



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И.
Носова»



УТВЕРЖДАЮ
Директор ИЭиАС
В.Р. Храмшин

03.02.2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

БИБЛИОТЕКИ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON

Направление подготовки (специальность)
09.04.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы
Программное обеспечение для цифровизации предприятий и организаций

Уровень высшего образования - магистратура

Форма обучения
очная

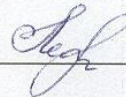
Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Вычислительной техники и программирования
Курс	1
Семестр	1

Магнитогорск
2026 год

Рабочая программа составлена на основе ФГОС ВО - магистратура по направлению подготовки 09.04.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 918)

Рабочая программа рассмотрена и одобрена на заседании кафедры
Вычислительной техники и программирования
29.01.2026, протокол № 7

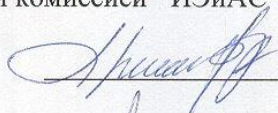
Зав. кафедрой



О.С. Логунова

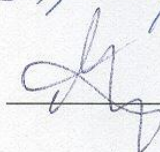
Рабочая программа одобрена методической комиссией ИЭиАС
03.02.2026 г. протокол № 5

Председатель



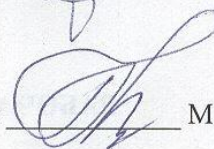
В.Р. Храмшин

Рабочая программа составлена:
доцент кафедры ВТиП,



Я.А. Григорьев

Рецензент:
директор НИИ «Промбезопасность», д-р техн. наук



М.Ю. Наркевич

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2027 - 2028 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2028 - 2029 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

1 Цели освоения дисциплины (модуля)

Данная дисциплина предполагает изучение библиотеки стандартных модулей и принципов разработки программных систем.

Целью освоения дисциплины "Библиотеки языка программирования Python" является формирование навыков разработки алгоритмов и программных средств, используя библиотеки языка программирования Python.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Библиотеки языка программирования Python входит в обязательную часть учебного плана образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Изучение курса дисциплины "Библиотеки языка программирования Python" базируется на знаниях, полученных при обучении на бакалавриате при изучении основ программирования, алгоритмов и теории сложности, логического программирования, объектно-ориентированного программирования.

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Выполнение и защита выпускной квалификационной работы

Основы теории машинного обучения

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Библиотеки языка программирования Python» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ОПК-2	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач;
ОПК-2.1	Разрабатывает алгоритмы для решения профессиональных задач
ОПК-2.2	Разрабатывает программные средства с использованием современных технологий разработки программного обеспечения, в том числе с применением интеллектуальных технологий
ОПК-6	Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования;
ОПК-6.1	Определяет необходимость в разработке компонент программно-аппаратных комплексов обработки информации и автоматизированного проектирования
ОПК-8	Способен осуществлять эффективное управление разработкой программных средств и проектов.
ОПК-8.1	Оценивает эффективность управления разработкой программных средств и проектов

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 4 зачетных единиц 144 академических часов, в том числе:

- контактная работа – 48,05 академических часов;
- аудиторная – 45 академических часов;
- внеаудиторная – 3,05 академических часов;
- самостоятельная работа – 60,25 академических часов;
- в форме практической подготовки – 0 академических часов;
- подготовка к экзамену – 35,7 академических часов

Форма аттестации - экзамен

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в академических часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Программирование на Python								
1.1 Структуры данных, условный оператор, циклы, функции, исключения и их обработка	1		2		4	Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Устный опрос	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1
1.2 Создание и подключение модулей, объектно-ориентированное программирование. Инкапсуляция, наследование, полиморфизм. Перегрузка операторов. Декораторы				2		4	Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Устный опрос
Итого по разделу			4		8			
2. Основные библиотеки Python								
2.1 Интерактивная оболочка для языка программирования Python. Jupyter	1	1	2		4	1. Подготовка к лабораторному занятию 2. Выполнение лабораторной работы 3. Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Устный опрос	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1
2.2 Структуры данных и инструменты для анализа. Pandas		1	2		4	1. Подготовка к лабораторному занятию 2. Выполнение	1. Беседа - обсуждение 2. Проверка лабораторной	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1

						лабораторной работы 3. Самостоятельное изучение учебной и научной литературы	работы 3. Устный опрос	
2.3 Работа с векторами и матрицами. NumPy и SciPy	1	1	2		4	1. Подготовка к лабораторному занятию 2. Выполнение лабораторной работы 3. Самостоятельное изучение учебной и научной литературы	Проверка лабораторной работы	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1
2.4 Создание двумерных диаграмм и графиков. Matplotlib		1	2		4	1. Подготовка к лабораторному занятию	1. Беседа - обсуждение 2. Устный опрос	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1
Итого по разделу		4	8		16			
3. Библиотеки Python для работы с данными								
3.1 Алгоритмы для машинного обучения и интеллектуального анализа данных. Scikit-learn	1	2	4		8	1. Подготовка к лабораторному занятию	1. Беседа - обсуждение 2. Устный опрос	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1
3.2 Оценка и улучшение математических выражений. Theano		1	2		4	1. Подготовка к лабораторному занятию 2. Выполнение лабораторной работы 3. Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Проверка лабораторной работы 3. Устный опрос	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1
3.3 Фреймворк для обучения нейронных сетей. TensorFlow		1	2		4	1. Подготовка к лабораторному занятию 2. Выполнение лабораторной работы 3. Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Проверка лабораторной работы 3. Устный опрос	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1
Итого по разделу		4	8		16			
4. Библиотеки Python для интеллектуального анализа и обработки естественного языка								
4.1 Создание ботов-пауков. Scrapy	1	2			5	1. Подготовка к лабораторному	1. Беседа - обсуждение	ОПК-2.1, ОПК-2.2,

						занятию 2. Самостоятельное изучение учебной и научной литературы	2. Устный опрос	ОПК-8.1, ОПК-6.1
4.2 Обработка естественного языка. NLTK	1		2		8	1. Подготовка к лабораторному занятию 2. Выполнение лабораторной работы 3. Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Проверка лабораторной работы 3. Устный опрос	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1
4.3 Извлечение данных в интернете. Pattern			2		2	1. Подготовка к лабораторному занятию 2. Выполнение лабораторной работы 3. Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Проверка лабораторной работы 3. Устный опрос	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1
Итого по разделу		2	4		15			
5. Библиотеки Python для визуализации								
5.1 Специфическая визуализация: карты тепла, временные ряды и скрипичные диаграммы. Seaborn	1	2			2	1. Подготовка к лабораторному занятию 2. Выполнение лабораторной работы 3. Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Проверка лабораторной работы	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1
5.2 Интерактивные и масштабируемые графики в браузерах, используя виджеты JavaScript. Bokeh			2		2	1. Подготовка к лабораторному занятию 2. Выполнение лабораторной работы 3. Самостоятельное изучение учебной и научной литературы	1. Беседа - обсуждение 2. Проверка лабораторной работы	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1
5.3 Создание карт. Basemap		3				1. Подготовка к лабораторному занятию 2. Выполнение	1. Беседа - обсуждение 2. Проверка лабораторной	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1

						лабораторной работы 3. Самостоятельное изучение учебной и научной литературы	работы 3. Устный опрос	
5.4 Создание и анализ графов и сетевых структур. NetworkX	1		4		1,25	1. Подготовка к лабораторному занятию 2. Выполнение лабораторной работы 3. Самостоятельное изучение учебной и научной литературы	Коллоквиум	ОПК-2.1, ОПК-2.2, ОПК-8.1, ОПК-6.1
Итого по разделу		5	6		5,25			
Итого за семестр		15	30		60,25		экзамен	
Итого по дисциплине		15	30		60,25		экзамен	

5 Образовательные технологии

1. Традиционные образовательные технологии, ориентированные на организацию образовательного процесса и предполагающую прямую трансляцию знаний от преподавателя к студенту.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.

3. Интерактивные технологии – организация образовательного процесса, которая предполагает активное и нелинейное взаимодействие всех участников, достижение на этой основе лично значимого для них образовательного результата.

Формы учебных занятий с использованием специализированных интерактивных технологий:

Лекция «обратной связи» – лекция–провокация (изложение материала с заранее запланированными ошибками), лекция-беседа, лекция-дискуссия, лекция-конференция.

4. Информационно-коммуникационные образовательные технологии – организация образовательного процесса, основанная на применении программных сред и технических средств работы со знаниями в различных предметных областях.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины

а) Основная литература:

Маккинни, У. Python и анализ данных / У. Маккинни ; перевод с английского А. А. Слинкина. — 2-ое изд., испр. и доп. — Москва : ДМК Пресс, 2020. — 540 с. — ISBN 978-5-97060-590-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131721> (дата обращения: 16.03.2020). — Режим доступа: для авториз. пользователей.

б) Дополнительная литература:

1. Саммерфилд, М. Python на практике : учебное пособие / М. Саммерфилд ; перевод с английского А.А. Слинкин. — Москва : ДМК Пресс, 2014. — 338 с. — ISBN 978-5-97060-095-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/66480> (дата обращения: 16.03.2020). — Режим доступа: для авториз. пользователей.

2. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2019. — 161 с. — (Бакалавр. Прикладной

курс). — ISBN 978-5-534-10971-9. — С. 104 — 127 — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/437489/p.104-127> (дата обращения: 16.03.2020)

в) Методические указания:

1. Торшина, О. А. Объектно-ориентированное программирование : учебное пособие / О. А. Торшина ; МГТУ. - Магнитогорск : МГТУ, 2018. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=3613.pdf&show=dcatalogues/1/1524595/3613.pdf&view=true> (дата обращения: 04.10.2019). - Макрообъект. - Текст : электронный. - ISBN 978-5-9967-1132-1. - Сведения доступны также на CD-ROM.

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
Anaconda Python	свободно распространяемое ПО	бессрочно
Atom Editor	свободно распространяемое ПО	бессрочно
NotePad++	свободно распространяемое ПО	бессрочно
JetBrains PyCharm Community Edition	свободно распространяемое ПО	бессрочно
JetBrains IDEA Community Edition	свободно распространяемое ПО	бессрочно

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка
Электронная база периодических изданий ООО «ИВИС»	https://eivis.ru/
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: https://elibrary.ru/project_risc.asp

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

1. Лекционная аудитория Мультимедийные средства хранения, передачи и представления информации
2. Компьютерный класс. Персональные компьютеры с виртуальной машиной для установки серверного ПО, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.
3. Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки. Все классы УИТ и АСУ с персональными компьютерами, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.
4. Аудиторий для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Ауд. 282 и классы УИТ и АСУ.
5. Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации. Классы УИТ и АСУ.
6. Помещения для хранения и профилактического обслуживания учебного оборудования. Центр информационных технологий – ауд. 372.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

В течение семестра каждый студент выполняет лабораторные занятия.

Лабораторная работа №1.

Задание 1. Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от 0 до 10000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести максимальное значение среди трёхзначных элементов массива, делящихся на 5. Если в исходном массиве нет элемента, значение которого является трёхзначным числом и при этом кратно 5, то вывести сообщение «Не найдено».

Задание 2. Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Опишите

на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 13. В данной задаче под парой подразумевается два подряд идущих элемента массива. Например, для массива из пяти элементов: 6; 2; 13; -26 ; 14 — ответ: 3.

Задание 3. По каналу связи передаётся последовательность положительных целых чисел, все числа не превышают 1000. Количество чисел известно, но может быть очень велико. Затем передаётся контрольное значение последовательности — наибольшее число R , удовлетворяющее следующим условиям:

1) R — произведение двух различных переданных элементов последовательности («различные» означает, что не рассматриваются квадраты переданных чисел; допускаются произведения различных элементов последовательности, равных по величине);

2) R делится на 21.

Если такого числа R нет, то контрольное значение полагается равным 0.

В результате помех при передаче как сами числа, так и контрольное значение могут быть искажены.

Напишите эффективную, в том числе по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет проверять правильность контрольного значения. Программа должна напечатать отчёт по следующей форме:

Вычисленное контрольное значение: ...

Контроль пройден (или — Контроль не пройден)

Перед текстом программы кратко опишите используемый Вами алгоритм решения.

На вход программе в первой строке подаётся количество чисел N . В каждой из последующих N строк записано одно натуральное число, не превышающее 1000. В последней строке записано контрольное значение.

Пример входных данных:

6
70
21
997
7
9
300
21000

Пример выходных данных для приведённого выше примера входных данных:

Вычисленное контрольное значение: 21000

Контроль пройден

Лабораторная работа №2.

Рассмотреть основные библиотеки Python. Создать пример, реализующий основные возможности этих библиотек. Подготовить доклад.

Лабораторная работа №3.

Рассмотреть библиотеки Python для работы с данными. Создать пример, реализующий основные возможности этих библиотек. Подготовить доклад.

Лабораторная работа №4.

Рассмотреть библиотеки Python для интеллектуального анализа и обработки естественного языка. Создать пример, реализующий основные возможности этих библиотек. Подготовить доклад.

Лабораторная работа №5.

Рассмотреть библиотеки Python для визуализации данных. Создать пример, реализующий основные возможности этих библиотек. Подготовить доклад.

Самостоятельная подготовка к коллоквиуму происходит в процессе самоподготовки по каждой теме при изучении курса:

1. Структуры данных, условный оператор, циклы, функции, исключения и их обработка.
2. Создание и подключение модулей.
3. Объектно-ориентированное программирование.
4. Инкапсуляция, наследование, полиморфизм.
5. Перегрузка операторов. Декораторы.
6. Интерактивная оболочка для языка программирования Python. Jupyter.
7. Структуры данных и инструменты для анализа. Pandas.
8. Работа с векторами и матрицами. NumPy и SciPy.
9. Создание двумерных диаграмм и графиков. Matplotlib.
10. Алгоритмы для машинного обучения и интеллектуального анализа данных. Scikit-learn.
11. Оценка и улучшение математических выражений. Theano.
12. Фреймворк для обучения нейронных сетей. TensorFlow.
13. Создание ботов-пауков. Scrapy.
14. Обработка естественного языка. NLTK.
15. Извлечение данных в интернете. Pattern.
16. Специфическая визуализация: карты тепла, временные ряды и скрипичные диаграммы. Seaborn.
17. Интерактивные и масштабируемые графики в браузерах, используя виджеты JavaScript. Bokeh.
18. Создание карт. Basemap.
19. Создание и анализ графов и сетевых структур. NetworkX.

7 Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
ОПК-2: Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач		
ОПК-2.1	Разрабатывает алгоритмы для решения профессиональных задач	<p><i>Перечень теоретических вопросов</i></p> <p>1. Структуры данных, условный оператор, циклы, функции, исключения и их обработка.</p>
ОПК-2.2	Разрабатывает программные средства с использованием современных технологий разработки программного обеспечения, в том числе с применением интеллектуальных технологий	<p>2. Создание и подключение модулей.</p> <p>3. Объектно-ориентированное программирование.</p> <p>4. Инкапсуляция, наследование, полиморфизм.</p> <p>5. Перегрузка операторов. Декораторы.</p> <p>6. Интерактивная оболочка для языка программирования Python.</p> <p>Jupyter.</p> <p>7. Структуры данных и инструменты для анализа. Pandas.</p> <p>8. Работа с векторами и матрицами. NumPy и SciPy.</p> <p>9. Создание двумерных диаграмм и графиков. Matplotlib.</p> <p>10. Алгоритмы для машинного обучения и интеллектуального анализа данных. Scikit-learn.</p> <p>11. Оценка и улучшение математических выражений. Theano.</p> <p>12. Фреймворк для обучения нейронных сетей. TensorFlow.</p> <p>13. Создание ботов-пауков. Scrapy.</p> <p>14. Обработка естественного языка. NLTK.</p> <p>15. Извлечение данных в интернете. Pattern.</p> <p>16. Специфическая визуализация: карты тепла, временные ряды и скрипичные диаграммы. Seaborn.</p> <p>17. Интерактивные и масштабируемые графики в браузерах, используя виджеты JavaScript. Bokeh.</p> <p><i>Практические задания</i></p> <p>Задание 1. Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от 0 до 10000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести максимальное значение среди трёхзначных элементов массива, делящихся на 5. Если в исходном массиве нет элемента, значение которого является трёхзначным числом и при этом кратно 5, то вывести сообщение «Не найдено».</p> <p>Задание 2. Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от -10 000 до 10 000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 13. В данной задаче под парой подразумевается два подряд идущих элемента массива. Например, для</p>

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>массива из пяти элементов: 6; 2; 13; – 26; 14 – ответ: 3.</p> <p><i>Задания на решение задач из профессиональной области, комплексные задания</i></p> <p>1. Рассмотреть основные библиотеки Python. Создать пример, реализующий основные возможности этих библиотек. Подготовить доклад.</p> <p>2. Рассмотреть библиотеки Python для работы с данными. Создать пример, реализующий основные возможности этих библиотек. Подготовить доклад.</p> <p><i>Тесты</i></p> <p>1. Выберите верные утверждения для функций в Python.</p> <p>а) Python не поддерживает вложенные функции б) Функция может возвращать только одно значение в) Функция может принимать неограниченное количество аргументов г) Функция выполняется только при ее вызове, и мы можем повторно использовать ее в программе</p> <p>2. Выберите правильную функцию, чтобы получить случайное целое число в диапазоне от до , кратное .</p> <p>а) random.random(10, 21, 3) б) random.randint(10, 20, 3) в) random.randrange(10, 21, 3)</p> <p>3. Если внутри функции не используется оператор return, функция вернет:</p> <p>а) 0 б) Null в) None г) Значение по умолчанию</p>
<p>ОПК-6: Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования</p>		

Код индикатора	Индикатор достижения компетенции	Оценочные средства
ОПК-6.1	<p>Определяет необходимость в разработке компонент программно-аппаратных комплексов обработки информации и автоматизированного проектирования</p>	<p><i>Перечень теоретических вопросов</i></p> <ol style="list-style-type: none"> 1. Фреймворк для обучения нейронных сетей. TensorFlow. 2. Создание ботов-пауков. Scrapy. 3. Обработка естественного языка. NLTK. 4. Извлечение данных в интернете. Pattern. 5. Специфическая визуализация: карты тепла, временные ряды и скрипичные диаграммы. Seaborn. 6. Интерактивные и масштабируемые графики в браузерах, используя виджеты JavaScript. Bokeh. 7. Создание карт. Basemap. 8. Создание и анализ графов и сетевых структур. NetworkX. <p><i>Практические задания</i></p> <p>По каналу связи передаётся последовательность положительных целых чисел, все числа не превышают 1000. Количество чисел известно, но может быть очень велико. Затем передаётся контрольное значение последовательности – наибольшее число R, удовлетворяющее следующим условиям:</p> <ol style="list-style-type: none"> 1) R – произведение двух различных переданных элементов последовательности («различные» означает, что не рассматриваются квадраты переданных чисел; допускаются произведения различных элементов последовательности, равных по величине); 2) R делится на 21. <p>Если такого числа R нет, то контрольное значение полагается равным 0. В результате помех при передаче как сами числа, так и контрольное значение могут быть искажены. Напишите эффективную, в том числе по используемой памяти, программу, которая будет проверять правильность контрольного значения.</p> <p><i>Задания на решение задач из профессиональной области, комплексные задания</i></p> <p>Рассмотреть библиотеки Python для интеллектуального анализа и обработки естественного языка. Создать пример, реализующий основные возможности этих библиотек. Подготовить доклад.</p> <p><i>Тесты</i></p> <ol style="list-style-type: none"> 1. При выполнении следующего кода на выходе мы получим: <pre>def func(): x = 15 print(x)</pre> $x = 10$ func() <ol style="list-style-type: none"> а) Ошибку б) 15 в) 1510

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>г) 10</p> <p>2. При выполнении следующего кода на выходе мы получим: def func(): global a print(a) a = "hello" print(a) a = "world" func() print(a) а) hello world world б) world hello hello в) world hello world г) hello hello world</p> <p>3. Выберите правильное объявление func(), чтобы можно было успешно выполнить следующие вызовы функции: а) func(15, 60, 50) б) func(1, 2) в) def func(*data) г) def func(**kwargs) д) def func(args*) е) Это сделать невозможно</p>
ОПК-8: Способен осуществлять эффективное управление разработкой программных средств и проектов		
ОПК-8.1	Оценивает эффективность управления разработкой программных средств и проектов	<p><i>Задания на решение задач из профессиональной области, комплексные задания</i> Рассмотреть библиотеки Python для визуализации данных. Создать пример, реализующий основные возможности этих библиотек. Подготовить доклад.</p> <p><i>Тесты</i> 1. При выполнении следующего кода на выходе мы получим: num = 1 def func(): global num num = num + 3 print(num) func() print(num) а) 1 1 б) 4 1 в) 1 4 г) 4 4 д) runtime error</p> <p>2. Какой из следующих заголовков функций</p>

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>правильный?</p> <p>а) <code>def fun(a = 2, b, c = 3)</code> б) <code>def fun(a, b = 2, c = 3)</code> верно в) <code>def fun(a = 2, b = 3, c)</code> г) <code>def fun(a, b, c = 3, d)</code></p> <p>3. Какие из следующих записей можно использовать для вызова <code>func()</code> с указанным ниже заголовком? Выберите правильные варианты ответа <code>def func(a, b, c, d)</code>:</p> <p>а) <code>func (a = 1, 2, 3, 4)</code> б) <code>func (a = 1, b = 2, c = 3, 4)</code> в) <code>func (a = 1, b = 2, c = 3, d = 4)</code> г) <code>func (1, 2, 3, d = 4)</code> д) <code>func (1, 2, 3, 4)</code></p> <p>4. Каков правильный порядок использования <code>* args</code>, <code>** kwargs</code> и формальных аргументов (<code>fargs</code>) в заголовке функции?</p> <p>а) <code>some_func(** kwargs, * args, fargs)</code> б) <code>some_func(fargs, * args, ** kwargs)</code> верно в) порядок не имеет значения г) <code>some_func(* args, ** kwargs, fargs)</code></p>

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине «Библиотеки языка программирования Python» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме экзамена.

Экзамен по дисциплине проводится в устной форме по экзаменационным билетам, каждый из которых включает два теоретических вопроса и одно практическое задание.

Показатели и критерии оценивания экзамена:

– на оценку **«отлично»** (5 баллов) – обучающийся демонстрирует высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в ситуациях повышенной сложности.

– на оценку **«хорошо»** (4 балла) – обучающийся демонстрирует средний уровень сформированности компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

– на оценку **«удовлетворительно»** (3 балла) – обучающийся демонстрирует пороговый уровень сформированности компетенций: в ходе контрольных мероприятий допускаются ошибки, проявляется отсутствие отдельных знаний, умений, навыков, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

– на оценку **«неудовлетворительно»** (2 балла) – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

– на оценку **«неудовлетворительно»** (1 балл) – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.