?

2. Раскройте назначение и структуру сетевого протокола HTTP/HTTPS. Из каких компонентов состоит HTTP-запрос (стартовая строка, заголовки, тело) и HTTP-ответ сервера?
3. Проведите сравнительный анализ основных методов HTTP-запросов: в чём разница между методами GET и POST при передаче параметров веб-серверу?
4. На какие основные группы делятся коды состояний сервера (HTTP status codes)? Что означают коды 200 OK, 301 Moved Permanently, 403 Forbidden, 404 Not Found и 500 Internal Server Error?
5. Что такое DOM-модель (Document Object Model) веб-страницы? Как браузер строит древовидную структуру документа на основе HTML-кода?
6. Изучите основы языка разметки гипертекста (HTML). Каково назначение тегов, атрибутов и структуры вложенности элементов?
7. Какую роль выполняют каскадные таблицы стилей (CSS)? Опишите разницу между CSS-классами (class) и уникальными идентификаторами (id) элементов разметки.
8. Какие HTML-теги традиционно используются для структурирования текстового контента на веб-страницах (заголовки <h1>-<h6>, абзацы <p>, списки <ul>/<ol>, таблицы <table>, контейнеры <div>/<span>, ссылки <a>)?
9. Что такое инструменты разработчика в современных веб-браузерах (DevTools, вкладка Elements/Network)? Как использовать DevTools для визуального анализа структуры веб-страницы перед написанием парсера?
10. Каким образом заголовки запроса (например, User-Agent) позволяют веб-серверу идентифицировать клиента, и почему браузеры автоматически передают эти метаданные?
11. Как устроен механизм работы файлов cookie и сессий, и какое значение они имеют для сохранения состояния пользователя при веб-навигации?
12. Практическое задание: Откройте в браузере сайт любого сетевого СМИ. С помощью инструментов разработчика (DevTools) найдите HTML-тег, содержащий главный заголовок статьи, и определите все его CSS-классы и атрибуты. Выпишите полную строку тега.

## **Раздел 2. Разработка веб-парсеров и автоматизированный сбор текстовых данных**

1. Проведите разграничение между понятиями Web Scraping (извлечение данных со страниц) и Web Crawling (рекурсивный обход ссылок веб-сайта). Как эти технологии применяются для автоматического сбора лингвистических корпусов?
2. Сформулируйте этические и правовые аспекты автоматического сбора данных в сети Интернет. Зачем нужен файл robots.txt, как его анализировать и какие директивы (User-agent, Disallow, Crawl-delay) обязан соблюдать парсер?
3. Опишите возможности базовых библиотек Python для отправки сетевых запросов: requests и urllib. Как отправить GET-запрос и получить сырой HTML-код страницы в виде строки с помощью метода requests.get()?
4. Раскройте архитектуру и синтаксис открытой библиотеки BeautifulSoup. Как инициализировать объект парсера и передать ему HTML-документ?
5. Опишите методы поиска элементов в BeautifulSoup: методы find() и find\_all(). Каковы их параметры и возвращаемые типы данных (объект тега и список объектов)?
6. Как в BeautifulSoup осуществлять точечный поиск элементов по комбинации HTML-тега и его CSS-класса (параметр attrs или именованный аргумент class\_)?
7. Как извлечь очищенный текстовый контент из найденного HTML-тега (свойства .text, .get\_text()) и значения его атрибутов (например, извлечение ссылки из атрибута href тега <a>)?
8. С какими сбоями сетевого соединения (тайм-ауты, падение сервера) и исключениями библиотеки requests может столкнуться парсер? Как настроить безопасную обработку ошибок через блоки try-except?

9. Почему при проектировании парсера критически важно внедрять искусственные паузы между запросами (модуль `time`, функция `time.sleep()`), и как это защищает скрипт от блокировок по IP-адресу (анти-DdoS)?
10. Опишите архитектуру пайплайна пакетного сохранения собранных данных: как организовать цикл обхода веб-страниц и записывать извлеченный текст в структурированные файлы CSV или JSON?
11. Что такое динамический веб-контент (AJAX/JavaScript-рендеринг)? Почему библиотека `requests` не видит текст, подгружаемый скриптами, и в чём ограничения статического парсинга?
12. Практическое задание: Напишите скрипт на Python с использованием библиотек `requests` и `BeautifulSoup`, который отправляет запрос на учебную веб-страницу, находит все абзацы `<p class="article-text">`, извлекает из них чистый текст, очищает его от лишних пробелов и выводит построчно на экран.

### Раздел 3. Работа с открытыми веб-интерфейсами (API) и извлечение данных из социальных медиа

1. Что такое интерфейс прикладного программирования (Web API)? Раскройте суть и базовые принципы архитектурного стиля REST (клиент-серверное разделение, отсутствие состояния, единообразие интерфейса).
2. Как в REST API используются стандартные HTTP-методы (GET, POST, PUT, DELETE) для выполнения CRUD-операций над ресурсами?
3. Опишите принципы авторизации и аутентификации при работе с открытыми веб-интерфейсами. Что такое API-ключи, токены доступа (Bearer tokens) и зачем разработчики защищают свои API?
4. Как формируется строка запроса к API? Каковы правила передачи параметров фильтрации и пагинации данных через URL-строку (query parameters)?
5. Каковы преимущества обработки ответов API по сравнению с классическим веб-скрейпингом страниц? Почему ответы API возвращаются преимущественно в структурированном формате JSON?
6. Как с помощью библиотеки `requests` отправить авторизованный запрос к API и мгновенно преобразовать JSON-ответ сервера в стандартный словарь или список Python (метод `.json()`)?
7. Опишите сценарии и возможности автоматизированного сбора текстового контента, комментариев или постов из открытых источников, мессенджеров (например, Telegram) или открытых баз данных.
8. Что такое библиотеки-обёртки (API wrappers)? Каковы преимущества использования специализированных Python-библиотек, инкапсулирующих низкоуровневые HTTP-запросы к конкретным платформам?
9. С какими ограничениями на частоту и объём выгружаемых данных (Rate Limits) сталкивается лингвист при работе с Web API, и как обрабатывать HTTP-код состояния 429 Too Many Requests?
10. Опишите логику проектирования скриптов для непрерывного или периодического мониторинга динамического интернет-контента (новостных лент, Telegram-каналов).
11. В чём заключаются риски зависимости академического ИТ-проекта от внешних коммерческих Web API (изменение политик доступа, закрытие интерфейсов, платность)?
12. Практическое задание: Представьте, что API возвращает JSON-ответ в виде словаря: `{"status": "success", "data": [{"id": 1, "text": "Hello world"}, {"id": 2, "text": "Python NLP"}]}`. Напишите фрагмент кода на Python, который извлекает этот словарь из переменной ответа, итерируется по списку в ключе `"data"` и выводит на экран только значения по ключу `"text"`.

### Раздел 4. Базовые технологии манипулирования данными и хранение лингвистических ресурсов

1. С какими фундаментальными проблемами сталкивается исследователь при хранении больших объёмов собранных текстовых данных в обычных файлах (длинные строки, дублирование метаданных, медленный поиск, отсутствие транзакционности)?
2. Сформулируйте понятие о реляционных базах данных. Что такое таблица, строка (запись), столбец (поле), первичный ключ (Primary Key) и типы данных полей?
3. Раскройте назначение структурированного языка запросов SQL. Опишите синтаксис и параметры базового оператора извлечения информации SELECT (выбор полей, указание таблицы).
4. Как в SQL-запросах реализуется фильтрация данных с помощью оператора WHERE и логических условий (AND, OR, LIKE)? Приведите лингвистический пример поиска слов по текстовой маске.
5. Опишите синтаксис оператора INSERT INTO для добавления новых текстовых записей в базу данных.
6. Зачем нужна операция связывания таблиц JOIN? Как организация связей между таблицами (например, таблица Documents и таблица Sentences, связанные по внешнему ключу Foreign Key) устраняет избыточность хранения данных?
7. Охарактеризуйте особенности и преимущества легковесной встроенной СУБД SQLite. Почему она относится к классу автономных (serverless) СУБД и идеальна для локального хранения лингвистических ресурсов?
8. Опишите алгоритм работы с SQLite в Python с помощью встроенного модуля sqlite3: подключение к файлу базы (connect()), создание объекта курсора (cursor()), выполнение SQL-команд (execute()) и фиксация транзакций (commit()).
9. Зачем нужен метод курсора fetchall() и fetchone() после выполнения SQL-запроса SELECT? В каком виде модуль sqlite3 возвращает строки таблицы в Python (список кортежей)?
10. Опишите архитектуру пайплайна экспорта данных: как спроектировать скрипт, который автоматически передает извлеченные веб-парсером текстовые блоки и метаданные в таблицы локальной базы данных SQLite?
11. Как обеспечить безопасность SQL-запросов и защитить базу данных от критической уязвимости «SQL-инъекция» с помощью параметризованных запросов (передача кортежа аргументов через плейсхолдеры)?
12. Практическое задание: Напишите на Python скрипт с использованием модуля sqlite3, который подключается к локальной базе данных corpus.db, создает таблицу texts с тремя полями (id целое число первичный ключ, title текстовое, content текстовое) и фиксирует изменения.

## Раздел 5. Системы контроля версий и техническое документирование ИТ-проектов

1. Какова роль систем контроля версий в процессах коллективной и индивидуальной разработки программного обеспечения? В чём преимущество распределенных систем перед централизованными?
2. Раскройте архитектуру системы Git. Опишите три основные области Git: рабочая директория (Working Directory), область подготовленных файлов (Staging Area / Index) и локальный репозиторий (Git Directory / Repository).
3. Объясните функции базовых команд локального управления версиями: git init (инициализация), git status (проверка состояния) и git log (просмотр истории коммитов).
4. Как работает команда git add? Зачем файлы сначала переводятся в индекс (Staging Area), а не коммитятся напрямую? Что такое файл .gitignore и какие типы файлов (бэкапы, базы SQLite, кэш) нужно в него вносить?
5. Опишите назначение команды git commit. Каковы правила написания хороших, информативных сообщений к коммитам (commit messages)? Что означает флаг -m?

6. Как устроена работа с удаленными репозиториями на платформе GitHub? Опишите назначение команд связи и синхронизации: `git clone`, `git remote add`, `git push`, `git pull`.
7. Раскройте концепцию ветвления в Git. Зачем нужны ветки, как создать новую ветку (`git branch`, `git checkout -b`) и как выполнить слияние веток (`git merge`)? Что такое конфликт слияния (merge conflict) и как его разрешить?
8. Сформулируйте правила ведения профессионального репозитория лингвистического ИТ-проекта на GitHub: структура папок, публикация только исходного кода скриптов и защита главной ветки `main/master`.
9. Каковы требования к созданию сопутствующей технической документации ИТ-проекта? Какую роль играет файл описания `README.md` в корневом каталоге проекта?
10. Изучите синтаксис облегченного языка разметки Markdown: как оформлять заголовки разного уровня, списки (маркированные и нумерованные), выделение текста (полужирный, курсив), вставку ссылок, скриншотов и блоков программного кода?
11. Как составить качественную спецификацию и правила развертывания лингвистического скрипта в файле `README.md`, чтобы сторонний исследователь мог запустить ваш код без участия автора?
12. Практическое задание: Напишите на языке разметки Markdown фрагмент текста для файла `README.md`, содержащий: главный заголовок проекта, маркированный список из 3 пунктов с системными требованиями (версия Python, библиотеки) и блок с примером исходного кода Python (оформленный тремя обратными апострофами с указанием синтаксиса языка).

#### Критерии оценки работы на практическом занятии

| Критерии                                                                                                                                                                                                                                              | Максимальное количество баллов за занятие |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| <b>Устный опрос, коллоквиум</b>                                                                                                                                                                                                                       |                                           |
| Основные теоретические положения по вопросу раскрыты. Имеются элементы обоснования выводов. Имеются элементы систематизации информации, факты применения профессиональной терминологии. Очевидно использование источников рекомендованной литературы. | 5 баллов                                  |

#### 4.2.2. Темы докладов

1. Архитектура клиент-серверного взаимодействия (HTTP/HTTPS) как технологический базис функционирования современных лингвистических веб-ресурсов.
2. DOM-модель веб-страницы и основы селекции элементов (HTML/CSS) в задачах автоматизированного синтаксического разбора веб-контента.
3. Методология Web Scraping и Web Crawling: принципы проектирования масштабируемых парсеров для сбора языковых корпусов.
4. Правовые, этические и комплаенс-ограничения автоматического сбора данных в сети Интернет: анализ `robots.txt` и защита от блокировок.
5. Архитектурный стиль REST и интерфейсы прикладного программирования (Web API) в системах интеграции лингвистических данных.
6. Сравнительный анализ эффективности сбора текстовых данных через Web API и прямой парсинг HTML-страниц (BeautifulSoup).
7. Технологии потокового мониторинга динамического контента мессенджеров и социальных медиа средствами языка Python.

8. Проблема организации и долгосрочного хранения терабайтных массивов неструктурированного текста: от текстовых файлов к СУБД.
9. Реляционная модель данных и язык SQL как инструмент структурирования лингвистических ресурсов и корпусов примеров.
10. Легковесная СУБД SQLite в экосистеме Python: проектирование локальных баз данных для академических ИТ-проектов.
11. Безопасность баз данных в лингвистических веб-приложениях: анализ и предотвращение уязвимостей типа SQL-инъекций.
12. Системы контроля версий в управлении конфигурацией ПО: архитектура Git и организация индивидуального рабочего пространства разработчика.
13. Методология Git Flow и коллективная разработка элементов лингвистического ПО в распределенных междисциплинарных командах.
14. Платформа GitHub как инструмент построения профессионального портфолио, ведения открытых репозиторий (Open Source) и социального кодирования.
15. Разрешение конфликтов слияния (Merge Conflicts) в системах контроля версий: технологические подходы и инструменты.
16. Стандарты технического документирования ИТ-проектов: структура файла README.md и язык разметки Markdown в цифровой гуманитаристике.
17. Использование инструментов разработчика (DevTools) современных браузеров для аудита сетевой активности веб-приложений.
18. Проектирование комплексного автоматизированного пайплайна: от парсинга веб-страницы до реляционного хранения данных в SQLite и публикации релиза на GitHub.

#### Шкала и критерии оценки доклада

| Критерии                               | Показатели                                                                                                                                                                                                                                                                                                                                                                | Баллы |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1. Степень раскрытия сущности проблемы | <ul style="list-style-type: none"> <li>– соответствие теме доклада;</li> <li>– полнота и глубина раскрытия основных понятий;</li> <li>– умение работать с литературой, систематизировать и структурировать материал;</li> <li>– умение обобщать, сопоставлять различные точки зрения по рассматриваемому вопросу, аргументировать основные положения и выводы.</li> </ul> | 70    |
| 2. Обоснованность выбора источников    | <ul style="list-style-type: none"> <li>– круг, полнота использования литературных источников по теме;</li> <li>– привлечение новейших работ (журнальные публикации, материалы сборников научных трудов и т.д.).</li> </ul>                                                                                                                                                | 15    |
| 3. Изложение                           | – литературный стиль.                                                                                                                                                                                                                                                                                                                                                     | 15    |

Доклад оценивается по 100 балльной шкале, баллы переводятся в оценки успеваемости следующим образом:

90 – 100 баллов – «отлично»;

70 – 89 баллов – «хорошо»;

50 – 69 баллов – «удовлетворительно»;

менее 50 баллов – «неудовлетворительно».

### 4.3. Оценочные средства промежуточной аттестации

#### 4.3.1. Контрольные задания

**ОПК-3.1.1-1. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.**

С помощью какой встроенной функции открывается файл в Python:

- А. file.open()
- Б. open()
- В. read\_file()
- Г. start()
- Д. load()

|  |
|--|
|  |
|  |

**ОПК-3.1.1-2. Прочитайте текст и установите соответствие.**

Установите соответствие между характеристиками (1-2) и типами данных в Python (А-Б):

|                            |            |
|----------------------------|------------|
| А. Изменяемый тип данных   | 1. Словарь |
| Б. Неизменяемый тип данных | 2. Строки  |

Запишите выбранные цифры под соответствующими буквами:

|   |   |
|---|---|
| А | Б |
|   |   |

**ОПК-3.1.2-1. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.**

Как конкатенировать две строки a="Hello" и b="World!":

- А. a . b
- Б. a & b
- В. a + b
- Г. a join b
- Д. Другой ответ

|  |
|--|
|  |
|  |

**ОПК-3.1.2-2. Прочитайте текст и установите последовательность.**

Установите правильную последовательность операторов для выполнения следующих действий:  
 1) Разбить предложение на список слов (по пробелам); 2) Вырезать из этого списка элементы со второго по четвертый 3) Соединить эти элементы обратно в строку через дефис

- А. "-".join(words)
- Б. words = text.split()
- В. words = words[1:4]
- Г. words = text.split('-')

Запишите соответствующую последовательность букв слева направо:

|  |
|--|
|  |
|--|

**ОПК-3.1.3-1. Прочитайте текст и установите соответствие.**

Установите соответствие между характеристиками (1-2) и типами данных в Python (А-Б):

|                            |              |
|----------------------------|--------------|
| А. Изменяемый тип данных   | 1. Множество |
| Б. Неизменяемый тип данных | 2. Кортеж    |

Запишите выбранные цифры под соответствующими буквами:

|   |   |
|---|---|
| А | Б |
|   |   |

**ОПК-3.1.3-2. Прочитайте текст и установите последовательность.**

Установите правильную последовательность операторов для выполнения следующих действий:  
 1) Развернуть строку задом наперед; 2) Взять первые 5 символов получившейся строки; 3) Перевести их в верхний регистр.

- А. [:5]
- Б. [::-1]
- В. s
- Г. .upper()
- Д. .reverse()

Запишите соответствующую последовательность букв слева направо:

|  |
|--|
|  |
|--|

**ОПК-3.1.3-3. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.**

Какая функция позволяет получить данные от пользователя через консоль:

- А. get()
- Б. read()
- В. input()
- Г. scanf()
- Д. Другой ответ

|  |
|--|
|  |
|  |

**ОПК-3.2.1-1. Прочитайте текст и установите соответствие.**

Установите соответствие между характеристиками (1-2) и типами данных в Python (А-Б):

|                            |                     |
|----------------------------|---------------------|
| А. Изменяемый тип данных   | 1. Текстовая строка |
| Б. Неизменяемый тип данных | 2. Список           |

Запишите выбранные цифры под соответствующими буквами:

|   |   |
|---|---|
| А | Б |
|   |   |

|  |  |
|--|--|
|  |  |
|--|--|

**ОПК-3.2.1-2. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.**

Какой режим (mode) нужно указать при открытии файла только для чтения:

- А. 'w'
- Б. 'a'
- В. 'x'
- Г. 'r'
- Д. Другой ответ

|  |
|--|
|  |
|  |

**ОПК-3.2.2-1. Прочитайте текст и установите соответствие.**

Установите соответствие между характеристиками (1-2) и типами данных в Python (А-Б):

|                            |                    |
|----------------------------|--------------------|
| А. Изменяемый тип данных   | 1. Множества       |
| Б. Неизменяемый тип данных | 2. Булево значение |

Запишите выбранные цифры под соответствующими буквами:

|   |   |
|---|---|
| А | Б |
|   |   |

**ОПК-3.2.2-2. Прочитайте текст и установите последовательность.**

Установите правильную последовательность операторов для выполнения следующих действий:  
 1) Выбрать строку 2) Убрать лишние пробелы по краям строки; 3) Заменить все точки на запятые; 4) Привести всю строку к нижнему регистру.

- А. .lower()
- Б. text
- В. .strip()
- Г. .replace('.', ',')

Запишите соответствующую последовательность букв слева направо:

|  |
|--|
|  |
|--|

**ОПК-3.2.2-3. Прочитайте текст и установите последовательность.**

Установите правильную последовательность операторов для выполнения следующих действий:  
 1) Найти индекс первого вхождения символа "@"; 2) Взять срез строки от начала до этого индекса (не включая его); 3) Вывести результат на экран.

- А. email[:idx]
- Б. print(
- В. idx = email.find("@")
- Г. )

Запишите соответствующую последовательность букв слева направо:

|  |
|--|
|  |
|--|

**ОПК-3.2.3-1. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.**

Какой метод считывает всё содержимое файла целиком в одну строку:

- А. read()
- Б. readline()
- В. readlines()
- Г. get\_text()
- Д. Другой ответ

|  |
|--|
|  |
|  |

**ОПК-3.2.3-2. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.**

Какой тип данных у переменной double в результате выполнения следующей программной структуры:

```
double=lambda x: x**2
print(type(double))
```

- А. целое число
- Б. вещественное число
- В. функция
- Г. список
- Д. программа не сработает из-за ошибки

|  |
|--|
|  |
|  |

**ОПК-3.2.3-3. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа.**

Какой значение будет выведено в консоль в результате выполнения следующей программной конструкции:

```
a=5
b=2

def mul():
    a+=1
    b+=2
    return(a*b)
print(mul())
```

- А. 10

- Б. 18
- В. 24
- Г. 0
- Д. программа не работает из-за ошибки

|  |
|--|
|  |
|  |

#### 4.3.2. Ключи к контрольным заданиям

|             |                                                                                       |
|-------------|---------------------------------------------------------------------------------------|
| ОПК-3.1.1-1 | В<br>Файлы в Python открываются функцией open().                                      |
| ОПК-3.1.1-2 | A1B2                                                                                  |
| ОПК-3.1.2-1 | В<br>Конкатенация строк в Python осуществляется операцией +.                          |
| ОПК-3.1.2-2 | БВА                                                                                   |
| ОПК-3.1.3-1 | A1B2                                                                                  |
| ОПК-3.1.3-2 | ВБАГ                                                                                  |
| ОПК-3.1.3-3 | В<br>Ввод данных с консоли в Python осуществляется функцией input().                  |
| ОПК-3.2.1-1 | A2B1                                                                                  |
| ОПК-3.2.1-2 | Г<br>Режим чтения файла в Python осуществляется параметром 'r'.                       |
| ОПК-3.2.2-1 | A1B2                                                                                  |
| ОПК-3.2.2-2 | БВГА                                                                                  |
| ОПК-3.2.2-3 | ВБАГ                                                                                  |
| ОПК-3.2.3-1 | А<br>Чтение всего текстового файла в одну большую строку осуществляет функция read(). |
| ОПК-3.2.3-2 | В<br>lambda – это функция.                                                            |
| ОПК-3.2.3-3 | Д<br>a,b – это глобальные переменные.                                                 |

#### Шкала и критерии оценки текущего тестирования

| Число правильных ответов     | Оценка                       |
|------------------------------|------------------------------|
| 90-100% правильных ответов   | Оценка «отлично»             |
| 70-89% правильных ответов    | Оценка «хорошо»              |
| 50-69% правильных ответов    | Оценка «удовлетворительно»   |
| Менее 50% правильных ответов | Оценка «неудовлетворительно» |

## **5. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ В ОТНОШЕНИИ ЛИЦ ИЗ ЧИСЛА ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ**

Профессорско-преподавательский состав знакомится с психолого-физиологическими особенностями обучающихся инвалидов и лиц с ограниченными возможностями здоровья, индивидуальными программами реабилитации инвалидов (при наличии). При необходимости осуществляется дополнительная поддержка преподавания психологами, социальными работниками, прошедшими подготовку ассистентами.

В соответствии с методическими рекомендациями Министерства науки и высшего образования Российской Федерации, в курсе предполагается использовать социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создания комфортного психологического климата в студенческой группе. Подбор и разработка учебных материалов производятся с учётом предоставления материала в различных формах: аудиальной, визуальной, с использованием специальных технических средств и информационных систем.

Согласно требованиям к порядку реализации образовательной деятельности в отношении инвалидов и лиц с ОВЗ, установленным Министерством науки и высшего образования Российской Федерации, необходимо иметь в виду, что:

1) инвалиды и лица с ОВЗ по зрению имеют право присутствовать на занятиях вместе с ассистентом, оказывающим обучающемуся необходимую помощь;

2) инвалиды и лица с ОВЗ по слуху имеют право на использование звукоусиливающей аппаратуры.

При проведении промежуточной аттестации по дисциплине обеспечивается соблюдение следующих общих требований:

- проведение аттестации для инвалидов в одной аудитории совместно с обучающимися, не являющимися инвалидами, если это не создает трудностей для инвалидов и иных обучающихся при промежуточной аттестации;

- присутствие в аудитории ассистента (ассистентов), оказывающего обучающимся инвалидам необходимую техническую помощь с учётом их индивидуальных особенностей (занять рабочее место, передвигаться, прочитать и оформить задание, общаться с экзаменатором);

- пользование необходимыми обучающимся инвалидам техническими средствами при прохождении промежуточной аттестации с учётом их индивидуальных особенностей;

- обеспечение возможности беспрепятственного доступа обучающихся инвалидов в аудитории, туалетные и другие помещения, а также их пребывания в указанных помещениях.

По письменному заявлению обучающегося инвалида продолжительность прохождения испытания промежуточной аттестации (зачёта, экзамена, и др.) обучающимся инвалидом может быть увеличена по отношению к установленной продолжительности его сдачи:

- продолжительность сдачи испытания, проводимого в письменной форме, — не более чем на 90 минут;

- продолжительность подготовки обучающегося к ответу, проводимому в устной форме, — не более чем на 20 минут.

В зависимости от индивидуальных особенностей обучающихся с ОВЗ Университет обеспечивает выполнение следующих требований при проведении аттестации:

а) для слепых:

- задания и иные материалы для прохождения промежуточной аттестации оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением для слепых, либо зачитываются ассистентом;

- письменные задания выполняются обучающимися на бумаге рельефно-точечным шрифтом Брайля или на компьютере со специализированным программным обеспечением для слепых, либо надиктовываются ассистенту;

- при необходимости обучающимся предоставляется комплект письменных принадлежностей и бумага для письма рельефно-точечным шрифтом Брайля, компьютер со специализированным программным обеспечением для слепых;

б) для слабовидящих:

- задания и иные материалы для сдачи экзамена оформляются увеличенным шрифтом;

- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;

- при необходимости обучающимся предоставляется увеличивающее устройство, допускается использование увеличивающих устройств, имеющихся у обучающихся;

в) для глухих и слабослышащих, с тяжелыми нарушениями речи:

- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающимся предоставляется звукоусиливающая аппаратура индивидуального пользования;

- по их желанию испытания проводятся в письменной форме;

г) для лиц с нарушениями опорно-двигательного аппарата (тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания выполняются обучающимися на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;

- по их желанию испытания проводятся в устной форме.

О необходимости обеспечения специальных условий для проведения аттестации обучающийся должен сообщить письменно не позднее, чем за 10 дней до начала аттестации. К заявлению прилагаются документы, подтверждающие наличие у обучающегося индивидуальных особенностей (при отсутствии указанных документов в организации).

При необходимости для обучающихся с инвалидностью процедура оценивания результатов обучения может проводиться в несколько этапов.