

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Ястребов Олег Александрович  
Должность: Ректор  
Дата подписания: 13.05.2026 16:38:16  
Уникальный программный ключ:  
ca953a0120d891083f939673078ef1a989dae18a

**Федеральное государственное автономное образовательное учреждение высшего образования  
«Российский университет дружбы народов имени Патриса Лумумбы»**

**Институт экологии**

(наименование основного учебного подразделения (ОУП) – разработчика ОП ВО)

## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

### **ОСНОВЫ ПРОГРАММИРОВАНИЯ НА PYTHON**

(наименование дисциплины/модуля)

**Рекомендована МССН для направления подготовки/специальности:**

### **18.03.02 ЭНЕРГО- И РЕСУРСОСБЕРЕГАЮЩИЕ ПРОЦЕССЫ В ХИМИЧЕСКОЙ ТЕХНОЛОГИИ, НЕФТЕХИМИИ И БИОТЕХНОЛОГИИ**

(код и наименование направления подготовки/специальности)

**Освоение дисциплины ведется в рамках реализации основной профессиональной образовательной программы высшего образования (ОП ВО):**

### **ЭНЕРГО- И РЕСУРСОСБЕРЕГАЮЩИЕ ПРОЦЕССЫ В ХИМИЧЕСКОЙ ТЕХНОЛОГИИ, НЕФТЕХИМИИ И БИОТЕХНОЛОГИИ**

(наименование (профиль/специализация) ОП ВО)

## 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Дисциплина «Основы программирования на Python» входит в программу бакалавриата «Энерго- и ресурсосберегающие процессы в химической технологии, нефтехимии и биотехнологии» по направлению 18.03.02 «Энерго- и ресурсосберегающие процессы в химической технологии, нефтехимии и биотехнологии» и изучается в 3 семестре 2 курса. Дисциплину реализует Департамент экологической безопасности и менеджмента качества продукции. Дисциплина состоит из 4 разделов и 15 тем и направлена на изучение основных принципов программирования на языках высокого уровня.

Целью освоения дисциплины является подготовка студентов к эффективному использованию языка Python для решения задач информационной безопасности, этики и правовых норм, включая поиск, анализ и обработку данных, разработку алгоритмов, применение структур данных и контрольных конструкций, а также оценку достоверности информации и построение логических выводов.

## 2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины «Основы программирования на Python» направлено на формирование у обучающихся следующих компетенций (части компетенций):

Таблица 2.1. Перечень компетенций, формируемых у обучающихся при освоении дисциплины (результаты освоения дисциплины)

Шифр	Компетенция	Индикаторы достижения компетенции (в рамках данной дисциплины)
УК-12	Способен к взаимодействию в условиях современной информационной культуры и цифровой экономики с учетом требований информационной безопасности, этических и правовых норм	УК-12.1 Осуществляет поиск нужных источников информации и данных, воспринимает, анализирует, запоминает и передает информацию с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач; УК-12.2 Проводит оценку информации, ее достоверность, строит логические умозаключения на основании поступающих информации и данных;

## 3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина «Основы программирования на Python» относится к части, формируемой участниками образовательных отношений блока 1 «Дисциплины (модули)» образовательной программы высшего образования.

В рамках образовательной программы высшего образования обучающиеся также осваивают другие дисциплины и/или практики, способствующие достижению запланированных результатов освоения дисциплины «Основы программирования на Python».

Таблица 3.1. Перечень компонентов ОП ВО, способствующих достижению запланированных результатов освоения дисциплины

Шифр	Наименование компетенции	Предшествующие дисциплины/модули, практики*	Последующие дисциплины/модули, практики*
УК-12	Способен к взаимодействию в условиях современной информационной культуры и цифровой экономики с учетом требований информационной безопасности, этических и правовых норм	Цифровая грамотность;	Моделирование энерго- и ресурсосберегающих процессов; Основы энерго- и ресурсосбережения; Инфографика и технология презентаций**; Цифровые деловые коммуникации**;

<b>Шифр</b>	<b>Наименование компетенции</b>	<b>Предшествующие дисциплины/модули, практики*</b>	<b>Последующие дисциплины/модули, практики*</b>
			Искусственный интеллект в устойчивом развитии;

\* - заполняется в соответствии с матрицей компетенций и СУП ОП ВО

\*\* - элективные дисциплины /практики

#### 4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общая трудоемкость дисциплины «Основы программирования на Python» составляет «2» зачетные единицы.

Таблица 4.1. Виды учебной работы по периодам освоения образовательной программы высшего образования для очной формы обучения.

Вид учебной работы	ВСЕГО, ак.ч.		Семестр(-ы)
			3
Контактная работа, ак.ч	34		34
Лекции (ЛК)	0		0
Лабораторные работы (ЛР)	34		34
Практические/семинарские занятия (СЗ)	0		0
Самостоятельная работа обучающихся, ак.ч.	29		29
Контроль (экзамен/зачет с оценкой), ак.ч.	9		9
Общая трудоемкость дисциплины ак.ч.	ак.ч.	72	72
	зач.ед.	2	2

Общая трудоемкость дисциплины «Основы программирования на Python» составляет «2» зачетные единицы.

Таблица 4.2. Виды учебной работы по периодам освоения образовательной программы высшего образования для заочной формы обучения.

Вид учебной работы	ВСЕГО, ак.ч.		Семестр(-ы)
			3
Контактная работа, ак.ч	8		8
Лекции (ЛК)	0		0
Лабораторные работы (ЛР)	8		8
Практические/семинарские занятия (СЗ)	0		0
Самостоятельная работа обучающихся, ак.ч.	62		62
Контроль (экзамен/зачет с оценкой), ак.ч.	2		2
Общая трудоемкость дисциплины ак.ч.	ак.ч.	72	72
	зач.ед.	2	2

## 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Таблица 5.1. Содержание дисциплины (модуля) по видам учебной работы\*

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы*
Раздел 1	Знакомство с Python	1.1	Переменные. Ввод и вывод. Арифметические операции.	Понятие переменной, правила именования, динамическая типизация. Базовые типы данных: int, float, str. Функция print(): форматирование вывода (f-строки, метод format). Функция input(): чтение строки, преобразование в числа. Арифметические операторы (+, -, *, /, //, %, **), приоритет выполнения, преобразование типов (явное и неявное).	ЛР
		1.2	Списки	Определение списка как упорядоченной изменяемой коллекции. Создание: [], list(). Индексация (положительная и отрицательная), получение срезов (slice). Базовые операции: добавление (append, extend, insert), удаление (del, pop, remove), изменение элемента. Проверка вхождения (in), длина (len), поиск индекса (index).	ЛР
		1.3	Функции	Определение функции с def, параметры и аргументы. Возврат значения return. Локальные и глобальные переменные, область видимости. Передача аргументов: позиционные, именованные, значения по умолчанию. Простые примеры (вычисление площади, приветствие).	ЛР
Раздел 2	Основные операции Python	2.1	Оператор цикла while	Синтаксис while, условие продолжения. Бесконечные циклы, прерывание break. Пропуск итерации continue. Блок else при нормальном завершении. Примеры: вычисление суммы, факториала, ввод данных до маркера. Вложенные циклы while.	ЛР
		2.2	Оператор цикла for	Синтаксис for ... in. Итерация по строкам, спискам, кортежам, словарям. Функция range() для генерации числовых рядов. Перебор индексов через range(len()). Вложенные циклы. Управление: break, continue, else. Сравнение с while (когда что использовать).	ЛР
		2.3	Условный оператор if	Синтаксис if, elif, else. Логические выражения: сравнение (==, !=, <, >, <=, >=), операторы and, or, not. Проверка вхождения in. Вложенные условия. Типичные ошибки (присваивание = вместо сравнения ==).	ЛР
		2.4	Тернарный оператор	Синтаксис: значение_если_истина if условие else значение_если_ложь. Применение для простых присваиваний (например, определение максимума, условный вывод). Ограничения: не подходит для сложных блоков. Примеры замены коротких if-else.	ЛР
		2.5	Обработка исключительных ситуаций	Конструкция try/except. Перехват конкретных исключений: ValueError, ZeroDivisionError, TypeError. Блоки else (выполняется, если нет исключения) и finally (выполняется всегда). Генерация исключений через raise. Примеры: обработка ошибок ввода-вывода, преобразования типов.	ЛР
Раздел 3	Списки и кортежи в Python	3.1	Кортежи. Основные операции со списками и кортежами	Кортеж (tuple) как неизменяемая упорядоченная коллекция. Создание: (), tuple(). Отличия от списков. Общие операции: индексация, срезы, конкатенация (+), умножение (*), len(), in. Когда использовать кортежи (защита от изменений, ключи в словарях).	ЛР
		3.2	Создание выборки на основе списков и кортежей	Извлечение подмножества элементов с помощью срезов с шагом ([start:stop:step]). Генераторы списков ([выражение for элемент in последовательность if условие]). Аналоги для кортежей (обёртка в tuple()). Фильтрация и преобразование элементов. Примеры выборки чётных чисел, квадратов значений.	ЛР
		3.3	Вложенные списки и кортежи	Создание многомерных структур (список списков, список кортежей). Доступ к элементам через несколько индексов (matrix[i][j]). Итерация по вложенным структурам	ЛР

Номер раздела	Наименование раздела дисциплины	Наименование темы		Содержание темы	Вид учебной работы*
				(вложенные циклы for). Изменение элементов во вложенных списках (кортежи внутри остаются неизменяемыми). Практические примеры: матрицы, таблицы, хранение координат.	
		3.4	Копирование списков и кортежей	Отличие присваивания ( $a = b$ ) от копирования (создание нового объекта). Поверхностное копирование: срезы [:], list(), copy(). Глубокое копирование: модуль copy, функция deepcopy(). Проблема вложенных объектов (изменение внутреннего списка в копии). Примеры для списков и кортежей (кортежи не требуют глубокого копирования, если внутри неизменяемые типы).	ЛР
		3.5	Функции и методы для работы со списками	Методы списков: append, extend, insert, remove, pop, clear, index, count, sort (с ключом key), reverse. Встроенные функции: len, max, min, sum, sorted (возвращает новый список), reversed (итератор). Отличия методов от функций: изменение исходного списка vs создание нового. Примеры сортировки, подсчёта, удаления дубликатов.	ЛР
Раздел 4	Множества и словари	4.1	Множества. Операции с множествами и операции на множествах	Множество (set) как неупорядоченная коллекция уникальных элементов. Создание: {} (не пустой), set(). Добавление add, удаление remove (ошибка при отсутствии) и discard (без ошибки), clear. Теоретико-множественные операции: объединение (  или union), пересечение (&), разность (-), симметрическая разность (^). Проверка подмножества (<=, <). Применение: удаление дубликатов из списка, проверка общих элементов.	ЛР
		4.2	Словари. Операции со словарями	Словарь (dict) как отображение «ключ → значение». Создание: {}, dict(). Доступ по ключу: [] (вызывает исключение) и get (возвращает значение по умолчанию). Добавление и изменение элементов. Удаление: del, pop (возвращает значение), popitem (удаляет произвольную пару). Методы keys(), values(), items() для итерации. Проверка наличия ключа (in). Генераторы словарей. Примеры: подсчёт частоты символов, группировка данных.	ЛР

\* - заполняется только по ОЧНОЙ форме обучения: ЛК – лекции; ЛР – лабораторные работы; СЗ – практические/семинарские занятия.

## 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Таблица 6.1. Материально-техническое обеспечение дисциплины

Тип аудитории	Оснащение аудитории	Специализированное учебное/лабораторное оборудование, ПО и материалы для освоения дисциплины (при необходимости)
Компьютерный класс	Компьютерный класс для проведения занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная персональными компьютерами (в количестве 12 шт.), доской (экраном) и техническими средствами мультимедиа презентаций.	
Для самостоятельной работы	Аудитория для самостоятельной работы обучающихся (может использоваться для проведения семинарских занятий и консультаций), оснащенная комплектом специализированной мебели и компьютерами с доступом в ЭИОС.	

\* - аудитория для самостоятельной работы обучающихся указывается **ОБЯЗАТЕЛЬНО!**

## 7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### Основная литература:

1. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2022. — 210 с. — (Высшее образование). — ISBN 978-5-534-14638-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/492920> (дата обращения: 30.03.2025).

2. Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2022. — 286 с. — (Высшее образование). — ISBN 978-5-534-14350-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/496893> (дата обращения: 30.03.2025)

### Дополнительная литература:

1. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2022. — 235 с. — (Высшее образование). — ISBN 978-5-534-02816-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/489920> (дата обращения: 30.03.2025).

2. Черпаков, И. В. Основы программирования: учебник и практикум для вузов / И. В. Черпаков. — Москва : Издательство Юрайт, 2022. — 219 с. — (Высшее образование). — ISBN 978-5-9916-9983-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/489747> (дата обращения: 30.03.2025)

### Ресурсы информационно-телекоммуникационной сети «Интернет»:

1. ЭБС РУДН и сторонние ЭБС, к которым студенты университета имеют доступ на основании заключенных договоров

- Электронно-библиотечная система РУДН – ЭБС РУДН <https://mega.rudn.ru/MegaPro/Web>
- ЭБС «Университетская библиотека онлайн» <http://www.biblioclub.ru>
- ЭБС «Юрайт» <http://www.biblio-online.ru>
- ЭБС «Консультант студента» [www.studentlibrary.ru](http://www.studentlibrary.ru)
- ЭБС «Знаниум» <https://znanium.ru/>

2. Базы данных и поисковые системы

- Sage <https://journals.sagepub.com/>
- Springer Nature Link <https://link.springer.com/>
- Wiley Journal Database <https://onlinelibrary.wiley.com/>
- Наукометрическая база данных Lens.org <https://www.lens.org>

*Учебно-методические материалы для самостоятельной работы обучающихся при освоении дисциплины/модуля\*:*

1. Курс лекций по дисциплине «Основы программирования на Python».

\* - все учебно-методические материалы для самостоятельной работы обучающихся размещаются в соответствии с действующим порядком на странице дисциплины **в ТУИС!**

**РУКОВОДИТЕЛЬ ОП ВО**

Доцент

---

Должность

**РУКОВОДИТЕЛЬ БУП**

Директор департамента

---

Должность

**РАЗРАБОТЧИКИ**

Доцент

---

Должность

Харламова М.Д.

---

Фамилия И.О

Савенкова Е.В.

---

Фамилия И.О

Ледащева Т.Н.

---

Фамилия И.О