



**Куйбышевский филиал федерального государственного бюджетного  
образовательного учреждения высшего образования «Новосибирский  
государственный педагогический университет»**

**УТВЕРЖДАЮ**

Декан  
Факультет психолого-педагогического  
образования

В.А.Кобелев

(подпись)

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
Программирование**

Направление подготовки:

**44.03.05 Педагогическое образование (с двумя профилями подготовки)**

Направленность (профиль):

**Математика и Информатика**

Уровень высшего образования:

**бакалавриат**

Форма обучения:

**очная**

Куйбышев 2019

## **СОСТАВИТЕЛИ:**

Кандидат педагогических наук, доцент кафедры математики, информатики и методики преподавания В.В.Моторин

## **РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УЧЕБНОМ ПРОЦЕССЕ**

на заседании кафедры математики, информатики и методики преподавания (КФ) (протокол №9 от 15.05.2019 г.)

Заведующий кафедрой математики, информатики и методики преподавания (КФ) И. А. Дудковская

# 1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## 1.1 Цель освоения дисциплины:

- изучение методов программирования для овладения знаниями в области технологии программирования;
- подготовка к осознанному использованию, как языков программирования, так и методов программирования;
- обучение разработке алгоритмов на основе структурного подхода;
- закрепление навыков алгоритмизации и программирования на основе изучения языка программирования Python;
- знакомство с основными структурами данных и типовыми методами обработки этих структур.

## 1.2 Место дисциплины в структуре образовательной программы

Программа дисциплины разработана в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки), утвержденным приказом Минобрнауки России от 22.02.2018 г. №125, профессиональными стандартами: педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель), утвержденным приказом Министерства труда и социальной защиты Российской Федерации от 18.10.2013 г. №544н, педагог профессионального обучения, профессионального образования и дополнительного профессионального образования, утвержденным приказом Министерства труда и социальной защиты Российской Федерации от 08.09.2015 г. №608н, педагог дополнительного образования детей и взрослых, утвержденным приказом Министерства труда и социальной защиты Российской Федерации от 05.05.2018 г. №298н.

Дисциплина относится к обязательной части блока 1 «Дисциплины» учебного плана образовательной программы, изучается в 5, 6, 7, 8 семестрах. Трудоемкость дисциплины: 9 ЗЕ / 324 часа, в том числе 120 часов - контактная работа с преподавателем, 136 часа - самостоятельная работа (таблица 2).

## 1.3 Планируемые результаты обучения по дисциплине

Дисциплина направлена на формирование компетенции(-ий), представленных в таблице 1.

Таблица 1

### Планируемые результаты обучения по дисциплине

Код и наименование компетенции	
Индикаторы достижения компетенции	Планируемые результаты обучения по дисциплине
<b>ОПК-2 Способен участвовать в разработке основных и дополнительных образовательных программ, разрабатывать отдельные их компоненты (в том числе с использованием информационно-коммуникационных технологий)</b>	
ОПК-2.1 Знает: историю, теорию, закономерности и принципы построения и функционирования образовательных систем, роль и место образования в жизни личности и общества; основы дидактики, основные принципы деятельностного подхода, виды и приемы современных образовательных технологий; пути достижения образовательных результатов в области ИКТ.	Знать: основные принципы обучения и программированию с использованием современных образовательных технологий. Уметь: выбирать образовательные
ОПК-2.2 Умеет: классифицировать образовательные системы и образовательные технологии; разрабатывать и применять отдельные компоненты основных и дополнительных образовательных программ в реальной и виртуальной образовательной среде.	технологии для обучения и программированию, разрабатывать программы элективных курсов по программированию.
ОПК-2.3 Владеет: приемами разработки и реализации программ учебных дисциплин в рамках основной общеобразовательной программы; средствами формирования	Владеть: приемами разработки и реализации программ по

<p>умений, связанных с информационно-коммуникационными технологиями (далее – ИКТ); действиями реализации ИКТ, для разработки программ и технологий: на уровне пользователя, на общепедагогическом уровне; на уровне преподаваемого (ых) предметов (отражающая профессиональную ИКТ-компетентность соответствующей области человеческой деятельности).</p>	<p>программированию; средствами ИКТ, для разработки программ и других материалов по программированию.</p>
<p><b>ПК-3 способен применять предметные знания при реализации образовательного процесса</b></p>	
<p>ПК-3.1 Знает: закономерности, принципы и уровни формирования и реализации содержания образования, структуру, состав и дидактические единицы содержания предмета.</p>	<p>Знать: методы структурного программирования, как наиболее распространённого и</p>
<p>ПК-3.2 Умеет: осуществлять отбор учебного содержания для реализации в различных формах обучения в соответствии с дидактическими целями и возрастными особенностями учащихся.</p>	<p>эффективного метода разработки программных продуктов. Уметь:</p>
<p>ПК-3.3 Владеет: предметным содержанием образования по предмету; умениями отбора вариативного содержания с учетом взаимосвязи урочной и внеурочной формы обучения.</p>	<p>основе структурного подхода. Владеть: навыками алгоритмизации и программирования на основе изучения языка программирования Python.</p>

## **2 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ**

### **Пятый семестр**

#### **Тема 1. Базовые понятия языка Python**

Язык Python в качестве языка учебного назначения. Дидактические характеристики языка Python. Имена (идентификаторы). Правила задания имён. Операция присваивания. Операция ввода `input()`. Операция вывода на экран. Организация комментариев. Однострочные и многострочные комментарии. Встроенная справочная система.

#### **Тема 2. Типы данных, конвертация типов**

Числовой тип данных. Строковый тип данных. Логический тип данных. Конвертация типов данных.

#### **Тема 3. Математические и логические операции**

Библиотечный модуль `math`. Подключение модуля `math` в программе. Основные функции модуля `math`. Логические операции: `and`, `or`, `not`. Использование логических операций в условиях.

### **Шестой семестр**

#### **Тема 1. Алгоритмические структуры: следование и ветвление**

Линейная последовательность операторов. Изменение хода выполнения программы при помощи операции ветвления. Различные способы организации ветвления алгоритма при помощи оператора `if`. Условие в операторе `if`. Реализация операции выбора (множественное ветвление) при помощи оператора `if`.

#### **Тема 2. Алгоритмические структуры: циклы**

Цикл с условием. Управление ходом выполнения цикла `while`. Цикл с параметром. Использование цикла `for` для операций над элементами последовательности. Применение функции `range()` для создания арифметических последовательностей. Изменение хода выполнения циклов при помощи операторов `break` и `continue`. Ветвь `else` в циклах.

#### **Тема 3. Коллекции: строка, кортеж, список**

Классификация коллекций. Понятие последовательности. Индексированные коллекции. Коллекции языка Python: строка, кортеж, список. Определение, свойства и операции над строками. Определение, свойства и операции над кортежами. Определение, свойства и операции над списками.

#### **Тема 4. Коллекции: множество, словарь**

Неиндексированные коллекции. Множество: операция проверки на принадлежность, операции сравнения, математические операции. Словари. Ключи и значения словаря. Создание словаря. Операции над словарями. Цикл по элементам словаря.

#### **Тема 5. Методы коллекций**

Основные методы коллекций. Методы списков, их использование. Методы строк: форматирование, поиск, модификация, проверка содержимого, объединение и разделение. Использование методов коллекций для решения практических задач.

### **Седьмой семестр**

#### **Тема 1. Формирование массивов**

Создание, обработка и получение результатов операций над матрицами с использованием вложенных списков. Библиотека `NumPy`. Анатомия массивов библиотеки `NumPy`. Создание массивов на основе кортежей, списков, арифметических последовательностей. Заполнение массива одинаковыми элементами. Формирование массива при помощи генератора случайных чисел.

#### **Тема 2. Методы работы с векторами и матрицами**

Стандартные матрицы, нулевая, единичная, с указанием элементов главной диагонали, заполненная мусором из оперативной памяти. Пересоздание массивов методом изменения размеров исходного массива. Использование метода `reshape` совместно с другими методами. Вывод информации о массиве.

#### **Тема 3. Решение задач линейной алгебры**

Операции над матрицами. Поэлементные операции. Операция сравнения. Проверка на принадлежность. Сумма элементов. Максимальный и минимальный элемент. Библиотечный модуль `linalg`. Возведение матрицы в степень. Вычисление значения многочлена. Вычисление определителя матрицы, вычисление системы линейных уравнений. Скалярное произведение двух векторов. Матричное произведение двух массивов.

## Восьмой семестр

### Тема 1. Описание и вызов функций

Функция - структурный элемент программы. Описание функции. Параметры и аргументы функции. Тело функции. Вызов функции. Область действия имён при использовании функций.

### Тема 2. Передача параметров и возврат результатов

Использование return для возврата результата работы функции в основную программу. Значения по умолчанию. Локальные и глобальные объекты при использовании функций в программе.

### Тема 3. Область действия имён

Правила использования имён для локальных и глобальных параметров. Локальные объекты функции. Глобальные объекты основной программы.

### Тема 4. Функция open() для работы с текстовыми файлами

Открытие файлов. Файловые переменные. Режимы использования файла: чтение, запись, добавление. Обработка файла. Методы работы с текстовыми файлами. Операции чтения/записи. Указатель файла. Операции с указателем. Закрытие файла, операция close(). Автозакрытие файла при помощи оператора with, стандартные примеры операций с текстовыми файлами.

### Тема 5. Методы модуля pickle для работы с файлами

Возможности модуля pickle при работе с текстовыми файлами. Операция консервации данных. Операция расконсервации данных. Бинарный формат записи файла. Режимы бинарных операций записи/чтения файлов при помощи модуля pickle.

## Содержание работ по дисциплине

Таблица 2

Содержание работы	Виды и формы работы, час				Всего, час	Код компетенции	
	Лекции	Лабораторные	Практические	Консультации			
<b>Пятый семестр</b>							
Тема 1. Базовые понятия языка Python	2		4		2	8	ОПК-2
Тема 2. Типы данных, конвертация типов	2		6		4	12	ОПК-2
Тема 3. Математические и логические операции	2		8		6	16	ОПК-2
Подготовка к зачету							ОПК-2
<b>Шестой семестр</b>							
Тема 1. Алгоритмические структуры: следование и ветвление	2		4		4	10	ОПК-2, ПК-3
Тема 2. Алгоритмические структуры: циклы	2		8		4	14	ОПК-2, ПК-3
Тема 3. Коллекции: строка, кортеж, список	4		8		6	18	ОПК-2, ПК-3
Тема 4. Коллекции: множество, словарь	4		8		8	20	ОПК-2, ПК-3
Тема 5. Методы коллекций	2		4		4	10	ОПК-2
Подготовка к экзамену				2	34	36	ОПК-2, ПК-3
<b>Седьмой семестр</b>							
Тема 1. Формирование массивов	1		2		12	15	ПК-3
Тема 2. Методы работы с векторами и матрицами	2		6		24	32	ПК-3
Тема 3. Решение задач линейной алгебры	1		4		20	25	ПК-3
Подготовка к зачету							ПК-3
<b>Восьмой семестр</b>							

Тема 1. Описание и вызов функций	2		4		8	14	ОПК-2, ПК-3
Тема 2. Передача параметров и возврат результатов	2		4		8	14	ОПК-2
Тема 3. Область действия имён	2		4		8	14	ОПК-2
Тема 4. Функция open() для работы с текстовыми файлами	2		4		10	16	ПК-3
Тема 5. Методы модуля pickle для работы с файлами	2		4		8	14	ПК-3
Подготовка к экзамену				2	34	36	ОПК-2, ПК-3
Итого по дисциплине	34		82	4	204	324	

### **3 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Для успешного освоения дисциплины следует ознакомиться с содержанием разделов и тем по дисциплине (см. п. 2), следовать технологической карте при выполнении самостоятельной работы (табл. 3), использовать рекомендованные ресурсы (п. 4) и выполнять требования внутренних стандартов университета.



## 4 УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 4.1 Основная учебная литература

1. Гниденко, И. Г. Технологии и методы программирования: учеб. пособие для прикладного бакалавриата / И.Г.Гниденко, Ф.Ф.Павлов, Д.Ю. Федоров. – М.: Издательство Юрайт, 2019. – 235с. – (Серия : Бакалавр. Прикладной курс). ISBN 978-5-534-02816-41.
2. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учеб.пособие для прикладного бакалавриата / Д.Ю. Федоров. – 2-е изд., перераб. и доп. – М.: Издательство Юрайт, 2019. – 161 с. – (Серия : Бакалавр. Прикладной курс). ISBN 978-5-534-10971-9

### 4.2 Дополнительная учебная литература

1. Златопольский, Дмитрий Михайлович. Основы программирования на языке Python / Д. М. Златопольский. - Москва : ДМК Пресс, 2017. - 284 с. : ил. - ISBN 978-5-97060-552-3 : 545-00.
2. Ночка, Евгений Иванович. Основы алгоритмизации и программирования на языке Питон : учебник / Е. И. Ночка. - Москва : КУРС, 2017. - 208 с. - (Среднее профессиональное образование). - ISBN 978-5-906818-75-1 : 505-37.
3. Подсадников, Алексей Владимирович Основы программирования на языке Python 3 : учебное пособие / А. В. Подсадников, К. В. Розов, А. В. Русецкий ; Новосибирский государственный педагогический университет. - Новосибирск : НГПУ, 2019. - 221 с. : ил. - Библиогр.: с. 220. - URL: <https://lib.nspu.ru/views/library/85491/read.php> (дата обращения: 23.11.2019) . - Доступна эл. версия в ЭБС НГПУ. - ISBN 978-5-00104-478-9. - Текст : электронный
4. Серебряков В.А. и др. Теория и реализация языков программирования. Изд. 2-е, доп. и испр :Учебное пособие./ В.А. Серебряков, М.П. Галочкин, Д.Р. Гончар, М.Г. Фуругян.- М.: МЗ Пресс, 2006. – 352 с.: ил. ISBN 94073-094-9
5. Теория и реализация языков программирования [Электронный ресурс] : учебное пособие / В. А. Серебряков, М. П. Галочкин, Д. Р. Гончар, М. Г. Фуругян. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 372 с. - URL: <http://www.iprbookshop.ru/73731.html> (дата обращения: 31.10.2019) . - Доступна эл. версия. ЭБС "IPRBooks"

### 4.3 Ресурсы открытого доступа

1. ЭБС «IPRbooks».. URL: <http://www.iprbookshop.ru>.
2. Открытый видеохостинг YouTube.. URL: <https://www.youtube.com/>
3. Персональные сайты преподавателей университета [Электронный ресурс]. URL: <http://prepod.nspu.ru/>

### 4.4 Технологическая карта самостоятельной работы студента

Таблица 3

Темы дисциплины	Перечень учебно-методического обеспечения (номер источника из п.п. 4.1-4.3)
Задания для самостоятельной работы	
<b>Пятый семестр</b>	
<b>Тема 1. Базовые понятия языка Python</b>	Основная учебная литература: 1 Дополнительная учебная литература: 2 Ресурсы открытого доступа: 2

## Задание 01

### Линейные алгоритмы

- ⊕ Составление программы с использованием линейного алгоритма.
- ◇ Даны координаты трёх вершин треугольника:  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ . Найти его периметр и площадь, используя формулу для расстояния между двумя точками на плоскости. Для нахождения площади треугольника использовать формулу Герона.

```
print('Координаты первой вершины треугольника')
x1 = float(input('x1: ')); y1 = float(input('y1: '))
print('Координаты второй вершины треугольника')
x2 = float(input('x2: ')); y2 = float(input('y2: '))
print('Координаты третьей вершины треугольника')
x3 = float(input('x3: ')); y3 = float(input('y3: '))
a=((x1-x2)**2+(y1-y2)**2)**(1/2) # Сторона a
b=((x2-x3)**2+(y2-y3)**2)**(1/2) # Сторона b
c=((x1-x3)**2+(y1-y3)**2)**(1/2) # Сторона c
P=a+b+c; p=P/2 # Периметр и полупериметр
S=(p*(p-a)*(p-b)*(p-c))**(1/2) # Площадь по формуле Герона
print('Периметр треугольника: ', P)
print('Площадь треугольника: ', S)
```

- ☑ Координаты первой вершины треугольника  
x1: 0  
y1: 0  
Координаты второй вершины треугольника  
x2: 0  
y2: 5  
Координаты третьей вершины треугольника  
x3: 5  
y3: 0  
Периметр треугольника: 17.071067811865476  
Площадь треугольника: 12.5

Тема 2. Типы данных, конвертация типов

Основная учебная литература: 2  
Дополнительная учебная литература: 3  
Ресурсы открытого доступа: 1

## Задание 02

### Целый тип данных

- ⊕ Составление программы, с использованием данных только целого типа. Операцию ветвления не использовать.
- ◇ Дан номер некоторого года (целое положительное число). Определить соответствующий ему номер столетия, учитывая, что, к примеру, началом 20 столетия был 1901 год.
- ☞ `ГОД = int(input('Год: '));`  
`СТОЛЕТИЕ=(ГОД-1)//100 #Сдвигаем на один год назад для удобства счёта`  
`print(СТОЛЕТИЕ+1, 'столетие')`
- ☑ Год: 1901  
20 столетие

Тема 3. Математические и логические операции

Основная учебная литература: 1  
Дополнительная учебная литература: 5  
Ресурсы открытого доступа: 3

## Задание 03

### Логический тип данных

- ⊕ Составление программы с использованием логического типа `bool`. Результатом работы программы должно быть одно из значений: `True` или `False`. Операцию ветвления не использовать.
- ◇ Даны координаты двух различных полей шахматной доски  $x_1, y_1, x_2, y_2$  - целые числа, лежащие в диапазоне `[1; 8]`. Проверить истинность высказывания: «Слон за один ход может перейти с одного поля на другое».
- ☞ `print('Может ли слон за один ход перейти с одного поля на другое?')`  
`print('True — может, False — нет.')`  
`print('Первое поле шахматной доски из диапазона [1;8]')`  
`x1 = int(input('x1: ')); y1 = int(input('y1: '))`  
`print('Второе поле шахматной доски из диапазона [1;8]')`  
`x2 = int(input('x2: ')); y2 = int(input('y2: '))`  
`ОТВЕТ=bool(abs(x1-x2)==abs(y1-y2))`  
`print('Ответ:', ОТВЕТ)`
- ☑ Может ли слон за один ход перейти с одного поля на другое?  
True — может, False — нет.  
Первое поле шахматной доски из диапазона `[1;8]`  
`x1: 2`  
`y1: 3`  
Второе поле шахматной доски из диапазона `[1;8]`  
`x2: 4`  
`y2: 5`  
Ответ: `True`

Подготовка к зачету

Основная учебная литература: 1, 2

	Дополнительная учебная литература: 2, 3, 5 Ресурсы открытого доступа: 1, 2, 3
<b>Шестой семестр</b>	
<b>Тема 1. Алгоритмические структуры: следование и ветвление</b>	Основная учебная литература: 2 Дополнительная учебная литература: 3 Ресурсы открытого доступа: 2
<b>Задание 04</b>	
<b>Ветвления</b>	
<ul style="list-style-type: none"> <li>⊕ Составление программы, с использованием операции ветвления.</li> <li>◇ Для данного вещественного <math>x</math> найти значение следующей функции <math>f</math>, принимающей вещественные значения:  <math display="block">f(x) = \begin{cases} 2 \cdot \sin(x), &amp; \text{если } x &gt; 0 \\ 6 - x, &amp; \text{если } x \leq 0 \end{cases}</math> </li> <li>☐ <pre>from math import * # Подключение модуля math для поддержки функции sin X = float(input('Вещественное значение x: ')); if X&gt;0: F=2*sin(X) else: F=6-X print('x =', X, ' f(x) =', F)</pre> </li> <li>☑ Вещественное значение <math>x</math>: 12.25  <math>x = 12.25 \quad f(x) = -0.6222387099622546</math> </li> </ul>	
<b>Тема 2. Алгоритмические структуры: циклы</b>	Основная учебная литература: 1, 2 Дополнительная учебная литература: 2, 4 Ресурсы открытого доступа: 1, 3

## Задание 05

### Цикл while

- ⊕ Составление программы, с использованием цикла while.
- ◇ Даны положительные числа  $A$ ,  $B$ ,  $C$ . На прямоугольнике размера  $A \times B$  размещено максимально возможное количество квадратов со стороной  $C$  (без наложений). Найти количество квадратов, размещённых на прямоугольнике. Операции умножения и деления не использовать.
- ▢ 

```
print('Стороны прямоугольника')
A=float(input('A: '))
B=float(input('B: '))
print('Сторона квадрата')
C=float(input('C: '))

СЧЁТ=0 # Счётчик количества квадратов
сА=0 # Размещение квадратов на прямоугольнике вдоль стороны A
сВ=0 # Размещение квадратов на прямоугольнике вдоль стороны B
while A-сА>=C: # Пока есть место для очередного квадрата на стороне A
    сА=сА+C # Размещаем квадрат вдоль стороны A
    while B-сВ>=C: # Пока есть место для очередного квадрата на стороне B
        сВ=сВ+C # Размещаем квадрат вдоль стороны B
        СЧЁТ=СЧЁТ+1 # Считаем количество квадратов
        сВ=0 # Переходим к следующему слою квадратов
print('Максимальное количество квадратов \
на прямоугольнике:', СЧЁТ)
```
- ☑ Стороны прямоугольника  
A: 75  
B: 90  
Сторона квадрата  
C: 12  
Максимальное количество квадратов на прямоугольнике: 42

## Задание 06

### Цикл for

⊕ Составление программы, с использованием цикла for.

◇ Дано вещественное число  $X$  и целое число  $N (> 0)$ . Найти значение выражения:

$$1 - X^2/(2!) + X^4/(4!) - \dots + (-1)^N \cdot X^{2N}/((2 \cdot N)!)$$

Полученное число является приближенным значением функции  $\cos$  в точке  $X$ .

```
from math import* # Модуль math для вычисления факториала
N=int(input('Целое число N (N>0): '))
X=float(input('Вещественное число X: '))
S=ЗНАК=1 # Сумма и знако чередование изначально равны 1
for i in range(2, 2*N, 2): # Цикл от 2 до 2N с шагом 2
    ЗНАК=-ЗНАК # Знако чередование
    S=S+ЗНАК*(pow(X,i)/factorial(i))
print('Сумма выражения:', S)
print('cos(X)=', cos(X))
```

☑ Целое число  $N (N>0)$ : 10  
Вещественное число  $X$ : 5.325798  
Сумма выражения: 0.575527870778507  
 $\cos(X) = 0.5756583220969361$

Тема 3. Коллекции: строка, кортеж, список

Основная учебная литература: 1  
Дополнительная учебная литература: 4, 5  
Ресурсы открытого доступа: 1, 2

## Задание 07

### Максимумы и минимумы

- ⊕ Нахождение максимальных и минимальных элементов и величин.
- ◇ Дано целое число  $N$  и набор из  $N$  нулей и единиц. Найти количество элементов самой длинной последовательность одинаковых цифр, т.е. нулей или единиц.
- ☞ `import random # Генератор случайных чисел`  
`N=int(input('Целое число N (N>0): '))`  
`A=[]`  
`for i in range(N):`  
`A.append(random.randint(0,1))`  
`print(A)`  
`TMP=0 # Временная переменная для подсчёта длин групп`  
`СЧЁТ=[] # Список длин групп`  
`for i in range(len(A)-1):`  
`TMP=TMP+1 # Подсчёт кол-ва элементов в текущей группе`  
`if A[i]!=A[i+1]: # Если два соседних элемента не равны, то`  
`СЧЁТ.append(TMP) # добавляем длину группы в список СЧЁТ,`  
`TMP=0 # обнуляем счётчик для новой группы,`  
`IND=i+1 # фиксируем индекс места «стыка» групп.`  
`СЧЁТ.append(len(A)-IND) # Количество элементов в последней группе`  
`print(max(СЧЁТ)) # Самая длинная последовательность цифр`
- ☑ Целое число  $N$  ( $N>0$ ): 30  
[1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1]  
7

## Задание 12

### Строки

- ⊕ Составление программы обработки строк.
- ◇ Дана строка, состоящая из русских слов, набранных заглавными буквами. Преобразовать каждое слово в строке, заменив в нем все последующие вхождения его первой буквы на символ «.» (точка). Например, слово «МИНИМУМ» надо преобразовать в «МИНИ.У.»
- 📄 

```
print('Строка текста на русском языке заглавными буквами:')
Строка=input()
S=Строка.split() # S - список слов введённой строки
New=[] # Новый список слов
for слово in S:
    замена=слово[1:].replace(слово[0],'.') # Требуемая задачей замена
    New.append(слово[0]+замена) # Формирование нового списка слов
Строка=' '.join(New) # Формирование новой строки из списка
print('Изменённая строка:', Строка)
```
- ☑ Строка текста на русском языке заглавными буквами:  
АБРАКАДАБРА МИНИМУМА И МАКСИМУМА  
Изменённая строка: АБР.К.Д.БР. МИНИ.У.А И МАКСИ.У.А

Тема 4. Коллекции: множество, словарь

Основная учебная литература: 1  
Дополнительная учебная литература: 2  
Ресурсы открытого доступа: 1



## Задание 11

### Множества

⊕ Оптимизация алгоритма решения задачи за счёт использования множеств.

⊖ На трёх участках возделывают сельскохозяйственные культуры. Известны виды культур, выращиваемых на каждом из участков. Определить виды тех культур, которые: возделывают на каждом из участков; возделывают хотя бы на одном из участков; не возделывают ни на одном участке. Культуры растений: горох, капуста, картофель, морковь, редис, салат, томат, укроп.

☑ `Культура=('горох', 'капуста', 'картофель', 'морковь', 'редис', 'салат', 'томат', 'укроп')`

`# Номера культур на каждом участке (множества):`

`Участок1={1,2,3}`

`Участок2={0,1,2,3,4}`

`Участок3={1,2,3,6}`

`На_каждом = Участок1 & Участок2 & Участок3`

`Хотя_бы_на_одном = Участок1 | Участок2 | Участок3`

`Ни_на_одном = set(range(len(Культура))) - Хотя_бы_на_одном`

`print()`

`print('На каждом:', end=' ')`

`for i in На_каждом: print(Культура[i], end=' ')`

`print()`

`print('Хотя бы на одном:', end=' ')`

`for i in Хотя_бы_на_одном: print(Культура[i], end=' ')`

`print()`

`print('Ни на одном:', end=' ')`

`for i in Ни_на_одном: print(Культура[i], end=' ')`

☑ На каждом: капуста картофель морковь

Хотя бы на одном: горох капуста картофель морковь редис томат

Ни на одном: салат укроп

#### Подготовка к экзамену

Основная учебная литература: 1, 2

Дополнительная учебная литература: 2, 3, 4, 5

Ресурсы открытого доступа: 1, 2, 3

#### Седьмой семестр

#### Тема 1. Формирование массивов

Основная учебная литература: 2

Дополнительная учебная литература: 3

Ресурсы открытого доступа: 2

## Задание 09

### Одномерные массивы

⊕ Составление программы обработки элементов одномерного массива.

◇ Дано число R и массив размера N. Найти два соседних элемента массива, сумма которых наиболее близка к числу R. Вывести эти элементы в порядке возрастания их индексов.

📄 import random

```
N=int(input("Размер массива: "))
```

```
R=int(input("Число R: "))
```

```
A=[]
```

```
for i in range(N):
```

```
    A.append(random.random()*(10))
```

```
    print("%6.2f % A[i], end=' ')
```

```
РАЗНИЦЫ=[] # Список разниц между суммами соседних элементов и R
```

```
for i in range(N-1): РАЗНИЦЫ.append(abs(A[i]+A[i+1]-R))
```

```
ind=РАЗНИЦЫ.index(min(РАЗНИЦЫ)) # Индекс наименьшей из разниц
```

```
print()
```

```
print('Два соседних элемента,
```

```
    сумма которых наиболее близка к числу', R, ':',
```

```
    '%5.2f % A[ind], %5.2f % A[ind+1])
```

☑ Размер массива: 10

Число R: 5

0.62 9.31 6.69 6.45 1.21 3.47 0.52 2.99 4.66 8.12

Два соседних элемента, сумма которых наиболее близка к числу 5:

1.21 3.47

**Тема 2. Методы работы с векторами и матрицами**

Основная учебная литература: 1  
Дополнительная учебная литература: 5  
Ресурсы открытого доступа: 2

## Задание 10

### Двумерные массивы

⊕ Составление программы обработки элементов двумерного массива.

◇ Дана матрица размера  $M \times N$ . Найти минимальный среди элементов тех строк, которые упорядочены либо по возрастанию, либо по убыванию. Если упорядоченные строки в матрице отсутствуют, то вывести 0.

```
import random
print('Размерность матрицы M x N')
m=int(input('M: '))
n=int(input('N: '))

# Формирование матрицы:
A=[]
for i in range(m):
    tmp=[]
    for j in range(n):
        tmp.append(random.randrange(100))
        print('%4i' % tmp[j], end=' ')
    print()
    A.append(tmp)

# Дублирование матрицы A в B для организации алгоритма решения
B=[]
for i in range(len(A)):
    tmp=[]
    for j in range(len(A[1])): tmp.append(A[i][j])
    B.append(tmp)

MIN=[] # Список для сбора минимальных элементов
OK=False # OK – проверка наличия возрастающих или убывающих строк
for i in range(len(A)):
    B[i].sort() # Сортировка строк в матрице B по возрастанию
    if A[i]==B[i]: # Возрастающая строка из B равна строке из A
        OK=True # Есть возрастающие строки
        MIN.append(min(A[i])) # Добавляем минимальный в MIN
    B[i].reverse() # Сортировка строк в матрице B по убыванию
    if A[i]==B[i]: # Убывающая строка из B равна строке из A
        OK=True # Есть убывающие строки
        MIN.append(min(A[i])) # Добавляем минимальный в MIN

if OK:
    print('Минимальный элемент из элементов строк,')
    print('упорядоченных по возрастанию или убыванию:', min(MIN))
```

Подготовка к зачету

Основная учебная литература: 1, 2  
Дополнительная учебная литература: 3, 5  
Ресурсы открытого доступа: 2

Восьмой семестр

**Задание 08****Функции**

✚ Описание функции с использованием её в ходе дальнейшего выполнения программы.

◇ Описать функцию  $LENG(x_A, y_A, x_B, y_B)$  вещественного типа, находящую длину отрезка  $AB$  на плоскости по координатам его концов:

$$|AB| = ((x_A - x_B)^2 + (y_A - y_B)^2)^{1/2}$$

( $x_A, y_A, x_B, y_B$  — вещественные параметры). С помощью этой функции найти длины отрезков  $AB, AC, AD$ , если даны координаты точек  $A, B, C, D$ .

```

print('Координаты точки A')
Ax=float(input('x: '))
Ay=float(input('y: '))
print('Координаты точки B')
Bx=float(input('x: '))
By=float(input('y: '))
print('Координаты точки C')
Cx=float(input('x: '))
Cy=float(input('y: '))
print('Координаты точки D')
Dx=float(input('x: '))
Dy=float(input('y: '))

def LENG(Xa,Ya,Xb,Yb): # Описание функции
    return pow((pow(Xa-Xb,2)+pow(Ya-Yb,2)), 1/2)

# Вызов функции:
print('Длина отрезка AB =', LENG(Ax,Ay,Bx,By))
print('Длина отрезка AC =', LENG(Ax,Ay,Cx,Cy))
print('Длина отрезка AD =', LENG(Ax,Ay,Dx,Dy))

```

☑ Координаты точки A  
x: 23  
y: 56  
Координаты точки B  
x: 12  
y: 65  
Координаты точки C  
x: 20  
y: 62  
Координаты точки D  
x: 81  
y: 33

**Задание 13****Текстовые файлы**

⊕ Создание и обработка текстовых файлов.

◇ Дан текстовый файл example.txt на русском языке. Подсчитать в нём число появлений гласных строчных (маленьких) букв. Создать файл результата result.txt, строки которого содержат информацию вида: «буква --> число появлений» (например, «а --> 25»). Буквы, отсутствующие в тексте, в файл не включать.

```

with open('/home/user/code/example.txt', 'r') as f:
    Текст=f.read()

print('Текст на русском языке (файл example.txt):')
print('-----')
print(Текст)

a=e=ë=и=о=y=ы=э=ю=я=0 # Изначально обнуляем кол-во гласных букв
for i in Текст:
    if i=='a': a=a+1
    if i=='e': e=e+1
    if i=='ë': ë=ë+1
    if i=='и': и=и+1
    if i=='o': o=o+1
    if i=='y': y=y+1
    if i=='ы': ы=ы+1
    if i=='э': э=э+1
    if i=='ю': ю=ю+1
    if i=='я': я=я+1

with open('/home/user/code/result.txt', 'w') as f:
    if a!=0: f.write('a --> '+str(a)+'\n') # \n - перевод на новую строку
    if e!=0: f.write('e --> '+str(e)+'\n')
    if ë!=0: f.write('ë --> '+str(ë)+'\n')
    if и!=0: f.write('и --> '+str(и)+'\n')
    if o!=0: f.write('o --> '+str(o)+'\n')
    if y!=0: f.write('y --> '+str(y)+'\n')
    if ы!=0: f.write('ы --> '+str(ы)+'\n')
    if э!=0: f.write('э --> '+str(э)+'\n')
    if ю!=0: f.write('ю --> '+str(ю)+'\n')
    if я!=0: f.write('я --> '+str(я)+'\n')

print('Кол-во появлений гласных строчных букв (файл result.txt):')
with open('/home/user/code/result.txt', 'r') as f:
    print(f.read())

```

**Тема 5. Методы модуля pickle для работы с файлами**

Основная учебная литература: 2  
 Дополнительная учебная литература: 2  
 Ресурсы открытого доступа: 1

## Задание 14

### Бинарные файлы (модуль pickle)

✦ Создание базы данных (БД) в виде подходящей коллекции (например, списка или словаря) с использованием модуля pickle. Базу данных сохранить в файл dbpickle.dat. Если в задании специально не оговорено количество записей в базе, то их должно быть как минимум десять.

◇ БД: сведения о музыкальных альбомах различных групп (название группы, название альбома, год выпуска, стиль). Удалить из массива все альбомы в стиле pop и techno. Вывести информацию об альбомах в порядке возрастания года выпуска.

📄 import pickle

```
K=int(input('Количество записей в базе данных: '))
```

```
# Создание БД:
```

```
with open('/home/user/code/dbpickle.dat', 'wb') as f:
```

```
    for i in range(K):
```

```
        print('----- Запись №', i+1, '-----')
```

```
        tmp={}
```

```
        tmp['Группа']=input('Группа: ')
```

```
        tmp['Альбом']=input('Альбом: ')
```

```
        tmp['Год выпуска']=int(input('Год выпуска: '))
```

```
        tmp['Стиль']=input('Стиль: ')
```

```
        pickle.dump(tmp, f)
```

```
# Вывод из файла на экран и формирование списка альбомов:
```

```
Музыка=[]
```

```
print(); print('Исходный файл:')
```

```
with open('/home/user/code/dbpickle.dat', 'rb') as f:
```

```
    for i in range(K):
```

```
        tmp=pickle.load(f)
```

```
        print(tmp['Группа'], '\t\t', tmp['Альбом'], '\t\t\t',
```

```
              tmp['Год выпуска'], '\t', tmp['Стиль']) # \t - табуляция
```

```
        Музыка.append(tmp)
```

```
# Удаление всех альбомов в стиле pop и techno:
```

```
OK=True
```

```
while OK:
```

```
    OK=False; i=0
```

```
    while i<=len(Музыка)-1:
```

```
        if Музыка[i]['Стиль']=='pop' or Музыка[i]['Стиль']=='techno':
```

```
            del Музыка[i]; OK=True; break
```

```
        i=i+1
```

```
print(); print('Информация об альбомах без стилей pop и techno:')
```

Подготовка к экзамену

Основная учебная литература: 1, 2

Дополнительная учебная литература: 1, 2, 5

Ресурсы открытого доступа: 1, 2

4.5 Выполнение курсовой работы (проекта). Курсовая работа по дисциплине не предусмотрена.

## 5 РЕСУРСЫ, НЕОБХОДИМЫЕ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

### 5.1 Информационные технологии

Образовательный процесс осуществляется с применением локальных и распределенных информационных технологий (таблицы 4, 5).

#### Локальные информационные технологии

Таблица 4

Группа программных средств	Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства	Аудитория	Реквизиты подтверждающего документа
Офисные программы	LibreOffice	Ауд. No207 «Помещение для самостоятельной работы»	<a href="https://ru.libreoffice.org/about-us/license">https://ru.libreoffice.org/about-us/license</a>
Операционные системы	Manjaro Linux XFCE & KDE	Ауд. No207 «Помещение для самостоятельной работы»	<a href="http://gostrf.com/normadata/1/4293798/4293798256.htm">http://gostrf.com/normadata/1/4293798/4293798256.htm</a>
Научные расчеты	<ul style="list-style-type: none"> <li>• SageMath</li> <li>• Scilab</li> <li>• Maxima</li> <li>• PSPP</li> <li>• Среда статистических вычислений R</li> </ul>	Ауд. No207 «Помещение для самостоятельной работы»	<a href="http://gostrf.com/normadata/1/4293798/4293798256.htm">http://gostrf.com/normadata/1/4293798/4293798256.htm</a>
Графические редакторы	GIMP	Ауд. No207 «Помещение для самостоятельной работы»	<a href="https://www.gimp.org/about/COPYING">https://www.gimp.org/about/COPYING</a>
Браузеры (веб-обозреватели)	Firefox	Ауд. No207 «Помещение для самостоятельной работы»	<a href="https://rusgpl.ru/">https://rusgpl.ru/</a>

#### Распределенные информационные технологии

Таблица 5

Группа	Наименование
Библиотеки и образовательные ресурсы (в том числе персональные сайты преподавателей НГПУ)	Электронная библиотека НГПУ <a href="http://lib.nspu.ru">http://lib.nspu.ru</a>
	Персональные сайты преподавателей НГПУ <a href="http://prepod.nspu.ru">http://prepod.nspu.ru</a>
	Система электронных портфолио студентов НГПУ <a href="https://www.nspu.ru/portfolio/">https://www.nspu.ru/portfolio/</a>
	Электронная библиотека КФ ФГБОУ ВО «НГПУ» <a href="http://lib.kbnspu.ru/">http://lib.kbnspu.ru/</a>

## 5.2 Материально-техническое обеспечение дисциплины

Таблица 6

Номер и наименование (при наличии) помещения для осуществления образовательной деятельности	Перечень основного оборудования	Адрес места осуществления образовательной деятельности (местоположение согласно лицензии)
Учебная аудитория для проведения учебных занятий лекционного типа		
Ауд. №209 «Учебная аудитория лекционного типа занятий»	Комплект учебной мебели, Компьютерное оборудование: Компьютер в комплекте (с выходом в сеть "Интернет" и доступом к электронной информационно-образовательной среде университета) – 8 шт. Интерактивное оборудование: SMART доски – 1 шт.	632387, Новосибирская обл. г. Куйбышев, ул. Молодежная, дом 7
Большой лекционный зал	Комплект учебной мебели	632387, Новосибирская обл. г. Куйбышев, ул. Молодежная
Учебная аудитория для проведения учебных занятий семинарского типа (практические занятия, лабораторные занятия)/ Учебная аудитория для проведения групповых и индивидуальных консультаций/ Учебная аудитория для проведения текущего контроля и промежуточной аттестации		
Ауд. №210 «Компьютерный класс»	Комплект учебной мебели, Компьютерное оборудование: Компьютер в комплекте (с выходом в сеть "Интернет" и доступом к электронной информационно-образовательной среде университета) – 15 шт.	632387, Новосибирская обл. г. Куйбышев, ул. Молодежная, дом 7
Ауд. №211 «Компьютерный класс»	Комплект учебной мебели, Компьютерное оборудование: Компьютер в комплекте (с выходом в сеть "Интернет" и доступом к электронной информационно-образовательной среде университета) – 7 шт.	632387, Новосибирская обл. г. Куйбышев, ул. Молодежная, дом 7
Ауд. №212 «Компьютерный класс»	Комплект учебной мебели, Компьютерное оборудование: Компьютер в комплекте (с выходом в сеть "Интернет" и доступом к электронной информационно-образовательной среде университета) – 8 шт. Интерактивное оборудование: SMART доски – 1 шт.	632387, Новосибирская обл. г. Куйбышев, ул. Молодежная, дом 7
Ауд. №304 «Компьютерный класс»	Комплект учебной мебели, Компьютерное оборудование: Компьютер в комплекте (с выходом в сеть "Интернет" и доступом к электронной информационно-образовательной среде университета) – 9 шт.	632387, Новосибирская обл. г. Куйбышев, ул. Молодежная, дом 7
Помещение для самостоятельной работы обучающихся		
Ауд. №207 «Помещение для самостоятельной работы»	Комплект учебной мебели. Компьютерное оборудование:	632387, Новосибирская обл. г. Куйбышев, ул.



	Компьютеры в комплекте (с выходом в сеть "Интернет" и доступом к электронной информационно-образовательной среде университета) – 8 шт., Печатное и сканирующее оборудование: принтеры – 1 шт.	Молодежная
Помещение для хранения и профилактического обслуживания учебного оборудования		
Ауд. No 217А «Помещение для хранения и профилактического обслуживания оборудования»	Инвентарь: Тестер компьютерный – 1 шт. Специализированный инвентарь – 1 шт. Набор инструментов для оргтехники – 1 шт (28 предметов). Измерительное оборудование: Вольтметры – 1 шт., Мультиметр – 1 шт., Компьютерное оборудование: Компьютер в комплекте – 1 шт. Печатное оборудование: – 1 шт	632387, Новосибирская обл. г. Куйбышев, ул. Молодежная

## 6 ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

### 6.1 Оценочные материалы для проведения текущего контроля успеваемости

Таблица 7

№ п/п	Наименование темы	Код компетенции	Формы проверки
<b>Пятый семестр</b>			
1	Тема 1. Базовые понятия языка Python	ОПК-2	1. Срезовая работа 2. Практическое задание
2	Тема 2. Типы данных, конвертация типов	ОПК-2	1. Срезовая работа 2. Практическое задание
3	Тема 3. Математические и логические операции	ОПК-2	1. Срезовая работа 2. Практическое задание
<b>Шестой семестр</b>			
4	Тема 1. Алгоритмические структуры: следование и ветвление	ОПК-2, ПК-3	1. Срезовая работа 2. Практическое задание
5	Тема 2. Алгоритмические структуры: циклы	ОПК-2, ПК-3	1. Срезовая работа 2. Практическое задание
6	Тема 3. Коллекции: строка, кортеж, список	ОПК-2, ПК-3	1. Срезовая работа 2. Практическое задание
7	Тема 4. Коллекции: множество, словарь	ОПК-2, ПК-3	1. Срезовая работа 2. Практическое задание
8	Тема 5. Методы коллекций	ОПК-2	1. Срезовая работа 2. Практическое задание
<b>Седьмой семестр</b>			
9	Тема 1. Формирование массивов	ПК-3	1. Практическое задание
10	Тема 2. Методы работы с векторами и матрицами	ПК-3	1. Практическое задание
11	Тема 3. Решение задач линейной алгебры	ПК-3	1. Практическое задание
<b>Восьмой семестр</b>			
12	Тема 1. Описание и вызов функций	ОПК-2, ПК-3	1. Срезовая работа 2. Практическое задание
13	Тема 2. Передача параметров и возврат результатов	ОПК-2	1. Практическое задание
14	Тема 3. Область действия имён	ОПК-2	1. Практическое задание
15	Тема 4. Функция open() для работы с текстовыми файлами	ПК-3	1. Срезовая работа 2. Практическое задание
16	Тема 5. Методы модуля pickle для работы с файлами	ПК-3	1. Практическое задание

### 6.2 Оценочные материалы для проведения промежуточной аттестации обучающихся

Таблица 8

Оценочные материалы для промежуточной аттестации	
<b>Пятый семестр (Зачет)</b>	
<b>Код компетенции: ОПК-2</b>	
<ol style="list-style-type: none"> <li>1. Какими критериями должен обладать язык программирования учебного назначения?</li> <li>2. Расскажите об истории создания языка программирования Python.</li> <li>3. Перечислите и расскажите о ключевых возможностях языка Python.</li> <li>4. Что такое динамическая типизация?</li> <li>5. Типы данных: числовой.</li> <li>6. Типы данных: строковый.</li> <li>7. Типы данных: логический.</li> <li>8. Какие стандартные математические операции языка Python вы знаете?</li> <li>9. Составьте список десяти правильных имён идентификаторов. Список должен включать</li> </ol>	

имена как в верхнем, так и в нижнем регистре. Задание сдать в личный аккаунт GoogleКласса.

10. Произведите ввод фамилии, имени и отчества пользователя в отдельных строках.

11. Напишите однострочные комментарии к вашей программе.

12. Составьте код на языке Python для решения задачи. Даны два ненулевых числа. Найти сумму, разность, произведение и частное их квадратов.

**Код компетенции: ПК-3**

1. Каких правил следует придерживаться при создании идентификаторов.

2. Как работает операция присваивания в Python.

3. Расскажите про операцию ввода с клавиатуры.

4. Операция вывода на экран: продемонстрируйте различные способы её применения.

5. Оформление пользовательских комментариев.

6. Как производится конвертация типа данных в Python?

7. Использование модуля math в математических вычислениях.

8. Выведите на экран фамилию, имя, отчество пользователя в отдельных строках.

9. Напишите многострочный комментарий к вашей программе.

10. Дана масса M в килограммах. Используя операцию деления нацело, найти количество полных тонн в ней.

**Шестой семестр (Экзамен)**

**Код компетенции: ОПК-2**

1. Базовые понятия Python: имена, оператор присваивания, ввод, вывод, комментарии.

2. Типы данных: числа, строки, логический тип. Конвертация типов.

3. Коллекции в Python: строка.

4. Коллекции в Python: кортеж.

5. Коллекции в Python: список.

6. Коллекции в Python: множество.

7. Коллекции в Python: словарь.

8. Алгоритмические структуры: следование.

9. Алгоритмические структуры: ветвление, выбор.

10. Алгоритмические структуры: цикл while.

11. Даны два целых числа: A, B. Проверить истинность высказывания: «Числа A и B имеют одинаковую чётность».

12. Даны три целых числа. Найти количество положительных чисел в исходном наборе.

13. Даны два числа. Вывести порядковый номер меньшего из них.

14. Дано целое число. Вывести его строку-описание вида «отрицательное чётное число», «нулевое число», «положительное нечётное число» и т.д.

**Код компетенции: ПК-3**

1. Математические операции. Модуль math.

2. Операции сравнения и логические операции. Приоритет операций.

3. Методы коллекций: основные методы.

4. Методы коллекций: методы списков.

5. Методы коллекций: методы строк (форматирование, поиск).

6. Методы коллекций: методы строк (модификация).

7. Методы коллекций: методы строк (проверка содержимого).

8. Алгоритмические структуры: цикл for, функция range(), break и continue.

9. Ветвь else в циклах. Цикл по словарю.

10. Модуль random. Генератор случайных чисел. Случайный выбор из последовательности.

11. Дано целое число N ( $> 0$ ). Если оно является степенью числа 3, то вывести True, если не является — вывести False.

12. Дано целое число N ( $> 0$ ). Используя операции деления нацело и взятия остатка от деления, найти количество и сумму его цифр.

13. Дано целое число N ( $> 0$ ). С помощью операций деления нацело и взятия остатка от деления определить, имеется ли в записи числа N нечётные цифры. Если имеется, то вывести True, если нет — вывести False.

14. Дано число A ( $> 1$ ). Вывести наибольшее из целых чисел K, для которых сумма  $1 + 1/2 + \dots + 1/K$  будет меньше A, и саму эту сумму.

**Седьмой семестр (Зачет)**

**Код компетенции: ОПК-2**

1. Приведите основные сведения о библиотеке NumPy и массивах в ней.
2. Библиотека NumPy: использование метода reshape для формирования массивов.
3. Библиотека NumPy: получение информации о массиве.
4. Библиотека NumPy: общие методы работы с матрицами.
5. Возможности библиотеки NumPy при решении задач линейной алгебры.
6. Дана матрица размера  $M \times N$ . Для каждого столбца матрицы с чётным номером (2, 4, ...) найти сумму его элементов. Условный оператор не использовать.
7. Дана целочисленная матрица размера  $M \times N$ . Найти номер первого из её столбцов, содержащих максимальное количество одинаковых элементов.

**Код компетенции: ПК-3**

1. Библиотека NumPy: создание массивов на основе коллекций и последовательностей.
2. Библиотека NumPy: примеры стандартных матриц.
3. Библиотека NumPy: поэлементные операции над массивами.
4. Библиотека NumPy: произведение матриц и возведение в степень.
5. Библиотека NumPy: транспонирование матрицы и вычисление определителя.
6. Дана матрица размера  $M \times N$ . Найти количество её столбцов, элементы которых упорядочены по убыванию.
7. Дана целочисленная матрица размера  $M \times N$ . Найти элемент, являющийся максимальным в своей строке и минимальным в своём столбце. Если такой элемент отсутствует, то вывести 0.

**Восьмой семестр (Экзамен)****Код компетенции: ОПК-2**

1. Коллекции в Python: строка.
2. Коллекции в Python: кортеж.
3. Коллекции в Python: список.
4. Коллекции в Python: множество.
5. Коллекции в Python: словарь.
6. Методы коллекций: основные методы.
7. Определение функции, её описание в программе.
8. Механизм возврата результатов работы функции в основную программу.
9. Задания значений параметров функции по умолчанию.
10. Область действия имён при использовании функции в программе.
11. Описать функцию CircleS(R) вещественного типа, находящую площадь круга радиуса R (R — вещественное). С помощью этой функции найти площади трёх кругов с данными радиусами.
12. Описать функцию Even(K) логического типа, возвращающую True, если целый параметр K является чётным, и False в противном случае. С её помощью найти количество чётных чисел в наборе из 10 целых чисел.
13. Описать функцию DegToRad(D) вещественного типа, находящую величину угла в радианах, если дана его величина D в градусах (D — вещественное число,  $0 \leq D < 360$ ). С помощью функции DegToRad перевести из градусов в радианы пять данных углов.
14. Описать функцию IsPalindrome(K), возвращающую True, если целый параметр K ( $> 0$ ) является палиндромом (т.е. его запись читается одинаково слева направо и справа налево), и False в противном случае. С её помощью найти количество палиндромов в наборе из 10 целых положительных чисел.

**Код компетенции: ПК-3**

1. Методы коллекций: методы списков.
2. Методы коллекций: методы строк (форматирование, поиск).
3. Методы коллекций: методы строк (модификация).
4. Методы коллекций: методы строк (проверка содержимого)
5. Вызов функции в программе.
6. Дан символ C и строка S. Удвоить каждое вхождение символа C в строку S.
7. Дана строка, состоящая из русских слов, разделённых пробелами (одним или несколькими). Найти длину самого длинного слова.

## Критерии выставления отметок

Отметка «отлично» / «зачтено» (высокий уровень сформированности компетенций (-ии)) выставляется обучающемуся, который в процессе изучения дисциплины и по результатам промежуточной аттестации: - обнаружил системные знания по всем разделам программы дисциплины / модуля / практики, продемонстрировал способность к их самостоятельному пополнению, в том числе в рамках учебно-исследовательской и научно-исследовательской деятельности; - при выполнении заданий, предусмотренных программой, успешно продемонстрировал осваиваемые в рамках дисциплины / модуля / практики профессиональные умения; - представил результаты выполнения всех заданий для самостоятельной работы полностью и качественно, на творческом уровне, выразил личностную значимость деятельности; - при устном ответе высказал самостоятельное суждение на основе исследования теоретических источников, логично и аргументированно изложил материал, связал теорию с практикой посредством иллюстрирующих примеров, свободно ответил на дополнительные вопросы; - при выполнении письменного задания представил содержательный, структурированный, глубокий анализ сути и путей решения проблемы (задачи, задания);- при выполнении тестовых заданий дал правильные ответы на 85 – 100 % заданий.

Отметка «хорошо» / «зачтено» (средний уровень сформированности компетенций (-ии)) выставляется обучающемуся, который в процессе изучения дисциплины и по результатам промежуточной аттестации: - обнаружил знание основного материала по всем разделам программы дисциплины / модуля / практики в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, продемонстрировал способность к их самостоятельному пополнению; - при выполнении заданий, предусмотренных программой, смог продемонстрировать осваиваемые профессиональные умения, но допустил не принципиальные ошибки в их выполнении, которые смог исправить при незначительной помощи преподавателя; - представил результаты выполнения всех заданий для самостоятельной работы, указанных в программе дисциплины / модуля / практики, при этом задания выполнены полностью и качественно; - при устном ответе объяснил учебный материал, интерпретировал содержание, экстраполировал выводы; - при выполнении письменного задания представил репродуктивную позицию элементы анализа в описании сути и путей решения проблемы (задачи, задания), изложил логическую последовательность вопросов темы; - при выполнении тестовых заданий дал правильные ответы на 75 – 84 % заданий.

Отметка «удовлетворительно» / «зачтено» (пороговый уровень сформированности компетенций(-ии)) выставляется обучающемуся, который в процессе изучения дисциплины и по результатам промежуточной аттестации: - обнаружил знание основного материала по всем разделам программы дисциплины / модуля / практики в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, но знания имеют пробелы и плохо структурированы; при выполнении заданий, предусмотренных программой, в целом смог продемонстрировать осваиваемые профессиональные умения, но допустил ошибки в их выполнении, которые смог исправить при незначительной помощи преподавателя; - представил результаты выполнения всех заданий для самостоятельной работы, указанных в программе дисциплины / модуля / практики, при этом задания выполнены формально, кратко, рефлексия неполная или носит формальный характер, представлено поверхностное описание. - при устном ответе продемонстрировал знание базовых положений и ключевых понятий, верно воспроизвел учебное содержание без использования дополнительного материала; - при выполнении письменного задания представил репродуктивную позицию в описании сути и путей решения проблемы (задачи, задания);- при выполнении тестовых заданий дал правильные ответы на 60 – 74 % заданий.

Отметка «неудовлетворительно» / «незачтено» (компетенция(-ии) не сформирована(-ы)) выставляется обучающемуся, который в процессе изучения дисциплины и по результатам промежуточной аттестации: - обнаружил отсутствие знаний либо фрагментарные знания по основным разделам программы дисциплины / модуля / практики; - при выполнении заданий, предусмотренных программой, не смог продемонстрировать осваиваемые профессиональные

умения (допустил принципиальные ошибки в их выполнении, которые не смог исправить при указании на них преподавателем), либо не выполнил задания; - не выполнил предусмотренные учебным планом практические, лабораторные задания; - не полностью выполнил задания для самостоятельной работы, указанных в программе дисциплины / модуля /практики, либо задания выполнены неверно, очевиден плагиат;- при устном ответе допустил фактические ошибки в использовании научной терминологии и изложении учебного содержания, сделал ложные выводы; - при выполнении тестовых заданий дал правильные ответы на 0 – 59 % заданий.