

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой
МО ЭВМ



Абрамов Г.В.

25.04.2022 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Б1.В.06 Введение в язык программирования Python

1. Код и наименование направления подготовки/специальности:

02.03.02 Фундаментальная информатика и информационные технологии

2. Профиль подготовки/специализация:

Инженерия программного обеспечения

3. Квалификация (степень) выпускника: бакалавр

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины: МО ЭВМ

6. Составители программы: Тютин Антон Павлович,
преподаватель

7. Рекомендована: НМС факультета ПММ, протокол № 8 от 15.04.2022

отметки о продлении вносятся вручную)

8. Учебный год: 2024/2025

Семестр(ы): 5

9. Цели и задачи учебной дисциплины:

Цель дисциплины

- получение теоретических знаний о языке программирования Python и формирование навыков его использования;
- формирование способности применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных

Задачи дисциплины:

- изучение синтаксиса языка Python.
- изучение инструментов, предназначенных для создания программного обеспечения с использованием языка Python.
- изучение инструментов, предназначенных для увеличения эффективности и оптимизации при работе с языком.
- знакомство со встроенными и внешними библиотеками языка, принципами их использования, а также создания собственных;
- формирование и развитие навыков использования в профессиональной деятельности объектно-ориентированных систем программирования.

10. Место учебной дисциплины в структуре ООП: Дисциплина «Введение в язык программирования Python» входит в часть, формируемую участниками образовательных отношений блока Б1.

Изучение данного курса должно базироваться на знаниях обучающимися материалов курсов «Информатика и программирование» и «Объектно-ориентированное программирование». Дисциплина является предшествующей для изучения курса «Алгоритмы биоинформатики».

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-6	Способность применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии	ПК-6.1	Использует в профессиональной деятельности объектно-ориентированные системы программирования	знать: – синтаксис и семантику языка программирования Python уметь: – решать базовые задачи создания ПО с использованием языка Python и его стандартных библиотек владеть (иметь навык(и)): технологией разработки ПО для решения производственных и исследовательских задач с использованием языка программирования Python

12. Объем дисциплины в зачетных единицах/час. — 3/108.

Форма промежуточной аттестации зачет.

13. Трудоемкость по видам учебной работы

Вид учебной работы		Трудоемкость	
		Всего	По семестрам
			5 семестр
Аудиторные занятия		64	64
в том числе:	лекции	32	32
	практические	0	0
	лабораторные	32	32
Самостоятельная работа		44	44
Форма промежуточной аттестации (зачет – 0 час. / экзамен – __ час.)		0	0
Итого:		108	108

13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК *
1. Лекции			
1.1	Знакомство Python	История создания языка. Общее описание Python. Преимущества и недостатки языка. Базовый интерактивный режим. Базовый синтаксис языка: переменные, присваивания, выражения, получение и вывод данных.	Edu.vsu.ru
1.2	Типы данных	Числовые типы данных. Логический тип bool, Строки. Значение None. Списки, кортежи, множества. Словари.	Edu.vsu.ru
1.3	Управляющие конструкции	Условные операторы if-elif-else. Циклы while и for. Блоки и отступы.	Edu.vsu.ru
1.4	Функции	Базовые определения функций. Типы аргументов и возвращаемых значений. Аргументы *args и **keywargs. Использование функций как объектов.	Edu.vsu.ru
1.5	Модули, пакеты и области видимости	Определение модуля. Команда import. Глобальная переменная __name__. Библиотечные и сторонние модули. Пространства имён и области видимости. Создание модуля. Создание пакета.	Edu.vsu.ru

1.6	Работа с файлами и файловой системой	Использование библиотеки os. Получение информации о файлах и каталогах. Открытие файлов для чтения и записи, закрытие файлов. Использование команды with.	Edu.vsu.ru
1.7	Обработка исключений	Понятие исключений. Стандартные исключения в Python. Определение новых исключений. Обработка исключений при помощи команд try-except-else-finally.	Edu.vsu.ru
1.8	Объектно-ориентированное программирование	Определение классов. Поля и методы. Конструкторы. Создание экземпляров. Наследование. Создание свойств (property). Множественное наследование.	Edu.vsu.ru
1.9	Декораторы	Определение декоратора. Декораторы функций. Декораторы классов. Примеры декораторов.	Edu.vsu.ru
1.10	Регулярные выражения	Определение и синтаксис регулярных выражений. Библиотеки re и rehex: извлечение совпадений из строк, замена текста с использованием регулярных выражений.	Edu.vsu.ru
1.11	Использование библиотек Python	Стандартная библиотека и получение документации. Установка внешних библиотек Python. Использование виртуальных окружений.	Edu.vsu.ru
1.12	Графические интерфейсы	Знакомство с библиотеками для построения графического интерфейса.	Edu.vsu.ru
2. Практические занятия			
3. Лабораторные работы			
3.1	Знакомство с инструментами разработки и базовым синтаксисом	Создание простого сценария с использованием переменных, управляющих конструкций и командами консольного ввода и вывода.	
3.2	Продвинутые типы данных и функции	Использование функций и типов коллекций (списки, массивы, кортежи, словари).	
3.3	Модули	Выделение функций и переменных в отдельные модули. Подключение и использование собственных модулей.	
3.4	Объектно-ориентированное программирование	Создание классов и использование наследования.	
3.5	Работа с файловой системой	Работа с файлами и файловой системой, обработка системных и собственных исключений.	
3.6	Библиотеки Python	Использование библиотек для работы с сетью и регулярными выражениями.	
3.7	Графические интерфейсы	Использование библиотек для создания графического интерфейса.	

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)			
		Лекции	Лабораторные	Самостоятельная работа	Всего
	Знакомство Python	2	2	2	6
	Типы данных	2	2	2	6

Управляющие конструкции

2

2

2

6

	Функции	2	2	2	6
	Модули, пакеты и области видимости	2	2	2	6
	Работа с файлами и файловой системой	4	4	4	12
	Обработка исключений	2	2	2	6
	Объектно-ориентированное программирование	6	6	8	20
	Декораторы	2	2	2	6
	Регулярные выражения	2	2	2	6
	Использование библиотек Python	2	2	10	14
	Графические интерфейсы	4	4	6	14
	Итого:	32	32	44	108

14. Методические указания для обучающихся по освоению дисциплины

Работа с конспектами лекций, выполнение лабораторных заданий.

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1	<i>Изучаем Python, 4-е издание / М. Лутц. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.</i>
2	<i>Python. Экспресс-курс. 3-е изд. / Н. Седек. – Пер. с англ. – СПб.: Питер, 2019. – 480 с.</i>

б) дополнительная литература:

№ п/п	Источник
3	<i>Регулярные выражения: основы. / М. Мицджеральд. – Пер. с англ. – М.: ООО «И.Д. Вильямс», 2015. – 144 с.</i>

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)*:

№ п/п	Ресурс
4	https://docs.python.org/3/ – Документация Python 3

* Вначале указываются ЭБС, с которыми имеются договора у ВГУ, затем открытые электронно-образовательные ресурсы

16. Перечень учебно-методического обеспечения для самостоятельной работы (учебно-методические рекомендации, пособия, задачки, методические указания по выполнению практических (контрольных) работ и др.)

№ п/п	Источник

17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение и информационно-справочные системы (при необходимости)

Для реализации учебного процесса используется бесплатная интегрированная среда разработки PyCharm Community или бесплатный редактор исходного кода Visual Studio Code с поддержкой расширений, а так же интерпретатор Python 3, распространяемый открыто.

18. Материально-техническое обеспечение дисциплины:

Лекционная аудитория должна быть оборудована учебной мебелью, компьютером, мультимедийным оборудованием (проектор, экран, средства звуковоспроизведения), допускается переносное оборудование.

Практические занятия должны проводиться в специализированной аудитории, оснащенной учебной мебелью и персональными компьютерами с доступом в сеть Интернет (компьютерные классы, студии), мультимедийным оборудованием (мультимедийный проектор, экран, средства звуковоспроизведения), Число рабочих мест в аудитории должно быть таким, чтобы обеспечивалась индивидуальная работа студента на отдельном персональном компьютере.

Для самостоятельной работы необходимы компьютерные классы, помещения, оснащенные компьютерами с доступом к сети Интернет.

Программное обеспечение:

- ОС Windows
- LibreOffice (свободное и/или бесплатное ПО)
- Microsoft Visual Studio Community Edition (свободное и/или бесплатное ПО)
- Adobe Reader (свободное и/или бесплатное ПО)

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1.	Разделы 1 - 10 Знакомство Python. Типы данных. Управляющие конструкции. Функции. Модули, пакеты и области видимости. Работа с файлами и файловой системой. Обработка исключений. Объектно-ориентированное программирование. Декораторы. Регулярные выражения.	ПК-6	ПК-6.1	Задания для лабораторных работ

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
2.	Разделы 1-12 Знакомство Python. Типы данных. Управляющие конструкции. Функции. Модули, пакеты и области видимости. Работа с файлами и файловой системой. Обработка исключений. Объектно-ориентированное программирование. Декораторы. Регулярные выражения. Использование библиотек Python. Графические интерфейсы.	ПК-6	ПК-6.1	Задания для лабораторных работ
3	Разделы 1-12 Знакомство Python. Типы данных. Управляющие конструкции. Функции. Модули, пакеты и области видимости. Работа с файлами и файловой системой. Обработка исключений. Объектно-ориентированное программирование. Декораторы. Регулярные выражения. Использование библиотек Python. Графические интерфейсы.	ПК-6	ПК-6.1	Задания для лабораторных работ
Промежуточная аттестация форма контроля - экзамен				<i>Перечень вопросов Практическое задание</i>

20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

Лабораторные работы

20.2 Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств:

Примеры лабораторных работ

1. Создать игру «Угадай число», в которой компьютер «загадывает» случайное число, а пользователь пытается его отгадать.
2. Создать класс Дата (Date), который хранит год, месяц, день, час, минуту и секунду. Реализовать в нем методы изменения даты на заданное количество секунд, минут, часов. Создать перегрузку оператора вычитания для определения промежутка времени между двумя датами. Создать дочерний класс Событие (Event), который дополнительно хранит описание, длительность и период, за который необходимо напоминание об этом событии. Реализовать метод вычисления даты напоминания.
3. Создать пакет Напоминания (reminders), в котором определить модули Дата (date) и Событие (event) из предыдущей лабораторной работы. Подключить пакет к основному исходному файлу и продемонстрировать возможности классов.
4. Используя пакет reminders, создать графический интерфейс, позволяющий отобразить список событий с возможностью добавления, редактирования и удаления событий.
5. Используя регулярные выражения, получить с заданного сайта курс запрашиваемой пользователем валюты.

Описание критериев и шкалы оценивания компетенций (результатов обучения) при промежуточной аттестации

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
<i>Все лабораторные работы выполнены в полной мере. Продемонстрирована самостоятельность в изучении технологий, актуальных практик и подходов к написанию кода и решению задач. Продемонстрировано полное понимание использованных технологий. Сформирован навык отладки созданных сценариев. Продемонстрировано использование актуальных инструментов, использованы виртуальные окружения.</i>	<i>Повышенный уровень</i>	<i>Зачет</i>
<i>Все лабораторные работы выполнены в полной мере, однако не всегда есть полное понимание принципов работы языка или технологий, используются неоптимальные решения, не соблюдается стиль кода, что является важной составляющей при работе с языком Python. Сформирован навык отладки созданных сценариев. Продемонстрировано использование актуальных инструментов.</i>	<i>Базовый уровень</i>	<i>Зачет</i>
<i>Все лабораторные работы выполнены, однако есть существенные недочеты в созданном продукте, достаточно большие пробелы в понимании используемых технологий и подходов. Поверхностное владение инструментами разработки, отсутствие понимания различий между ними.</i>	<i>Пороговый уровень</i>	<i>Зачет</i>
<i>Не все лабораторные работы выполнены и/или отсутствует понимание примененных в работах технологий, при сдаче работ ответы на вопросы отсутствуют.</i>	<i>–</i>	<i>Не зачтено</i>