

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«07» марта 2024 г.
Протокол №1

**Рабочая программа дисциплины (модуля)
«Разработка на Python. Углубленный курс»**

Направление подготовки: 02.03.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Разработка

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Срок освоения программы: 4 года

Год набора: 2024

Москва
2024

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения	4
3. Тематический план	6
4. Содержание дисциплины (модуля)	6
5. Учебно-методическое обеспечение	7
6. Материально-техническое обеспечение	7
7. Методические и оценочные материалы	9

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Разработка на Python. Углубленный курс» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 02.03.01 Математика и компьютерные науки, профиль Разработка, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) дает умение программирования, используемым в различных областях, таких как веб-разработка, анализ данных и машинное обучение. Освоение Python позволяет эффективно решать сложные задачи в профессиональной деятельности.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 Математика и компьютерные науки, профиль Разработка и входит в Блок 1, часть, формируемую участниками образовательных отношений как дисциплина по выбору.

Дисциплина (модуль) изучается на 1 курсе в 1 семестре.

Цель изучения дисциплины (модуля): формирование у студентов навыков программирования на языке Python, включая разработку, отладку и оптимизацию приложений.

Задачи изучения дисциплины (модуля) направлены на формирование у студентов следующий знаний, умений и навыков:

- знание популярных библиотек Python и их применение;
- знание принципов работы с изображениями, API и асинхронным кодом;
- знание основы профилирования, шифрования и WebSocket;
- умение работать с внешними библиотеками и API;
- умение обрабатывать изображения и данные;
- умение разрабатывать асинхронные скрипты и Telegram-ботов;
- навык создавать прикладные Python-программы с использованием внешних сервисов и асинхронных задач.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
УК-1.	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1.	Знает методы поиска и анализа информации в области разработки, основные принципы критической оценки источников информации и их релевантности
		УК-1.2.	Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем
		УК-1.3.	Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации
ОПК-1.	Способен консультировать и использовать фундаментальные знания в области математического анализа, комплексного и функционального анализа алгебры, аналитической геометрии, дифференциальной геометрии и топологии, дифференциальных уравнений, дискретной математики и математической логики, теории вероятностей, математической статистики и случайных процессов, численных методов, теоретической механики в профессиональной деятельности	ОПК-1.1.	Знает основные концепции и теории в области математического анализа и смежных дисциплин; методы и подходы, используемые в различных областях математики
		ОПК-1.2.	Умеет применять математические методы для решения профессиональных задач
		ОПК-1.3.	Имеет практический опыт разработки и реализации математических моделей в профессиональной деятельности
ПК-1.	Способен формулировать задачи с математической точностью, обосновывать утверждения строго и анализировать полученные результаты в области математики и компьютерных наук	ПК-1.1.	Знает методы и подходы к формулированию задач, а также основные принципы математического доказательства и анализа результатов.
		ПК-1.2.	Умеет корректно ставить и формулировать математические задачи, применять строгие методы доказательства и

			анализировать полученные результаты
		ПК-1.3.	Имеет опыт работы с задачами в области математики и компьютерных наук, включая применение математических методов для решения практических задач

3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Контактная работа		Контроль	Самостоятельная работа	
Лекции	Семинары (практические занятия)					
1	Среда разработки и работа с библиотеками		5		6	Домашнее задание Подготовка к семинару
2	Модули и пакеты, оптимизация кода		5		6	Домашнее задание Подготовка к семинару
3	Командная работа		5		8	Домашнее задание Подготовка к семинару
4	Работа с API		5		6	Домашнее задание Подготовка к семинару
5	Асинхронное программирование и WebSocket		5		6	Домашнее задание Подготовка к семинару
6	Проектная неделя		5		6	Домашнее задание Проект
	<i>Зачет</i>			8		
	<i>Итого:</i>		30	8	38	
	<i>Объем дисциплины (модуля) (в ак. ч.)</i>	76				
	<i>Объем дисциплины (модуля) (в зач. ед.)</i>	2				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Среда разработки и работа с библиотеками	Интегрированная среда разработки. Сторонние библиотеки. Библиотека Pillow. Шифрование и стеганография
2	Модули и пакеты, оптимизация кода	Модуль и пакеты. Публикация собственных пакетов. Профайлинг кода
3	Командная работа	Распределение ролей и задач в команде
4	Работа с API	Работа с API. API Google Sheet
5	Асинхронное программирование и WebSocket	Основы асинхронного программирования. Асинхронные задачи. WebSocket. Асинхронные боты в телеграм

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Чернышев, С. А. Основы программирования на Python : учебник для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567821>.

Дополнительная литература:

1. Гниденко, И. Г. Технологии и методы программирования : учебник для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 241 с. — (Высшее образование). — ISBN 978-5-534-18130-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/581329>.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование»	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		

AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Разработка на Python. Углубленный курс» в рамках текущего контроля успеваемости используются такие виды учебной работы, как практические занятия, проект и домашние задания, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Участие в семинаре – активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре студентам рекомендуется заранее ознакомиться с темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их. При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

Проект – это целенаправленная деятельность, имеющая определенные цели, задачи и временные рамки, в результате которой создается уникальный продукт или услуга.

Для успешной подготовки проекта рекомендуется следовать следующим рекомендациям:

- четко определите цель и задачи проекта, чтобы понимать, какой результат вы хотите достичь;
- составьте план работы, разбив проект на этапы с указанием сроков выполнения каждого из них;
- используйте разнообразные источники информации и инструменты для исследования темы, чтобы обеспечить качественную основу для вашего проекта;
- регулярно проверяйте прогресс и вносите коррективы в план, если это необходимо, чтобы оставаться на правильном пути к завершению проекта.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов

и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Разработка на Python. Углубленный курс»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме *зачета*, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Зачтено	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	Зачтено	
8	Отлично	Зачтено	
7	Хорошо	Зачтено	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы,
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
5	Удовлетворительно	Зачтено	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Дисциплина (модуль) «Разработка на Python. Углубленный курс» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашние задания	30%	12	Работа с данными, датасетами, решение математических задач.
Участие в семинаре	20%	12	Активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии

Активность	Вес	Количество	Описание
Проект	50%	1	Решение бизнес-кейса с опорой на данные, подготовка презентации.

Формула расчёта итоговой оценки по дисциплине (модулю) «Разработка на Python. Углубленный курс»: $0,3 \times \text{среднее за домашние задания} + 0,2 \times \text{среднее за семинары} + 0,5 \times \text{среднее за проект}$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные вопросы для подготовки к семинарам

Среда разработки и работа с библиотеками

1. Что такое интегрированная среда разработки (IDE) и какие ключевые функции она предоставляет для программистов?
2. Как установить и настроить популярную IDE, такую как PyCharm или Visual Studio Code, для работы с Python?
3. Что такое сторонние библиотеки в Python и как их устанавливать с помощью pip?
4. Какие преимущества и недостатки использования сторонних библиотек в проектах?
5. Как использовать библиотеку Pillow для обработки изображений в Python? Приведите пример кода для изменения размера изображения.
6. Что такое шифрование данных и какие алгоритмы шифрования поддерживаются в Python (например, с использованием cryptography)?
7. Объясните понятие стеганографии и как ее можно реализовать с помощью Pillow для скрытия данных в изображениях.

Модули и пакеты, оптимизация кода

1. Что такое модули и пакеты в Python, и как их импортировать в программу?
2. Как создать собственный модуль или пакет в Python и организовать его структуру?
3. Какие шаги необходимо выполнить для публикации собственного пакета на PyPI?
4. Что такое профайлинг кода и зачем он нужен при оптимизации программ?
5. Как использовать инструменты профайлинга, такие как cProfile в Python, для анализа производительности кода?
6. Какие распространенные проблемы с производительностью можно выявить с помощью профайлинга и как их исправить?
7. Приведите пример оптимизации кода с использованием профайлинга: как ускорить медленный цикл или функцию.

Командная работа

1. Почему важно распределение ролей в командной разработке программного обеспечения?
2. Какие типичные роли существуют в команде разработчиков (например, разработчик, тестировщик, менеджер проекта) и их обязанности?
3. Как эффективно распределить задачи в команде, чтобы избежать конфликтов и перегрузок?
4. Какие инструменты (например, Jira или Trello) помогают в управлении задачами и распределении ролей?
5. Как обеспечить коммуникацию в команде для успешного распределения задач?
6. Какие проблемы могут возникнуть при распределении ролей в команде и как их решать?

7. Приведите пример успешного распределения ролей в команде для проекта по разработке веб-приложения.

Работа с API

1. Что такое API и как он используется для взаимодействия между приложениями?
2. Как отправить HTTP-запрос к API с помощью библиотеки requests в Python?
3. Какие основные типы HTTP-методов (GET, POST, PUT, DELETE) используются при работе с API?
4. Как аутентифицироваться при работе с API (например, с использованием API-ключей или OAuth)?
5. Что такое Google Sheets API и как его настроить для доступа к таблицам Google?
6. Приведите пример кода для чтения данных из Google Sheets с помощью API в Python.
7. Как обрабатывать ошибки и исключения при работе с API, включая API Google Sheets?

Асинхронное программирование и WebSocket

1. Что такое асинхронное программирование и чем оно отличается от синхронного?
2. Как использовать async и await в Python для создания асинхронных функций?
3. Что такое асинхронные задачи (tasks) и как их запускать с помощью asyncio?
4. Как обрабатывать несколько асинхронных задач параллельно в Python?
5. Что такое WebSocket и как он отличается от обычных HTTP-соединений?
6. Приведите пример реализации WebSocket-сервера или клиента с использованием библиотеки websockets в Python.
7. Как создать асинхронного бота для Telegram с использованием aiogram и асинхронных задач? Приведите базовый пример кода.

Примерные домашние задания

Домашнее задание «Среда разработки и работа с библиотеками»

1. Установить и настроить интегрированную среду разработки (IDE) по выбору (например, PyCharm, VSCode).
2. Написать скрипт, который загружает и отображает изображение с помощью библиотеки Pillow.
3. Реализовать простой алгоритм шифрования текста (например, Цезарь) и расшифровки.
4. Спрятать текстовое сообщение в изображении с помощью стеганографии (Pillow).
5. Создать README с инструкциями по установке и использованию вашего кода.

Домашнее задание «Модули и пакеты, оптимизация кода»

1. Разбить существующий скрипт на модули и оформить пакет с **init.py**.
2. Опубликовать пакет в локальном репозитории или на GitHub с инструкцией по установке.
3. Использовать профайлер (cProfile или аналог) для анализа производительности кода.
4. Оптимизировать наиболее ресурсоёмкий участок кода и сравнить результаты.
5. Написать отчет с описанием проделанной работы и выводами.

Домашнее задание «Командная работа»

1. Определить роли в команде (например, разработчик, тестировщик, менеджер проекта).
2. Сформировать план распределения задач с использованием системы управления проектами (Trello, Jira и т.п.).
3. Организовать ежедневные короткие встречи (stand-up) и составить протоколы.
4. Совместно разработать небольшой модуль и провести код-ревью.
5. Подготовить итоговый отчет о командной работе, выявленных проблемах и способах их решения.

Примерное задание для проекта

Проект по теме: Работа с API (API Google Sheet)

Цель проекта:

Разработать приложение, которое автоматически собирает, обрабатывает и визуализирует данные из Google Sheets с помощью API.

Задачи проекта:

- Ознакомиться с документацией Google Sheets API.
- Настроить доступ и авторизацию к API через OAuth 2.0.
- Реализовать скрипт для чтения и записи данных в таблицу Google Sheets.
- Обработать полученные данные (например, фильтрация, агрегация).
- Создать визуализацию данных (например, графики с использованием matplotlib или plotly).
- Разработать простой интерфейс (CLI или веб), позволяющий пользователю запускать обновление данных и просматривать результаты.

Последовательность выполнения:

1. Изучение API Google Sheets и настройка проекта в Google Cloud Console.
2. Реализация авторизации и тестового запроса на чтение данных.
3. Разработка функций для чтения и записи данных.
4. Обработка данных и подготовка к визуализации.
5. Создание визуализаций и интеграция с интерфейсом.
6. Тестирование и отладка приложения.
7. Подготовка документации и презентации проекта.

Критерии оценивания:

- Полнота и корректность реализации функционала работы с API.
- Качество и удобство интерфейса пользователя.
- Корректность и информативность визуализаций.
- Чистота и структура кода, наличие комментариев.
- Документированность проекта (README, инструкции).
- Умение аргументировать технические решения и демонстрировать работу приложения.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Какой статус код выведет следующая программа (ответ — число)? <pre>import requests response = requests.get("https://httpbin.org/status/418") print(response.status_code)</pre>	418	УК-1
2.	Какое число выведет следующая программа (ответ — число)? <pre>import math print(math.sqrt(81))</pre>	9 / 9.0	ПК-1

3.	Какое ключевое слово языка Python используется для объявления асинхронной функции? A. def B. await C. coroutine D. async	D	ОПК-1
----	--	---	-------