МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный гуманитарный университет»
(ФГАОУ ВО «РГГУ»)

ИСТОРИКО-АРХИВНЫЙ ИНСТИТУТ
ИСТОРИЧЕСКИЙ ФАКУЛЬТЕТ
Кафедра всеобщей истории

ПРОГРАММИРОВАНИЕ ДЛЯ ГУМАНИТАРНЫХ НАУК

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

46.04.01 История
Код и наименование направления подготовки/специальности

Искусственный интеллект и цифровые технологии в исторических исследованиях
Наименование направленности (профиля)/ специализации

Уровень высшего образования: магистратура

Форма обучения: *очная*

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2026

Программирование для гуманитарных наук
Рабочая программа дисциплины

Составители:
к.э.н., доц., заведующий кафедрой фундаментальной
и прикладной математики, А.Ю. Журавлев

УТВЕРЖДЕНО
Протокол заседания кафедры фундаментальной и прикладной математики
№ 8 от 06.12.2025

ОГЛАВЛЕНИЕ

1. Пояснительная записка

1.1. Цель и задачи дисциплины

Цель дисциплины: сформировать у студентов-гуманитариев базовые компетенции в области программирования и алгоритмического мышления, необходимые для самостоятельного решения прикладных задач исторического исследования, включая сбор, обработку, анализ и визуализацию данных, а также для критической оценки цифровых инструментов и методов.
Задачи дисциплины:

- познакомить студентов с основами синтаксиса, структурами данных и базовыми алгоритмами на языке программирования Python, как наиболее востребованного в Data Science и Digital Humanities;
- сформировать навыки автоматизации процессов обработки текстовых (в том числе исторических источников), табличных и сетевых данных;
- научить применять специализированные библиотеки Python для анализа данных (Pandas, NumPy), визуализации результатов (Matplotlib, Seaborn), работы с естественным языком и веб-скрапинга;
- развить умение декомпозировать исследовательскую задачу гуманитарного профиля на последовательные этапы, формализовать её и реализовать в виде скрипта или проекта;
- способствовать пониманию принципов работы с базами данных (SQL) и основ создания простых историко-ориентированных информационных систем.

1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компетенция (код и наименование)	**Индикаторы компетенций** (код и наименование)	**Результаты обучения**
ПК-1. Способен к самостоятельной подготовке и проведению научно-исследовательских работ с использованием знания фундаментальных и прикладных общепрофессиональных дисциплин, и профессиональных дисциплин направленности (профиля) учебного плана, с применением современного программного обеспечения; способен представлять результаты научных исследований, в том числе при подготовке и проведении	ПК-1.3. Демонстрирует знание современного программного обеспечения, тематических сетевых ресурсов, баз данных и информационных систем, необходимых для проведения исторического исследования	**Знать:** базовый синтаксис языка Python; основные структуры данных (списки, словари, кортежи, множества) и их применение для хранения исторической информации; принципы работы с внешними файлами (текстовые, CSV, JSON); названия и назначение ключевых библиотек для анализа данных (Pandas, NumPy) и визуализации (Matplotlib); основы объектно-ориентированного программирования применительно к моделированию исторических объектов. **Уметь:** писать простые скрипты на Python для автоматизации рутинных операций с файлами и текстами; читать и обрабатывать данные из CSV-файлов, экспортированных из исторических баз данных или электронных таблиц;

научных семинаров, конференций, подготовке и редактировании научных публикаций		преобразовывать данные из одного формата в другой (например, текст -> структурированная таблица); использовать базовые функции библиотек Pandas для фильтрации и первичного анализа наборов исторических данных. **Владеть:** навыками установки и настройки среды разработки (IDE) для Python; навыками поиска и установки необходимых пакетов с использованием менеджера pip; техникой отладки скриптов через анализ сообщений об ошибках; базовой терминологией программирования для корректной постановки технических задач и взаимодействия с разработчиками.
ПК-4. Способен ориентироваться в программном обеспечении информационных систем и баз данных историко-ориентированного профиля; создавать историко-ориентированные информационные системы и базы данных; способен использовать в конкретно-исторических исследованиях, основанных на информации массовых исторических источников, методы и технологии математической статистики и компьютерного моделирования, современной науки о данных	ПК-4.1. Умеет ориентироваться в программном обеспечении информационных систем и баз данных, умеет создавать историко-ориентированные информационные системы и базы данных, использовать в конкретно-исторических исследованиях методы и технологии математической статистики и компьютерного моделирования, современной науки о данных	**Знать:** принципы организации реляционных (табличных) данных и язык запросов SQL на уровне SELECT-запросов с условиями, сортировкой и простыми JOIN; основные этапы процесса анализа данных (Data Science) в гуманитарных науках: сбор, очистка, анализ, визуализация; основы регулярных выражений (regex) для поиска и извлечения паттернов из текстовых исторических источников; базовые концепции компьютерного моделирования (модель, параметры, симуляция). **Уметь:** подключаться к базам данных (например, SQLite) из Python-скрипта и выполнять запросы для получения данных; очищать и предобрабатывать неструктурированные текстовые данные (удаление стоп-слов, приведение к нижнему регистру, лемматизация с использованием сторонних библиотек); строить простые статистические графики (гистограммы, диаграммы рассеяния, линейные графики) для представления исторических тенденций; создавать простые скрипты для сбора данных с веб-страниц (веб-скрапинг) с использованием библиотек requests и BeautifulSoup при соблюдении

		этических и правовых норм. **Владеть:** методологией построения простого исследовательского пайплайна от исходных данных к результату с помощью последовательности скриптов; навыками документирования кода и создания README-файлов для исследовательских проектов; основами работы в системах контроля версий Git для учета изменений в коде и данных; навыками критической оценки возможностей и ограничений применяемых компьютерных методов для решения конкретных исторических вопросов.

1.3. Место дисциплины в структуре образовательной программы

Дисциплина «Программирование для гуманитарных наук» относится к элективным дисциплинам части, формируемой участниками образовательных отношений. Необходимы знания, умения и владения, сформированные в результате изучения дисциплины «Цифровые технологии в архивном деле. Электронные архивы». В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения последующих дисциплин как обязательной части учебного плана, так и части, формируемой участниками образовательных отношений.

2. Структура дисциплины

Общая трудоёмкость дисциплины составляет 3 з.е., 108 академических часа.

Структура дисциплины для очной формы обучения

Объем дисциплины в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Семестр	Тип учебных занятий	Количество часов
1	Лекции	14
1	Семинары	14
	Всего:	28

Объем дисциплины в форме самостоятельной работы обучающихся составляет 66 академических часов.

Структура дисциплины для очно-заочной формы обучения

Объем дисциплины в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Семестр	Тип учебных занятий	Количество часов

1	Лекции	12
1	Семинары	12
	Всего:	24

Объем дисциплины в форме самостоятельной работы обучающихся составляет 84 академических часа.

Структура дисциплины для заочной формы обучения

Объем дисциплины в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Семестр	Тип учебных занятий	Количество часов
1	Лекции	6
1	Семинары	2
2	Семинары	4
	Всего:	12

Объем дисциплины в форме самостоятельной работы обучающихся составляет 52 академических часа.

3. Содержание дисциплины

[Раздел 1: Введение в алгоритмическое мышление и основы Python]

Тема 1.1. Компьютерное мышление для гуманитария. От исследовательского вопроса к алгоритму.

Аннотация: Введение в дисциплину. Преодоление «цифрового барьера». Роль программирования в современном историческом исследовании: автоматизация, анализ, моделирование. Понятие алгоритма. Примеры алгоритмов в исторической работе (атрибуция источника, классификация, поиск паттернов). Среды разработки (IDE) для Python: установка, настройка, первый запуск. Переменные, базовые типы данных (целые числа, числа с плавающей точкой, строки, булевы значения), операторы ввода/вывода.

Тема 1.2. Управляющие конструкции: ветвление и циклы.

Аннотация: Логические выражения и операторы сравнения. Конструкция условного оператора if-elif-else. Практическое применение для принятия решений в коде (например, классификация событий по временным периодам). Циклы for и while. Итерация по последовательностям. Применение циклов для обработки множества однотипных объектов (списка архивных дел, коллекции оцифрованных писем). Понимание и предотвращение бесконечных циклов.

Тема 1.3. Структуры данных I: Списки и кортежи.

Аннотация: Понятие структуры данных. Список (list) как основная упорядоченная изменяемая коллекция. Методы работы со списками: добавление, удаление, поиск, сортировка. Срезы (slicing). Практикум: создание и обработка списка исторических дат, имен, артефактов. Кортеж (tuple) как неизменяемая упорядоченная коллекция. Его применение для хранения константных данных (координаты, неизменяемые параметры модели).

Тема 1.4. Структуры данных II: Словари и множества.

Аннотация: Словарь (dict) как структура «ключ-значение». Его фундаментальная важность для гуманитария: моделирование свойств объекта (личность: дата рождения, род деятельности,

библиография), создание индексов, частотных словарей. Методы перебора словаря. Множество (set) для работы с уникальными элементами (уникальные термины в тексте, множества участников событий). Операции над множествами (объединение, пересечение).

[Раздел 2: Функции, модульность и работа с файлами]

Тема 2.1. Функции и модульная организация кода.

Аннотация: Понятие функции как способа структурирования кода и избегания повторений. Определение функции (def), параметры, возвращаемое значение. Практикум: создание функций для типовых операций (расчет века по году, нормализация имени, проверка формата даты). Концепция модульности. Импорт стандартных модулей (math, random). Создание собственного модуля с утилитами для исторического исследования.

Тема 2.2. Работа с текстовыми файлами. Кодировки.

Аннотация: Чтение из и запись в текстовые файлы. Контекстный менеджер with. Проблема кодировок (UTF-8, CP1251) при работе с историческими текстами на русском языке. Практикум: чтение текста источника из файла, разбиение на строки и слова, подсчет базовой статистики (количество предложений, слов). Запись результатов анализа в отдельный файл-отчет.

Тема 2.3. Обработка структурированных данных: CSV и JSON.

Аннотация: Формат CSV (Comma-Separated Values) как мост между Excel/БД и программированием. Модуль csv для чтения и записи. Практикум: анализ данных переписи населения, базы военных потерь. Формат JSON как современный стандарт для хранения и передачи структурированных данных. Его использование в веб-API исторических проектов. Модуль json для сериализации и десериализации данных.

[Раздел 3: Продвинутые инструменты для анализа исторических данных]

Тема 3.1. Регулярные выражения (regex) для анализа текстов.

Аннотация: Введение в регулярные выражения как мощный инструмент поиска паттернов в текстах. Базовый синтаксис: символьные классы, квантификаторы, группы. Модуль re в Python. Практикум: поиск и извлечение дат в различных форматах из текста хроник, выделение имен собственных, очистка текста от посторонней разметки.

Тема 3.2. Введение в библиотеку Pandas для анализа табличных данных.

Аннотация: Библиотека Pandas как основной инструмент Data Science. Понятие DataFrame – «электронная таблица» в коде. Загрузка данных из CSV/Excel. Базовые операции: просмотр, выборка столбцов и строк (индексация, фильтрация). Группировка и агрегация данных (groupby). Практикум: анализ динамики социальных показателей по годам, сравнение данных по регионам.

Тема 3.3. Визуализация данных с Matplotlib и Seaborn.

Аннотация: Важность визуализации для презентации исторических результатов. Библиотека matplotlib. Построение линейных графиков, столбчатых диаграмм и гистограмм. Настройка подписей, легенд, оформления. Библиотека seaborn для создания более сложных и статистически ориентированных графиков. Практикум: визуализация динамики цен, численности населения, результатов голосований.

Тема 3.4. Основы веб-скрапинга для историка.

Аннотация: Этические и правовые аспекты сбора данных из интернета. Модуль requests для получения HTML-страниц. Парсинг HTML с помощью библиотеки BeautifulSoup. Поиск элементов по тегам, классам, идентификаторам. Извлечение структурированной информации с веб-страниц архивов, библиотек, музеев. Практикум: создание локального каталога оцифрованных документов с метаданными.

[Раздел 4: Взаимодействие с базами данных и основы ООП]

Тема 4.1. Основы SQL и работа с базами данных из Python.

Аннотация: Введение в реляционные базы данных. Язык SQL: оператор SELECT, фильтрация (WHERE), сортировка (ORDER BY), простое объединение таблиц (JOIN). Встроенная СУБД SQLite. Модуль sqlite3 в Python. Подключение к базе, выполнение запросов, получение результатов. Практикум: запросы к учебной базе данных «Исторические персоналии» или «Архивные фонды».

Тема 4.2. Объектно-ориентированное программирование (ООП) для моделирования исторических объектов.

Аннотация: Концепции ООП: класс, объект, атрибут, метод. Класс как способ моделирования сущности предметной области. Практикум: создание классов HistoricalFigure (историческая личность), Document (документ), Event (событие) с атрибутами и методами (например, расчет возраста на момент события). Демонстрация как ООП улучшает структуру и читаемость сложного исследовательского кода.

Тема 4.3. Заключение. Интеграция навыков в исследовательский пайплайн.

Аннотация: Обобщение пройденного материала. Построение полного исследовательского пайплайна на сквозном примере: от загрузки сырых данных (CSV/веб) -> их очистки и предобработки (Pandas, regex) -> анализа и визуализации (Pandas, Matplotlib) -> сохранения результатов и модели (JSON, классы). Обзор направлений для дальнейшего самостоятельного изучения (нейросети, анализ социальных сетей, GIS).

4. Образовательные технологии

Для проведения учебных занятий по дисциплине используются различные образовательные технологии. Для организации учебного процесса может быть использовано электронное обучение и (или) дистанционные образовательные технологии.

5. Оценка планируемых результатов обучения

5.1 Система оценивания

Текущий контроль

При оценивании докладов и участия в дискуссии на семинаре (максимальная оценка – 4 баллов) учитываются:

- степень раскрытия содержания материала (2 балла);
- изложение материала (грамотность речи, точность использования терминологии и символики, логическая последовательность изложения материала (1 балл);
- знание теории изученных вопросов, сформированность и устойчивость используемых при ответе умений и навыков (1 балла).

При оценивании результатов критического анализа текста исторических источников (максимальная оценка – 4 балла) учитывается:

- основательность проведённой критики источника (1 балл);
- уровень понимания извлечённой из текста источника информации (2 балла);
- грамотность и логичность изложения аналитических суждений (1 балл).

При оценивании исторического эссе (максимальная оценка – 20 баллов) учитывается:

- уровень использования научно-исследовательской литературы по теме (6 баллов);
- самостоятельность и аргументированность рассуждения по центральной проблеме эссе (10 баллов);

- грамотность и логичность письменного текста (4 балла).

Промежуточная аттестация (зачет)

При проведении промежуточной аттестации студент должен ответить на 2 вопроса теоретического характера.

При оценивании ответа на каждый из теоретических вопросов учитывается:

- полнота и правильность ответа (4-5 баллов за каждый из вопросов);
- аргументированность выводов (3-4 балла за каждый из вопросов);
- уровень понимания учебного материала (5-6 баллов за каждый из вопросов);
- грамотность и логичность изложения материала (4-5 баллов за каждый из вопросов).

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

<table>
<tr><td>100-балльная шкала</td><td colspan="2">Традиционная шкала</td><td>Шкала ECTS</td></tr>
<tr><td>95 – 100</td><td rowspan="2">отлично</td><td rowspan="5">зачтено</td><td>A</td></tr>
<tr><td>83 – 94</td><td>B</td></tr>
<tr><td>68 – 82</td><td>хорошо</td><td>C</td></tr>
<tr><td>56 – 67</td><td rowspan="2">удовлетворительно</td><td>D</td></tr>
<tr><td>50 – 55</td><td>E</td></tr>
<tr><td>20 – 49</td><td rowspan="2">неудовлетворительно</td><td rowspan="2">не зачтено</td><td>FX</td></tr>
<tr><td>0 – 19</td><td>F</td></tr>
</table>

5.2 Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	**Оценка по дисциплине**	**Критерии оценки результатов обучения по дисциплине**
100-83/ A,B	отлично	Выставляется обучающемуся, если он глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации. Обучающийся исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения. Свободно ориентируется в учебной и профессиональной литературе. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».
82-68/ C	хорошо	Выставляется обучающемуся, если он знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей. Обучающийся правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами. Достаточно хорошо ориентируется в учебной и профессиональной литературе. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
67-50/ D,E	удовлетво-рительно	Выставляется обучающемуся, если он знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации. Обучающийся испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами. Демонстрирует достаточный уровень знания учебной литературы по дисциплине. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».
49-0/ F,FX	неудовлет-ворительно	Выставляется обучающемуся, если он не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации. Обучающийся испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами. Демонстрирует фрагментарные знания учебной литературы по дисциплине. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.

5.3 Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

Контрольные вопросы для промежуточной аттестации в форме зачета:

1. Алгоритмическое мышление в гуманитарных науках: понятие, значение, примеры.
2. Базовые типы данных в Python. Их применение для хранения информации исторического характера.
3. Управляющие конструкции: ветвление (if-elif-else) и циклы (for, while). Примеры использования для анализа исторических данных.
4. Структуры данных в Python: списки, кортежи, словари, множества. Сравнительная характеристика и примеры использования в исследовательских задачах.
5. Функции в Python: назначение, синтаксис, параметры. Принцип модульности кода.
6. Организация работы с внешними файлами в Python. Чтение и запись текстовых файлов, проблема кодировок.
7. Обработка структурированных данных: форматы CSV и JSON. Библиотеки для работы с ними.
8. Регулярные выражения (regex): понятие, базовый синтаксис, применение для анализа текстовых исторических источников.
9. Библиотека Pandas: структура DataFrame, основные операции по загрузке, фильтрации и агрегации данных.

10. Визуализация данных в Python. Библиотеки Matplotlib/Seaborn, типы графиков для представления исторических процессов.
11. Основы веб-скрапинга. Этические нормы. Использование библиотек requests и BeautifulSoup для сбора данных.
12. Взаимодействие Python с базами данных. Основы SQL (SELECT, WHERE, JOIN) и использование модуля sqlite3.
13. Принципы объектно-ориентированного программирования (ООП). Моделирование исторических объектов (личность, документ, событие) с помощью классов.
14. Интеграция различных инструментов: построение исследовательского пайплайна от сырых данных до визуализации результатов.
15. Критическая рефлексия: возможности и ограничения методов программирования в исторических исследованиях.

Оценочные средства (материалы) для текущего контроля успеваемости

Вопросы закрытого типа (с одним верным вариантом ответа):

1. **Оператор print() в Python используется для:**
   а) получения ввода от пользователя
   б) объявления новой переменной
   в) вывода данных на экран
   г) написания комментариев в коде
2. **Какой тип данных в Python используется для хранения целых чисел?**
   а) string
   б) int
   в) float
   г) list
3. **Для создания упорядоченной изменяемой коллекции элементов в Python используется:**
   а) tuple
   б) list
   в) dict
   г) set
4. **Какой цикл используется, когда количество итераций известно заранее?**
   а) for
   б) while

в) if

г) elif

5. **Ключевое слово для определения собственной функции в Python:**

   а) function

   б) defun

   в) def

   г) lambda

6. **Какой метод словаря dict возвращает список всех ключей?**

   а) .values()

   б) .keys()

   в) .items()

   г) .get()

7. **Для открытия файла data.txt на чтение с автоматическим закрытием используется конструкция:**

   а) file = open("data.txt", "r")

   б) with open("data.txt", "r") as file:

   в) open("data.txt").read()

   г) file.open("data.txt")

8. **Библиотека Pandas используется в основном для:**

   а) создания веб-сайтов

   б) написания игр

   в) анализа и обработки табличных данных

   г) работы с компьютерным зрением

9. **Для создания DataFrame из CSV-файла в Pandas используется функция:**

   а) pd.read_excel()

   б) pd.read_csv()

   в) pd.from_csv()

   г) pd.open_csv()

10. **Какой SQL-запрос используется для выбора всех столбцов из таблицы persons?**

    а) SELECT * FROM persons;

    б) GET * FROM persons;

    в) FIND ALL IN persons;

    г) EXTRACT * FROM persons;

Вопросы открытого типа (на размышление и понимание):

1. Объясните разницу между списком (list) и кортежем (tuple) в Python. Приведите пример из исторического исследования, где уместнее использовать каждый из них.
2. Опишите алгоритм (можно словами или псевдокодом) программы, которая подсчитывает частоту упоминания различных географических названий в тексте дневника путешественника XIX века.
3. В чем заключается проблема кодировок (например, UTF-8 vs CP1251) при работе с текстовыми файлами на русском языке в Python? Как ее корректно решить?
4. Представьте, что у вас есть CSV-файл со столбцами Year, City, Population. С помощью библиотеки Pandas, какой последовательностью действий вы найдете год, в котором население выбранного города было максимальным?
5. Что такое регулярные выражения (regex)? Приведите пример шаблона для поиска в тексте дат формата "DD.MM.YYYY".
6. Объясните, как функция в Python повышает переиспользуемость кода. Приведите пример функции, полезной для историка (например, функция для перевода даты из старого стиля в новый).
7. В чем заключается основное преимущество использования словаря (dict) для хранения информации об исторической личности по сравнению с использованием нескольких отдельных переменных?
8. Что такое веб-скрапинг? Перечислите основные этические ограничения, которые должен учитывать исследователь при сборе данных с веб-сайтов.
9. Зачем историку, изучающему социальные сети прошлого (например, эпистолярные сети), могут понадобиться знания об объектно-ориентированном программировании (ООП)?
10. Опишите простой исследовательский пайплайн (последовательность шагов) от скачанного архива с текстовыми файлами писем до получения графика динамики объема корреспонденции по годам.

6. Учебно-методическое и информационное обеспечение дисциплины

6.1 Список источников и литературы

Основная литература:

1. **Маккинни, У. Python и анализ данных** / У. Маккинни. – СПб.: Питер, 2020. – 482 с. (Или более поздние издания).
2. **Лутошкин, Д.В. Python для гуманитариев: учебное пособие** / Д.В. Лутошкин. – М.: Издательский дом ВШЭ, 2021. – 256 с.
3. **Чэнь, Э. Интерактивный учебник по Python** / Э. Чэнь. – М.: ДМК Пресс, 2019. – 370 с.

4. **Седжвик, Р. Программирование на Python: учебный курс** / Р. Седжвик, К. Уэйн, Р. Дондеро. – СПб.: Питер, 2021. – 736 с.
5. **Головач, В.В. Визуализация данных с помощью Python и библиотек Matplotlib и Seaborn** / В.В. Головач. – М.: ДМК Пресс, 2022. – 304 с.

Дополнительная литература:

1. **Подольский, В.И. Цифровая история: введение в методы и технологии** / В.И. Подольский, А.И. Русаков. – М.: Аспект Пресс, 2022. – 288 с. (Главы, посвященные анализу данных).
2. **Мэтиз, Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения**/ Э. Мэтиз. – СПб.: Питер, 2021. – 704 с.
3. **Доусон, М. Программируем на Python** / М. Доусон. – СПб.: Питер, 2020. – 416 с.
4. **Ramalho, L. Fluent Python: Clear, Concise, and Effective Programming** / L. Ramalho. – O'Reilly Media, 2022. – 1016 p.
5. **Вандер Плас, Дж. Python для сложных задач: наука о данных и машинное обучение** / Дж. Вандер Плас. – СПб.: Питер, 2020. – 576 с.

Интернет-ресурсы:

1. Официальная документация по Python: **https://docs.python.org/3/**
2. Документация библиотеки Pandas: **https://pandas.pydata.org/docs/**
3. W3Schools. Python Tutorial: **https://www.w3schools.com/python/**
4. Репозиторий учебных материалов и датасетов по Digital Humanities: **https://programminghistorian.org/ (разделы на русском языке).**
5. Платформа для написания и хранения кода с поддержкой Jupyter Notebooks (для выполнения практических заданий): **https://colab.research.google.com/**

6.2 Профессиональные базы данных и информационно-справочные системы

Доступ к профессиональным базам данных: https://www.rsuh.ru/liber/resources.php

Информационные справочные системы:
1. Консультант Плюс
2. Гарант

7. Материально-техническое обеспечение дисциплины

Для проведения аудиторных занятий необходим стандартный набор специализированной учебной мебели и учебного оборудования, в том числе аудиторная доска (с магнитной поверхностью и набором приспособлений для крепления демонстрационных материалов), экран (на штативе или навесной). Для проведения семинаров, а также организации самостоятельной работы студентов необходим компьютерный класс с рабочими местами, обеспечивающими выход в Интернет. Кроме того, для информационно-ресурсного обеспечения семинаров необходим доступ к сканеру, копировальному аппарату и принтеру.

Реализация учебной программы должна обеспечиваться доступом каждого студента к информационным ресурсам – университетскому библиотечному фонду и сетевым ресурсам Интернет. Для использования ИКТ в учебном процессе необходимо наличие программного обеспечения, позволяющего осуществлять поиск информации в сети Интернет, систематизацию, анализ и презентацию информации, экспорт информации на цифровые носители.

Состав программного обеспечения:
1. Windows
2. Microsoft Office
3. Adobe Master Collection
4. Kaspersky Endpoint Security

Информационные справочные системы:
1. Консультант Плюс
2. Гарант

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением или могут быть заменены устным ответом; обеспечивается индивидуальное равномерное освещение не менее 300 люкс; для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; письменные задания оформляются увеличенным шрифтом; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.
- для глухих и слабослышащих: лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования; письменные задания выполняются на компьютере в письменной форме; экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.
- для лиц с нарушениями опорно-двигательного аппарата: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих: в печатной форме увеличенным шрифтом, в форме электронного документа, в форме аудиофайла.
- для глухих и слабослышащих: в печатной форме, в форме электронного документа.
- для обучающихся с нарушениями опорно-двигательного аппарата: в печатной форме, в форме электронного документа, в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих: устройством для сканирования и чтения с камерой SARA CE; дисплеем Брайля PAC Mate 20; принтером Брайля EmBraille ViewPlus;
- для глухих и слабослышащих: автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих; акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата: передвижными, регулируемыми эргономическими партами СИ-1; компьютерной техникой со специальным программным обеспечением.

9. Методические материалы

9.1 Планы семинарских занятий

Общие методические рекомендации для подготовки к семинарским занятиям: Семинары носят сугубо практический характер. Каждое занятие предполагает выполнение ряда заданий на компьютере. Студенты заранее знакомятся с теоретическим материалом лекций. Во время семинара они пишут код под руководством преподавателя, который выполняет роль ментора, помогая решать возникающие проблемы. Обязательно ведется работа с ошибками (разбор типичных синтаксических и логических ошибок). Часть заданий выполняется индивидуально, часть – в мини-группах с последующим разбором решений у доски (виртуальной доски кода). Оцениваются активность на занятии, выполнение обязательных заданий и представление решений.

[Тема 1: Первые программы. Переменные, ввод/вывод, арифметика.]

Задания:

1. Написать программу, которая запрашивает у пользователя год рождения исторической личности и вычисляет, сколько лет ей было в заданном году (например, в год начала Первой мировой войны).
2. Написать программу-калькулятор для перевода верст в километры по заданному коэффициенту.
3. Модифицировать программу (1), чтобы она выводила не только возраст, но и век, в котором персонаж родилась.

[Тема 2: Условные операторы и циклы.]

Задания:

1. Написать программу, которая классифицирует историческое событие по веку (ввод года) и выводит соответствующий период (Античность, Средневековье, Новое время и т.д.) по упрощенной схеме.
2. Написать программу, которая считывает N чисел (годы) и подсчитывает, сколько из них попадает в заданный исторический период (например, в «Длинный XIX век»).
3. Используя цикл, сгенерировать список лет от начала до конца правления монарха и вывести его.

[Тема 3-4: Работа со списками и словарями.]

Задания:

1. Создать список с именами членов исторической ассамблеи. Реализовать операции: добавление нового имени, удаление по индексу, сортировка в алфавитном порядке.
2. Создать словарь для исторической личности с ключами name, birth_year, death_year, occupation. Написать функцию, которая принимает такой словарь и возвращает продолжительность жизни.
3. На основе списка словарей (заранее подготовленного файла или введенного в код) найти самого долгоживущего персонажа.

[Тема 5: Функции и работа с текстовыми файлами.]

Задания:

1. Написать функцию, которая принимает строку (предложение из хроники) и возвращает список слов в ней.
2. Считать текст из файла chronicle.txt. Используя функцию из задания 1, подсчитать общее количество слов и уникальных слов в тексте.
3. Записать полученную статистику (общее кол-во слов, кол-во уникальных слов) в новый файл report.txt.

[Тема 6: Обработка CSV и JSON.]

Задания:

1. Имеется CSV-файл population.csv (город, год, население). Написать скрипт, который считывает его и выводит население заданного города в заданном году.
2. Преобразовать данные из CSV-файла в список словарей. Сохранить этот список в формате JSON в файл population.json.
3. Написать скрипт, который читает population.json и находит город с максимальным населением за весь период наблюдений.

[Тема 7-8: Регулярные выражения для анализа текста.]

Задания:

1. Найти и вывести все даты в формате DD.MM.YYYY в предоставленном тексте исторического документа.
2. Найти все упоминания денежных сумм (вида «10 рублей», «5 золотых») в тексте.
3. Очистить текст от цифр и знаков препинания, оставив только слова, разделенные пробелами.

[Тема 9-10: Анализ данных с Pandas.]

Задания:

1. Загрузить датасет с демографическими показателями (столбцы: Region, Year, Population, Literacy_Rate). Вывести первые 5 строк.
2. Отфильтровать данные только для одного региона. Построить линейный график изменения населения по годам.

3. Сгруппировать данные по году и найти средний уровень грамотности по всем регионам для каждого года.

[Тема 11-12: Визуализация результатов.]

Задания:

1. По данным из предыдущего семинара построить гистограмму распределения населения по регионам для выбранного года.
2. Создать scatter plot (диаграмму рассеяния), показывающий зависимость уровня грамотности от населения для всех записей.
3. Оформить графики: добавить названия осей, заголовок, легенду, изменить цвет и стиль линий.

[Тема 13-14: Основы веб-скрапинга (ознакомительное занятие).]

Задания (на учебном HTML-файле, имитирующем страницу архива):

1. С помощью BeautifulSoup извлечь все заголовки (<h2>) со страницы.
2. Извлечь текст всех абзацев (<p>).
3. Найти и извлечь все ссылки (<a href="...">) на странице.

[Тема 15-16: Работа с базой данных SQLite.]

Задания:

1. Подключиться к предоставленной базе данных history.db с таблицей books (id, title, author, year).
2. Написать SQL-запрос для выборки всех книг, изданных после 1900 года.
3. Написать Python-скрипт, который выполняет этот запрос и выводит результат в виде аккуратно отформатированного списка.

9.2 Методические рекомендации по подготовке письменных работ

Финальной письменной работой по дисциплине является **небольшой исследовательский проект (скрипт)**, сопровождаемый пояснительной запиской. Проект должен решать конкретную, пусть и небольшую, задачу в области исторических исследований с применением изученных инструментов программирования.

Структура пояснительной записки к проекту:

- **Введение:** Формулировка исследовательского вопроса или практической задачи (например, «Анализ динамики упоминаний политических партий в газетных заголовках 1917 года», «Созведение каталога писем из архива с автоматическим извлечением метаданных», «Визуализация миграционных потоков по данным переписи»).
- **Обзор инструментов и данных:** Описание используемых исходных данных (формат, источник, объем). Обоснование выбора инструментов (библиотеки Python, методы).
- **Основная часть (описание кода):** Пошаговое описание логики скрипта: загрузка данных, их предобработка, ключевые этапы анализа, визуализация. Приводятся наиболее важные фрагменты кода с комментариями.
- **Результаты и выводы:** Представление итогов работы программы (таблицы, графики, сформированные файлы). Интерпретация результатов с точки зрения поставленного исследовательского вопроса. Обсуждение ограничений и возможных улучшений.
- **Приложение:** Полный текст скрипта в виде файла .py или ссылка на репозиторий (например, GitHub). Скриншоты выполнения (опционально).

Критерии оценки: работоспособность кода, соответствие задачи возможностям дисциплины, ясность изложения в записке, сложность и оригинальность решения, оформление кода (комментарии, читаемость).
Объем пояснительной записки: 5-8 страниц. Объем кода: не менее 150-200 значимых строк.

9.3 Методические указания для обучающихся по освоению дисциплины

Курс построен по принципу «от простого к сложному» и сочетает лекционное изложение теории с интенсивной практикой на семинарах. Успешное освоение дисциплины возможно **только при регулярном самостоятельном написании кода**.
Лекции дают системное понимание концепций, синтаксиса и возможностей инструментов. Конспектирование лекций должно сопровождаться маркировкой моментов, требующих практической отработки.
Семинары являются ключевой формой работы. Необходимо приходить на семинар, предварительно ознакомившись с соответствующей лекционной темой. Активно участвуйте в разборе примеров, не бойтесь задавать вопросы при возникновении ошибок. Выполняйте все предложенные на занятии задания.
Самостоятельная работа:

1. **Практика:** Выделяйте время между занятиями на повторение и закрепление материала. Решайте дополнительные задачи, модифицируйте код с семинаров под собственные учебные сценарии.
2. **Работа с ошибками:** Ошибка (error) — основной инструмент обучения программированию. Внимательно читайте сообщения об ошибках в консоли, учитесь их понимать и исправлять.
3. **Документация и поиск:** Прививайте навык самостоятельного поиска информации. Учитесь пользоваться официальной документацией (docs.python.org), Stack Overflow, тематическими форумами. Формулируйте вопросы четко, приводя пример кода и текст ошибки.
4. **Подготовка проекта:** Начните думать над темой итогового проекта заранее. Постепенно пробуйте применять пройденные модули к своим данным или предложенным учебным датасетам.

Контроль: Текущий контроль осуществляется через активность на семинарах и выполнение заданий. Промежуточный контроль — защита итогового проекта. Регулярность практики напрямую влияет на формирование устойчивых навыков и итоговую оценку.

Приложение 1. Аннотация
рабочей программы дисциплины

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Цель дисциплины «Программирование для гуманитарных наук»: сформировать у студентов-гуманитариев базовые практические навыки алгоритмического мышления и программирования, необходимые для автоматизации задач обработки исторических данных, создания простых инструментов анализа и понимания принципов работы современных цифровых технологий в гуманитарных науках.

Задачи дисциплины:

- Дать представление об основах алгоритмизации, фундаментальных структурах данных и принципах структурного программирования.
- Ознакомить с синтаксисом и основными конструкциями языка программирования Python, как наиболее доступного и востребованного в научной среде.
- Научить применять полученные знания для решения типовых задач гуманитария: обработки текстовых корпусов, анализа структурированных данных (CSV, JSON), визуализации результатов.
- Сформировать навыки самостоятельного поиска технических решений, чтения документации и интеграции готовых программных библиотек в исследовательский процесс.
- Показать связь между программированием и профессиональными задачами историка (работа с базами данных, анализ текстов, веб-скрапинг исторических источников).

В результате освоения дисциплины обучающийся должен:

- **Знать:** базовый синтаксис Python, основные структуры данных, принципы работы с файлами и ключевые библиотеки для анализа данных (Pandas, Matplotlib), основы SQL, принципы анализа данных (Data Science), регулярные выражения, концепции объектно-ориентированного программирования.
- **Уметь:** писать скрипты для автоматизации обработки текстов и табличных данных, строить простые графики, запросы к базам данных из Python, очищать и анализировать текстовые данные, проводить простой веб-скрапинг, моделировать исторические объекты с помощью классов.

- **Владеть:** навыками работы в среде разработки, установки пакетов и базовой отладки кода, методологией построения исследовательского пайплайна, основами работы с Git, навыками критической оценки применяемых компьютерных методов.

Дисциплина носит выраженный практико-ориентированный характер и направлена на преодоление «цифрового барьера», обеспечивая студентов инструментарием для самостоятельного применения методов цифровых гуманитарных наук в их будущей профессиональной и исследовательской деятельности.