ГОУ ВПО РОССИЙСКО-АРМЯНСКИЙ (СЛАВЯНСКИЙ) УНИВЕРСИТЕТ

Составлен в соответствии с государственными требованиями к минимуму содержания и уровню подготовки выпускников по направлению 01.03.02 Прикладная математика и информатика и Положением «Об УМКД РАУ».

УТВЕРЖДАЮ:

Директор института математики и информатики
канд.физ.-мат.наук
Дарбинян А.А.
_____ 2023г.

Институт: Математики и инфроматики
Название института

Кафедра: Системное программирование
Название кафедры

Автор(ы): **Сматян Хачатур**

Ученое звание, ученая степень, Ф.И.О

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС

Дисциплина: Б1.В.05 Основы Python
Код и название дисциплины согласно учебному плану

Специальность: 01.03.02 Прикладная математика и информатика
Код и название специальности

Направление: Прикладная математика и информатика
Название направления

ЕРЕВАН

Структура и содержание УМКД

1. **Аннотация**
   1.1. Взаимосвязь дисциплины с другими дисциплинами учебного плана специальности (направления)
   Дисциплина «Специальный курс 2» относится к вариативной части Блока 1 "Дисциплины (модули)" учебного плана. Данная дисциплина логически и содержательно методически связана с такими дисциплинами как «Основы программирования», «Теория алгоритмов и математическая логика», «Объектно-ориентированное программирование», «Основы программирования». Является логически связанной с математическими дисциплинами, рассматривает объекты таких дисциплин как «Дискретная математика» с точки зрения программирования.
   1.2. Требования к исходным уровням знаний, умений и навыков студентов для прохождения дисциплины (что должен знать, уметь и владеть студент для прохождения данной дисциплины): знания парадигмы структурного программирования, умения анализировать структуры данных, навыки разработки алгоритмов, иметь базовые навыки в написании программ на процедурных и объектно-ориентированных языках, быть знакомым с наиболее часто встречающимися структурами данных, уметь ими пользоваться и знать внутреннюю организацию.
   1.3. Предварительное условие для прохождения (дисциплина(ы), изучение которых является необходимой базой для освоения данной дисциплины): Основы информатики, Объектно-ориентированное программирование, Основы программирования, Архитектура компьютеров (язык ассемблера).
2. **Содержание**
   2.1. Цели и задачи дисциплины
   Цели дисциплины: формирование у студентов комплекса основных навыков разработки программ на языке программирования Python 3, организации изолированного окружения разработки и приёмами исследования стороннего исходного кода.
   Задачи дисциплины:
   - всесторонне ознакомить слушателей с языком программирования,
   - предоставить возможность наработки навыков программирования на языке во время решения домашних заданий,
   - представить эффективные приёмы программирования на Python на примере некоторых инструментально-прикладных компонентов
   - осветить современные тенденции программирования, нашедшие отражение в структуре языка, в том числе мультипарадигмальный подход при разработке программ.

   2.2. Требования к уровню освоения содержания дисциплины (какие компетенции (знания, умения и навыки) должны быть сформированы у студента ПОСЛЕ прохождения данной дисциплины)
   В результате изучения данной дисциплины студенты на примере доступного языка программирования Python освоят объектно-ориентированное и функциональное программирование, позволяющие быстро перейти к решению задач в

соответствующих предметных областях. Освоение языка Python позволяет быстро создавать как прототипы программных систем, так и сами программные системы, помогает в интеграции программного обеспечения для решения научных и производственных задач

Трудоемкость дисциплины и виды учебной работы по учебному плану.

Виды учебной работы	Всего, в акад. часах	Распределение по семестрам						
		_I__ сем	_II_ сем	III_ сем	IV сем.	V сем	VI сем.	VII сем.
1	3	4	5	6	7	10	11	
1. Общая трудоемкость изучения дисциплины по семестрам , в т. ч.:	**72**							
1.1. Аудиторные занятия, в т. ч.:	**36**					**36**		
1.1.1. Лекции								
1.1.2. Практические занятия, в т. ч.	**36**					**36**		
1.1.2.1. Обсуждение прикладных проектов								
1.1.2.2. Кейсы								
1.1.2.3. Деловые игры, тренинги								
1.1.2.4. Контрольные работы								
1.1.3. Семинары								
1.1.4. Лабораторные работы								
1.1.5. Другие виды аудиторных занятий								
1.2. Самостоятельная работа, в т. ч.:	**36**					**36**		
1.2.1.Подготовка к экзаменам								
1.2.2.Другие виды самостоятельной работы, в т.ч. (можно указать)								
1.2.2.1. Письменные домашние задания								
1.2.2.2. Курсовые работы								
1.2.2.3. Эссе и рефераты								
1.3. Консультации								
1.4. Другие методы и формы занятий (контроль)								
Итоговый контроль (Экзамен,Зачет, диф. зачет/указать)						**Зачет**		

2.3.2. Распределение объема дисциплины по темам и видам учебной работы

Разделы и темы дисциплины	**Всего (ак. часов)**	**Лекции (ак. часов)**	**Практ. занятия (ак. часов)**	**Семина-ры (ак. часов)**	**Лабор. (ак. часов)**	**Другие виды занятий (ак. часо**

						в)
1	**2=3+4+5+6+7**	**3**	**4**	**5**	**6**	**7**
Модуль 1.	**72**		**36**			
История и место языка Python	**4**		**2**			
Командная строка и объекты	**4**		**2**			
Пространства имён и простейшие операторы	**4**		**2**			
Стандартные типы данных	**4**		**2**			
Логические выражения, условные операторы и цикл	**4**		**2**			
Последовательности и цикл for	**4**		**2**			
Функции и замыкание	**4**		**2**			
Строки	**4**		**2**			
Словари и их применение	**4**		**2**			
Множества	**4**		**2**			
Итераторы и генераторы	**4**		**2**			
Введение в классы	**4**		**2**			
ООП в Python3 и перегрузка операций	**4**		**2**			
Наследование и дескрипторы	**4**		**2**			
Исключения и множественное наследование	**4**		**2**			
Декораторы	**4**		**2**			
Работа с файлами и стандартные модули	**4**		**2**			
Контрольная работа	**4**		**2**			
ИТОГО	**36**		**36**			

2.3.3 Содержание разделов и тем дисциплины

Модуль 1

Тема 1. История и место языка Python в современном мире.

Содержание темы: Начальные сведения о языке программирования Python.

Задания для самостоятельной работы: Изучение литературы по теме "История появления и формирования языка Python".

Тема 2. Командная строка и объекты

Содержание темы: Интерпретатор командной строки, редакторы, объекты и выражения с ними
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 3. Пространства имён и простейшие операторы
Содержание темы: Пространство имен, виды и набор операторов. Операторные скобки
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 4. Стандартные типы данных
Содержание темы: Типы данных, операции применимые к различным типам данным
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 5. Логические выражения, условные операторы и цикл
Содержание темы: Алгебра логики, сравнения, условные операторы, цикл while
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 6. Последовательности и цикл for
Содержание темы: цикл for, последовательности, виды последовательностей, операции
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 7. Функции и замыкание
Содержание темы: Функция Python3 как именованный алгоритм. Задание и вызов функции. Распаковка и запаковка параметров. Замыкание функции.
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 8. Строки
Содержание темы: Строки как последовательности. Методы. Форматирование строк. Байтовые строки.
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 9. Словари и их применение
Содержание темы: Словари. Методы. Применение.
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 10. Множества
Содержание темы: Множество, задание, работы с элементами, операции над множествами
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 11. Итераторы и генераторы
Содержание темы: Итераторы, генераторы, параметрические генераторы
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 12. Итераторы и генераторы
Содержание темы: Итераторы, генераторы, параметрические генераторы
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 13. Введение в классы
Содержание темы: Классы, объявление класса, поля и методы класса, экземпляр класса.
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 14. ООП в Python3 и перегрузка операций
Содержание темы: Объектная модель Python. Перегрузка операций, специальные методы.
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 15. Наследование и дескрипторы
Содержание темы: Иерархия пространств имён. Наследование. Дескрипторы
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме.

Тема 15. Исключения и множественное наследование
Содержание темы: Множественное наследование, MRO C3. Исключения. Собственные исключения. Оператор with
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме

Тема 16. Декораторы
Содержание темы: Декораторы, параметрические декораторы, декораторы методов и классов
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме
Тема 17. Работа с файлами и стандартные модули
Содержание темы: Понятие модуля. Обзор стандартной библиотеки. модули расширения. Подключение к модулю.
Задания для самостоятельной работы: Изучение литературы по теме, написание программ по теме

Тема 18. Контрольная работа

Распределение весов по формам контроля

	Вес формы текущего контроля в результирующей оценке текущего контроля			Вес формы промежуточного контроля и результирующей оценки текущего контроля в итоговой оценке промежуточного контроля			Вес итоговых оценок промежуточных контролей в результирующей оценке промежуточного контроля	Вес оценки результирующей оценки промежуточных контролей и оценки итогового контроля в результирующей оценке итогового контроля
Вид учебной работы/контроля	**М1**	**М2**	**М3**	**М1**	**М2**	**М3**		
Контрольная работа						1		
Письменные домашние задания			1					
Вес результирующей оценки текущего контроля в итоговых оценках промежуточных контролей						1		
Вес итоговой оценки 1-го промежуточного контроля в результирующей оценке промежуточных контролей							0	
Вес итоговой оценки 2-го промежуточного контроля в результирующей оценке промежуточных контролей							0	
Вес итоговой оценки 3-го промежуточного контроля в результирующей оценке промежуточных контролей т.д.							1	
Вес результирующей оценки промежуточных контролей в результирующей оценке итогового контроля								0,5
Экзамен/зачет (оценка итогового контроля)								0,5
	∑ = 1	**∑ = 1**	**∑ = 1**	**∑ = 1**	**∑ = 1**	**∑ = 1**	**∑ = 1**	**∑ = 1**

2.3.4 **Краткое содержание семинарских/практических занятий и лабораторного практикума**

Примерный список заданий для проведения текущей и промежуточной аттестации:

1. Написать программу, которая выводит строку `Hello, world` (в точности)
2. Ввести три *python3-объекта* через запятую (с помощью `eval (input())`), вывести их через пробел, причём в обратном порядке

3. Написать *функцию* по имени **function ()** от трёх параметров, которая возвращает кортеж из двух объектов — последнего и первого
4. Ввести через запятую три числа: a, b и c, вывести все вещественные решения уравнения ax2+bx+c=0. При a≠0 это уравнение превращается в квадратное. Решения выводить через пробел в порядке возрастания, если решений нет, вывести **0**, если их бесконечно много — **-1**.

5. Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 ... (число повторяется столько раз, чему равно). На вход программе передаётся неотрицательное целое число n — столько элементов последовательности должна отобразить программа. На выходе ожидается последовательность чисел, записанных через пробел в одну строку. Например, если n = 7, то программа должна вывести 1 2 2 3 3 3 4.
6. Написать *генератор* по имени **generator ()**, которому передаётся параметр — целое число (возможно, отрицательное!), а он возвращает по очереди три значения: это число с обратным знаком, это число, преобразованное в строку и количество десятков в этом числе..
7. Напишите программу, на вход которой подаётся прямоугольная матрица в виде последовательности строк, заканчивающихся строкой, содержащей только строку "end" (без кавычек). Программа должна вывести матрицу того же размера, у которой каждый элемент в позиции i, j равен сумме элементов первой матрицы на позициях (i-1, j), (i+1, j), (i, j-1), (i, j+1). У крайних символов соседний элемент находится с противоположной стороны матрицы. В случае одной строки/столбца элемент сам себе является соседом по соответствующему направлению.
8. Ввести текст, состоящий из нескольких строк (заканчивается пустой строкой).
9. Ввести строку, содержащую произвольные символы (кроме символа «@»). Затем ввести строку-шаблон, которая может содержать символы
10. Написать **функцию** squares(w,h,(X,Y,s,c),...) со следующими параметрами: w и h — ширина и высота «экрана», за которыми следуют 0 или больше 4-элементных последовательностей вида (X,Y,s,c), где X и Y — координаты левого верхнего угла квадрата, s — длина его стороны
11. Ввести строку, содержащую произвольные символы (кроме символа «@»). Затем ввести строку-шаблон, которая может содержать символы @. Проверить, содержится ли в исходной строке подстрока, совпадающая со строкой-шаблоном везде, кроме символов @; на месте @ в исходной строке должен стоять ровно один произвольный символ.
12. Ввести построчно текст, состоящий из пробелов, переводов строки и латинских букв, и заканчивающийся пустой строкой. Вывести слово, которое чаще других встречается в тексте, если оно такое одно, и ---, если таких слов несколько.

13. Написать **класс** `vector`, представляющий нечто, похожее на вектор. Должна поддерживаться операция вывода в формате, представленном в примере, конструирование из произвольной числовой последовательности ненулевой длины, а также сложение с числовой последовательностью такой же длины (в том числе с другим `vector`)
14. Написать **класс** `we`, содержащий поле `count`, в котором хранится информация о количестве существующих экземпляров этого класса.

2.3.5 Материально-техническое обеспечение дисциплины

Основная литература:

1. Allen B. Downey. Think Python: How to Think Like a Computer Scientist, 2nd Edition, — O'Reilly, 2016
2. Mark Summerfield. Programming in Python 3, A Complete Introduction to the Python Language, Second Edition. Addison-Wesley, 2010

Перечень ресурсов информационно-телекоммуникационной сети:

1. http://greenteapress.com/thinkpython2/html/index.html
2. https://docs.python.org/3/tutorial/index.html
3. http://pythontutor.com/

Программное обеспечение:

1. Интерпретатор языка программирования Python3.6 с комплектом стандартных модулей, включая IDE Idle3