



1920

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

филиал Федерального государственного бюджетного образовательного
учреждения высшего образования

«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

в г. Новороссийске

Кафедра информатики и математики

УТВЕРЖДАЮ

Проректор по работе с филиалами

ФГОУ ВО «Кубанский

государственный университет»

А.А.Евдокимов



01.07 2019 г.

РАБОЧАЯ ПРОГРАММА

**Б1.О.30 ИНТЕРПРЕТИРУЕМЫЕ ЯЗЫКИ
ПРОГРАММИРОВАНИЯ**

Направление подготовки: 01.03.02 Прикладная математика и информатика

Направленность (профиль): Математическое и информационное обеспечение
экономической деятельности

Программа подготовки: академическая

Форма обучения: очная

Квалификация (степень) выпускника: Бакалавр

Краснодар 2019

Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 01.03.02 Прикладная математика и информатика, утвержденного приказом Министерства образования и науки Российской Федерации № 9 от 10 января 2018 года.

Программу составил(и):

И.Г.Рзун , доцент канд.физ.-мат.наук



С.В. Дьяченко доцент канд.физ.-мат.наук



Рабочая программа дисциплины Интерпретируемые языки программирования обсуждена и утверждена на заседании кафедры Информатики и математики
протокол № 11 от 20.06. 2019 г.

Заведующий кафедрой (выпускающей) Рзун И.Г.



Рабочая программа одобрена на заседании учебно-методической комиссии филиала УГС 01.00.00 «Математика и механика»
21.06.2019 г. протокол № 10

Председатель УМК



С.В. Дьяченко

Рецензенты:

Кунина М.К. Директор по развитию ООО «АЙТИ БИЗНЕС ЮГ»

Адамович А.Е. Директор ООО «Финам - Новороссийск»

Содержание рабочей программы дисциплины

1 Цели и задачи изучения дисциплины	4
1.1 Цель освоения дисциплины	4
1.2 Задачи дисциплины	4
Основными задачами курса является:	4
1.3 Место дисциплины в структуре образовательной программы	4
1.4 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы	5
2. Структура и содержание дисциплины	8
2.1 Распределение трудоёмкости дисциплины по видам работ	8
2.2 Структура дисциплины:	9
2.3 Содержание разделов дисциплины:	9
2.3.1 Занятия лекционного типа	10
2.3.2 Занятия практического типа	11
2.3.3 Лабораторные занятия	11
2.3.4 Примерная тематика курсовых работ (проектов)	14
2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	14
Примерные вопросы для самостоятельной работы	15
3. Образовательные технологии	19
4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации	20
4.1 Фонд оценочных средств для проведения текущего контроля	20
4.2 Фонд оценочных средств для проведения промежуточной аттестации	21
5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	22
5.1 Основная литература:	22
5.2 Дополнительная литература:	23
5.3. Периодические издания:	23
6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	23
7. Методические указания для обучающихся по освоению дисциплины	24
8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)	26
8.1 Перечень информационных технологий.	26
8.2 Перечень необходимого программного обеспечения.	26
8.3 Перечень информационных справочных систем:	26
9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине	26

1 Цели и задачи изучения дисциплины

1.1 Цель освоения дисциплины

Целью дисциплины «Интерпретируемые языки программирования» является:

- изучение основных возможностей языка программирования Python, технологий структурного, модульного и объектно-ориентированного проектирования программ;
- овладение студентами практическими навыками написания и отладки программ;
- изучение современных методов визуального, объектно-ориентированного проектирования приложений, использующих в своей работе язык программирования Python.

При этом основное внимание необходимо уделить не рассмотрению максимально широкого круга вопросов, а на получение студентами глубоких знаний по фундаментальным основам информатики, на формирование у них общего информационного мировоззрения и на развитие алгоритмического мышления.

1.2 Задачи дисциплины

Основными задачами курса является:

- изучение объектно-ориентированного программирования на языке Python;
- изучение способов разработки оконно - графического интерфейса программ на языке Python;
- изучение способов создания сетевых программ на языке Python;
- изучение способов создания многопоточных программ на языке Python;
- овладение современными средами разработки программ на языке Python;
- способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям;
- способность критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности;
- способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения.

Изучение курса позволит студентам получить теоретическую базу, необходимую для успешного усвоения материала учебных дисциплин, связанных с программированием на различных языках программирования в различных средах, а в дальнейшем для их успешной работы и решения производственных задач на ЭВМ.

Студенты должны научиться выполнять разработку программ в различных визуальных средах, разрабатываемых в поддержку современных языков программирования. Уметь пользоваться широким спектром возможностей, предоставляемых этими средами.

1.3 Место дисциплины в структуре образовательной программы

Дисциплина «Интерпретируемые языки программирования» относится к выборочной части учебного плана. Она является логически и содержательно - методически связана с такими дисциплинами как «Основы программирования», «Методы программирования».

Данная дисциплина базируется на следующих дисциплинах: “Основы программирования”, “Методы программирования”, “Базы данных”. Знания, полученные при освоении данной дисциплины, необходимы при выполнении выпускной квалификационной работы.

1.4 Перечень планируемых результатов обучения по дисциплине , соотнесенных с планируемыми результатами освоения образовательной программы
Изучение данной учебной дисциплины направлено на формирование у обучающихся компетенций В процессе освоения дисциплины у студента формируются следующие компетенции: ОПК-4, ПК-4, ПК-5

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
1.	ОПК-4	Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	цели, задачи и особенности информационного поиска, значение и место библиографического поиска как важной части информационного поиска, особенности библиографического поиска; организационно-правовые основы информационной безопасности; методы обеспечения информационной безопасности; современные информационно-коммуникационные технологии.	- практически оценивать информацию с позиций ее актуальности, надежности и полноты; - применять современные информационные технологии систематизации и обработки информации; - проводить тематический и индексный поиск по заданному критерию; - применять современные операционные среды и информационно-коммуникационные технологии для информационного и библиографического поиска; - применять методы защиты информации при проектировании и разработке программных продуктов.	- навыками информационного и библиографического поиска с возможным использованием разных источников информации: карточных и электронных каталогов библиотек, библиографических картотек библиографических изданий, ресурсов открытого Интернета, библиографических баз данных. - способность решать стандартные задачи профессиональной деятельности с использованием результатов информационного и библиографического поиска.

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
					- навыками обеспечения защиты информации в процессе решения задач профессиональной деятельности.
2	ПК-4	Способен активно участвовать в разработке системного и прикладного программного обеспечения	круг задач профессиональной деятельности, в том числе задачи профессиональной деятельности, подлежащие решению в научно-исследовательском и производственном коллективе; основные этапы выполнения научно-исследовательской работы и работы по решению прикладных задач профессиональной деятельности; технологии проектной работы группы исполнителей по решению научно-исследовательской или	решать на современном уровне задачи своей профессиональной деятельности: ставить цели, выделять задачи работы и определять методы их достижения при решении задач профессиональной деятельности, проводить научные исследования и получать новые научные и прикладные результаты, анализировать полученные результаты, делать выводы в соответствии с поставленными целями; разрабатывать архитектуру и информационное обеспечение компьютерных сетей, разрабатывать системы цифровой	Навыками проектной работы по решению задач профессиональной деятельности; опытом разработки и исследования алгоритмов, вычислительных моделей и моделей данных для реализации элементов новых (или известных) сервисов систем информационных технологий, средствами администрирования и методами управления безопасностью компьютерных сетей.

№ п.п.	Индекс компет енции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
			производствен ной задачи, системы цифровой обработки изображений, средства компьютерной графики, мультимедиа и автоматизиров анного проектирован ия, математически е методы моделировани я информацион ных и имитационных моделей по тематике выполняемых научно- исследователь ских прикладных задач или опытно- конструкторск их работ, методику исследования автоматизиров анных систем и средств обработки информации.	обработки изображений, средства компьютерной графики, мультимедиа и автоматизирован ного проектирования	
3	ПК-5	Способен применять основные алгоритмические и программные решения в области информационно- коммуникационных технологий, а также	информационн ые источники (в том числе сети Интернет), необходимые для работы в профессиональ ной сфере;	использовать сетевые информационны е ресурсы в профессиональн ой деятельности с обеспечением защиты	навыками пользования сетевыми информацион ными ресурсами с обеспечением защиты

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
		участвовать в их разработке	организацию и структуру источников информации в глобальных компьютерных сетях; номенклатуру информационных изданий, услуг, баз данных, предлагаемых библиотеками и органами НТИ страны.	информации. собирать материал для выполнения научно-исследовательской работы с использованием глобальных компьютерных сетей. пользоваться библиотечным банком России; осуществлять поиск литературы в автоматизированном режиме по библиографическим базам данных; самостоятельно изучать информационные источники, применять их в практической работе.	информации. навыками работы в глобальных компьютерных сетях; навыками поиска, анализа и отбора информации в различных источниках, включая сетевые ресурсы сети Интернет. методиками информационного поиска в сети интернет.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 2 зач.ед. (72 часов), их распределение по видам работ представлено в таблице (для студентов ОФО).

Вид учебной работы	Всего часов	Семестры (часы)
		6
Контактная работа, в том числе:	50,3	50,3
Аудиторные занятия (всего):	48	48
Занятия лекционного типа	16	16
Лабораторные занятия	32	32
Занятия семинарского типа (семинары, практические занятия)		
Иная контактная работа:		
Контроль самостоятельной работы (КСР)	2	2
Промежуточная аттестация (ИКР)	0,2	0,2
Самостоятельная работа, в том числе:	21,8	21,8
Курсовая работа		

Проработка учебного (теоретического) материала	10	10
Выполнение индивидуальных заданий	11,8	11,8
Реферат		
Подготовка к текущему контролю		
Контроль: зачет		
Подготовка к зачету		
Общая трудоемкость	час.	72
	в том числе контактная работа	50,3
	зач. ед	2

2.2 Структура дисциплины:

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.
Разделы дисциплины, изучаемые в 6 семестре (для студентов ОФО)

	Наименование разделов	Количество часов						
		Всего	Контактная работа				Контр оль	Самост ятельн ая работа
			Л	ЛР	КСР	ИК Р		
1	2	3	4	5	6	7	8	9
1	Типы данных и операторы языка Python	16	4	8				4
2	Функции. Ввод - вывод	18	4	8				6
3	Объектно-ориентированное программирование	16	4	8				4
4	Конструкторы и деструкторы Наследование и перегрузка	21,8	4	8	2			7,8
	Итого по дисциплине:	71,8	16	32	2			21,8
	Промежуточная аттестация (ИКР)	0,2				0,2		
	<i>Контроль</i>							
	<i>Всего:</i>	72	16	32	2	0,2		21,8

2.3 Содержание разделов дисциплины:

Раздел 1. Основные типы данных. Константы. Выражения и операторы Python. Операции логического сдвига. Условные операторы. Операторы циклов. Указатели и ссылки. Разыменование указателей. Адресная арифметика. Одномерные массивы. Многомерные массивы. ОПК-4, ПК-4, ПК-5

Раздел 2. Функции в Python. Область действия имен. Способы передачи параметров в функции. Параметры командной строки. Структуры. Массивы структур. Указатели на структуры. Массивы структур с использованием указателей. Объединения. Директивы препроцессора. Макросы. Классы памяти. Функции ввода-вывода символов. Функции ввода-вывода строк. Форматированный ввод-вывод. Файловый ввод-вывод. Функции ввода-вывода символов. Файловый ввод-вывод. Функции ввода-вывода строк. Файловый форматированный ввод-вывод. Неформатированный ввод-вывод в нестандартные файловые потоки. ОПК-4, ПК-4, ПК-5

Раздел 3. Определение классов в Python. Свойства и методы. Работа с объектами через указатели. Неполное объявление класса. ОПК-4, ПК-4, ПК-5

Раздел 4. Конструкторы и деструкторы. Конструкторы с параметрами. Инициализация. Конструкторы по умолчанию. Статические данные и функции. Дружественные функции и классы. Наследование. Перегрузка функций. Перегрузка операций. ОПК-4, ПК-4, ПК-5

2.3.1 Занятия лекционного типа

№	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
1	Типы данных и операторы языка Python	Основные типы данных. Константы. Выражения и операторы Python. Операции логического сдвига. Условные операторы. Операторы циклов. Указатели и ссылки. Разыменование указателей. Адресная арифметика. Одномерные массивы. Многомерные массивы.	Вопросы для устного опроса
2	Функции. Ввод - вывод	Функции в Python. Область действия имен. Способы передачи параметров в функции. Параметры командной строки. Структуры. Массивы структур. Указатели на структуры. Массивы структур с использованием указателей. Объединения. Директивы препроцессора. Макросы. Классы памяти. Функции ввода-вывода символов. Функции ввода-вывода строк. Форматированный ввод-вывод. Файловый ввод-вывод. Функции ввода-вывода символов. Файловый ввод-вывод. Функции ввода-вывода строк. Файловый форматированный ввод-вывод. Неформатированный ввод-вывод в нестандартные файловые потоки.	Вопросы для устного опроса
3	Объектно-ориентированное программирование	Определение классов в Python. Свойства и методы. Работа с объектами через указатели. Неполное объявление класса.	Вопросы для устного опроса
4	Конструкторы и деструкторы. Наследование и перегрузка	Конструкторы и деструкторы. Конструкторы с параметрами. Инициализация. Конструкторы по умолчанию. Статические данные и функции. Дружественные функции и классы. Наследование. Перегрузка функций. Перегрузка	Вопросы для устного опроса

		операций.	
--	--	-----------	--

Вопросы для устного опроса

1. Основные типы данных. Константы.
2. Выражения и операторы Python. Операции логического сдвига.
3. Условные операторы.
4. Операторы циклов.
5. Указатели и ссылки. Разыменование указателей.
6. Адресная арифметика.
7. Одномерные массивы.
8. Многомерные массивы.
9. Функции в Python.
10. Область действия имен.
11. Способы передачи параметров в функции.
12. Параметры командной строки.
13. Структуры. Массивы структур.
14. Указатели на структуры. Массивы структур с использованием указателей.
15. Объединения.
16. Макросы.
17. Функции ввода-вывода строк.
18. Форматированный ввод-вывод.
19. Файловый ввод-вывод. Функции ввода-вывода строк.
20. Файловый форматированный ввод-вывод.
21. Неформатированный ввод-вывод в нестандартные файловые потоки.
22. Определение классов в Python. Свойства и методы.
23. Работа с объектами через указатели.
24. Конструкторы и деструкторы.
25. Конструкторы с параметрами. Инициализация.
26. Конструкторы по умолчанию.
27. Статические данные и функции.
28. Массивы объектов.
29. Контейнеры. Векторы. Итераторы. Алгоритмы.
30. Дружественные функции и классы.
31. Наследование.
32. Перегрузка функций
33. Перегрузка операций.

2.3.2 Занятия практического типа

Практические занятия - не предусмотрены

2.3.3 Лабораторные занятия

№	Наименование раздела	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1	Типы данных и операторы языка Python	Основные типы данных. Константы. Выражения и операторы Python. Операции логического сдвига. Условные операторы. Операторы циклов. Указатели и ссылки. Разыменование указателей. Адресная арифметика.	лабораторная

		Одномерные массивы. Многомерные массивы.	
2	Функции. Ввод - вывод	Функции в Python. Область действия имен. Способы передачи параметров в функции. Параметры командной строки. Структуры. Массивы структур. Указатели на структуры. Массивы структур с использованием указателей. Объединения. Директивы препроцессора. Макросы. Классы памяти. Функции ввода-вывода символов. Функции ввода-вывода строк. Форматированный ввод-вывод. Файловый ввод-вывод. Функции ввода-вывода символов. Файловый ввод-вывод. Функции ввода-вывода строк. Файловый форматированный ввод-вывод. Неформатированный ввод-вывод в нестандартные файловые потоки.	лабораторная
3	Объектно-ориентированное программирование	Определение классов в Python. Свойства и методы. Работа с объектами через указатели. Неполное объявление класса.	лабораторная
4	Конструкторы и деструкторы Наследование и перегрузка	Конструкторы и деструкторы. Конструкторы с параметрами. Инициализация. Конструкторы по умолчанию. Статические данные и функции. Дружественные функции и классы. Наследование. Перегрузка функций. Перегрузка операций.	лабораторная

Примерное содержание лабораторных работ

1. Создать массив типа *int* на 10 элементов и заполнить его случайными числами от 7 до 14. После заполнения перезаписать все числа, которые больше десяти: от хранимого значения отнять 10. Например в ячейке хранится число 12: $12 - 10 = 2$. Записать в эту ячейку 2. Записывая новое значение, используйте составные (комбинированные) операторы.

2. Заполнить массив из 50-ти элементов нечётными числами от 1 до 99. (используйте операцию остаток от деления, чтобы проверить число на чётность)

3. Объявить три массива. Первые два заполнить случайными значениями от 10 до 30. В элементы третьего массива записать сумму соответствующих элементов первых двух массивов. (в нулевую ячейку третьего – сумму нулевых ячеек первого и второго массивов и так далее). Затем найти среднее арифметическое элементов третьего массива, максимальное значение и минимальное значение, которые он хранит.

4. Объявить два целочисленных массива с разными размерами и написать функцию, которая заполняет их элементы значениями и показывает на экран. Функция должна принимать два параметра – массив и его размер.

5. Необходимо создать двумерный массив 5 x 5. Далее написать функцию, которая заполнит его случайными числами от 30 до 60. Создать еще две функции, которые находят максимальный и минимальный элементы этого двумерного массива.

6. Написать игру в которой имитируется бросание кубиков компьютером и пользователем. В игре 2 кубика и на каждом из них может выпасть от 1 до 6 очков. Реализовать определение программой первого ходящего. Каждый делает по четыре броска. После бросков показать, нарисованные символами кубики и количество очков, выпавших на них. После пары бросков (бросок компьютера + бросок пользователя) выводить на экран промежуточный результат – количество набранных очков игроком и компьютером. В конце сообщить о том, кто выиграл по итогам всех бросков.

7. Создать класс *Tiles* (кафель), который будет содержать поля с открытым доступом: *brand*, *size_h*, *size_w*, *price* и метод класса *getData()*. В главной функции объявить пару объектов класса и внести данные в поля. Затем отобразить их, вызвав метод *getData()*.

8. Определить класс *Children*, который содержит такие поля (члены класса): закрытые – имя ребенка, фамилию и возраст, публичные – методы ввода данных и отображения их на экран. Объявить два объекта класса, внести данные и показать их.

9. Преобразовать строки двумерного массива в столбцы. Например было:

11	22
33	44
55	66
77	88

стало:

11	33	55	77
22	44	66	88

Класс будет содержать два метода: один заполнит массив значениями, второй произведет замену значений строк на значения столбцов.

10. Объявите указатель на массив типа *double* и предложите пользователю выбрать его размер. Далее напишите четыре функции: первая должна выделить память для массива, вторая – заполнить ячейки данными, третья – показать данные на экран, четвертая – освободить занимаемую память. Программа должна предлагать пользователю продолжать работу (создавать новые динамические массивы) или выйти из программы.

11. Объявите указатель на массив типа *int* и выделите память для 12-ти элементов. Необходимо написать функцию, которая поменяет значения четных и нечетных ячеек массива. Например, есть массив из 4-х элементов:

	ячейка 0	ячейка 1	ячейка 2	ячейка 3
Исходные данные массива	1	2	3	4
	ячейка 0	ячейка 1	ячейка 2	ячейка 3
Данные после работы функции	2	1	4	3

12. Объявить и заполнить двумерный динамический массив случайными числами от 10 до 50. Показать его на экран. Для заполнения и показа на экран написать отдельные функции. (подсказка: функции должны принимать три параметра – указатель на динамический массив, количество строк, количество столбцов). Количество строк и столбцов выбирает пользователь.

2.3.4 Примерная тематика курсовых работ (проектов)

Курсовые работы не предусмотрены

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

№	Наименование раздела	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1.	Проработка учебного (теоретического) материала	Огнева, М. В. Программирование на языке Python: практический курс : учебное пособие для бакалавриата и специалитета / М. В. Огнева, Е. В. Кудрина. — М. : Издательство Юрайт, 2017. [Электронный ресурс] - https://www.biblio-online.ru/viewer/7670D7EC-AC37-4675-8EAE-DD671BC6D0E4 , 05.10.2017. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — М. : Издательство Юрайт, 2017. [Электронный ресурс] - https://www.biblio-online.ru/viewer/DCD7188A-4AAB-4B59-84CD-40A05E3676A7 , 05.10.2017.
2.	Выполнение индивидуальных заданий	Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для академического бакалавриата / С. В. Зыков. — М. : Издательство Юрайт, 2017. [Электронный ресурс] - https://www.biblio-online.ru/viewer/E006A65E-B936-4856-B49E-1BA48CF1A52F , 05.10.2017. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — М. : Издательство Юрайт, 2017. [Электронный ресурс] - https://www.biblio-online.ru/viewer/DCD7188A-4AAB-4B59-84CD-40A05E3676A7 , 05.10.2017.

При изучении дисциплины «Интерпретируемые языки программирования» обязательными являются следующие формы самостоятельной работы:

- разбор теоретического материала по пособиям, конспектам лекций;
- самостоятельное изучение указанных теоретических вопросов;
- решение задач по темам занятий;
- подготовка к зачету

Примерные вопросы для самостоятельной работы.

1. Основные типы данных. Константы.
2. Выражения и операторы Python. Операции логического сдвига.
3. Условные операторы.
4. Операторы циклов.
5. Указатели и ссылки. Разыменование указателей.
6. Адресная арифметика.
7. Одномерные массивы.
8. Многомерные массивы.
9. Функции в Python.
10. Область действия имен.
11. Способы передачи параметров в функции.
12. Параметры командной строки.
13. Структуры. Массивы структур.
14. Указатели на структуры. Массивы структур с использованием указателей.
15. Объединения.
16. Макросы.
17. Функции ввода-вывода строк.
18. Форматированный ввод-вывод.
19. Файловый ввод-вывод. Функции ввода-вывода строк.
20. Файловый форматированный ввод-вывод.
21. Неформатированный ввод-вывод в нестандартные файловые потоки.
22. Определение классов в Python. Свойства и методы.
23. Работа с объектами через указатели.
24. Конструкторы и деструкторы.
25. Конструкторы с параметрами. Инициализация.
26. Конструкторы по умолчанию.
27. Статические данные и функции.
28. Массивы объектов.
29. Контейнеры. Векторы. Итераторы. Алгоритмы.
30. Дружественные функции и классы.
31. Наследование.
32. Перегрузка функций
33. Перегрузка операций.

Примерное содержание самостоятельных работ

Самостоятельная работа студента заключается в самостоятельном выборе темы проекта-программы, его архитектуры и полное создание его с основными функциональными возможностями.

Вопросы теста по языку Python.

Тест 1.

1. Используя директиву `#define`, как бы вы описали именованную константу, которая возвращает число секунд в году? Високосными годами следует пренебречь.

```
#define SECONDS_PER_YEAR (60UL * 60UL * 24UL * 365UL)
```

2. Напишите «стандартный» макрос MIN. То есть, макрос, который берет два аргумента и возвращает меньший из них.

```
#define MIN(A,B) ((A) <= (B) ? (A) : (B))
```

3. Каково назначение директивы препроцессора #error?

4. При программировании встраиваемых систем часто используются бесконечные циклы. Как реализовать бесконечный цикл в Python?

5. Используя переменную «a», запишите объявления для:

- (a) Целого;
- (b) Указателя на целое;
- (c) Указателя на указатель на целое;
- (d) Массива из десяти целых;
- (e) Массива из десяти указателей на целые;
- (f) Указателя на массив из десяти целых;
- (g) Указателя на функцию, которая принимает целочисленный аргумент и возвращает целое;
- (h) Массива из десяти указателей на функции, которая принимает целочисленный аргумент и возвращает целое;

6. В каких случаях используется ключевое слово static?

7. Что означает ключевое слово const?

8. Что означает ключевое слово volatile? Приведите три различных примера его использования.

9. При программировании встраиваемых систем приходится часто манипулировать битами в регистрах или переменных. Дана целая переменная «a», напишите два фрагмента кода. Первый должен установить 3-ий бит этой переменной. Второй должен очищать его. В обоих случаях, другие биты должны остаться без изменений.

11. Что делает следующий фрагмент кода и почему?

```
void foo(void)
{
    unsigned int a = 6;
    int b = -20;
    (a+b > 6) ? puts("> 6") : puts("<= 6");
}
```

12. Прокомментируйте следующий фрагмент кода:

```
unsigned int zero = 0;
unsigned int compzero = 0xFFFF; /*1's complement of zero */
```

13. Что делает следующий фрагмент кода и почему?

```
char *ptr;  
if ((ptr = (char *)malloc(0)) == NULL) {  
    puts("Got a null pointer");  
}  
else {  
    puts("Got a valid pointer");  
}
```

14. Си позволяет некоторые ужасные конструкции. Допустима ли эта конструкция, и если да, то что делает этот код?

```
int a = 5, b = 7, c;
```

```
c = a+++b;
```

Тест 2.

Зарезервированными словами в языке программирования Python, являются:

- a. ☐ fortran
- b. ☐ paskal
- c. ☐ integral

ключевыми словами в языке программирования Python, являются:

- a. ☐ extern
- b. ☐ back
- c. ☐ return

Формат константы с плавающей точкой в языке программирования Python, имеет вид: [цифры].[индекс] [E|e [+|-] цифры]

- a. ☒ Неправильно
- b. ☒ Правильно

Тип int в языке программирования Python, имеет следующий диапазон:

- a. ☒ 0X8000 - 0XFFFF
- b. ☒ 0 - 077777

функции в языке программирования Python, не могут возвращать:

- a. ☒ массивы
- b. ☒ функции

Ключевые слова для определения Целых типов данных в языке программирования Python, являются:

- a. ☐ unsigned
- b. ☐ signed
- c. ☐ short
- d. ☐ long double

- e. ☐ long

Ключевые слова для определения плавающих типов данных в языке программирования Python, являются:

- a. ☐ float
b. ☐ double
c. ☐ extra double
d. ☐ long float
e. ☐ long double

Величина типа double в языке программирования Python, занимает:

- a. ☐ 8 бит в памяти
b. ☐ 4 бита в памяти
c. ☐ 4 байта в памяти
d. ☐ 24 бита в памяти
e. ☐ 2 бита в памяти

Величина с модификатором типа float языка программирования Python, занимает:

- a. ☐ 4 байта в памяти
b. ☐ 8 байт в памяти
c. ☐ 8 бит в памяти
d. ☐ 16 бит в памяти
e. ☐ 2 байта в памяти

В структуре struct языка программирования Python, обязательно должен быть указан:

- a. ☐ 1 индекс
b. ☐ 1 указатель
c. ☐ 1 тип
d. ☐ ничего
e. ☐ несколько компонентов

Элементом структуры языка программирования Python, может быть:

- a. ☐ байтовое поле
b. ☐ адрес байтового поля
c. ☐ битовое поле
d. ☐ адрес битового поля
e. ☐ Поле нулевой длины

Согласно письма Министерства образования и науки РФ № МОН-25486 от 21.06.2017г «О разработке адаптированных образовательных программ» -Разработка адаптивной программы необходима в случае наличия в образовательной организации хотя бы одного обучающегося с ограниченными возможностями здоровья.

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии

С точки зрения применяемых методов используются как традиционные информационно-объяснительные лекции, так и интерактивная подача материала с мультимедийной системой. Компьютерные технологии в данном случае обеспечивают возможность разнопланового отображения алгоритмов и демонстрационного материала. Такое сочетание позволяет оптимально использовать отведенное время и раскрывать логику и содержание дисциплины.

Лекции представляют собой систематические обзоры основных аспектов дисциплины.

Лабораторные занятия позволяют научить применять теоретические знания при решении и исследовании конкретных задач. Лабораторные занятия проводятся в компьютерных классах, при этом практикуется работа в группах. Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы. Это обусловлено тем, что в процессе исследования часто встречаются задачи, для которых единых подходов не существует. Каждая конкретная задача при своем исследовании имеет множество подходов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

При освоении дисциплины используются следующие сочетания видов учебной работы с методами и формами активизации познавательной деятельности бакалавров для достижения запланированных результатов обучения и формирования компетенций.

В соответствии с требованиями ФГОС ВО по направлению подготовки бакалавров программа по дисциплине «Интерпретируемые языки программирования» предусматривает использование в учебном процессе следующих образовательных технологий: чтение лекций с использованием мультимедийных технологий; разбор конкретных ситуаций.

Компьютерные технологии позволяют проводить сравнительный анализ научных исследований по данной проблеме, являясь средством разнопланового отображения алгоритмов и демонстрационного материала.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и бакалаврами во время лекций и анализа результатов самостоятельной работы. Это обусловлено тем, что в процессе работы с языком программирования Python часто встречаются задачи, для которых единых подходов не существует. Каждая конкретная задача при своем исследовании имеет множество подходов, а это требует разбора и оценки целой совокупности конкретных ситуаций. Особенно этот подход широко

используется при определении адекватности математической модели, результатам компьютерных экспериментов.

Цель **лекции** – обзор понятий программирования на Python.

Цель **лабораторного занятия** – научить применять теоретические знания при решении и исследовании конкретных задач.

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

4.1 Фонд оценочных средств для проведения текущего контроля

Примерные вопросы для проведения текущего контроля.

1. Основные типы данных. Константы.
2. Выражения и операторы Python. Операции логического сдвига.
3. Условные операторы.
4. Операторы циклов.
5. Указатели и ссылки. Разыменование указателей.
6. Адресная арифметика.
7. Одномерные массивы.
8. Многомерные массивы.
9. Функции в Python.
10. Область действия имен.
11. Способы передачи параметров в функции.
12. Параметры командной строки.
13. Структуры. Массивы структур.
14. Указатели на структуры. Массивы структур с использованием указателей.
15. Объединения.
16. Макросы.
17. Функции ввода-вывода строк.
18. Форматированный ввод-вывод.
19. Файловый ввод-вывод. Функции ввода-вывода строк.
20. Файловый форматированный ввод-вывод.
21. Неформатированный ввод-вывод в нестандартные файловые потоки.
22. Определение классов в Python. Свойства и методы.
23. Работа с объектами через указатели.
24. Конструкторы и деструкторы.
25. Конструкторы с параметрами. Инициализация.
26. Конструкторы по умолчанию.
27. Статические данные и функции.
28. Массивы объектов.
29. Контейнеры. Векторы. Итераторы. Алгоритмы.
30. Дружественные функции и классы.
31. Наследование.
32. Перегрузка функций
33. Перегрузка операций.

Примерное содержание работ для проведения текущего контроля.

Самостоятельная работа студента заключается в самостоятельном выборе темы проекта-программы, его архитектуры и полное создание его с основными функциональными возможностями.

Учебная деятельность проходит в соответствии с графиком учебного процесса. Процесс самостоятельной работы контролируется во время аудиторных занятий и индивидуальных консультаций. Самостоятельная работа студентов проводится в форме изучения отдельных теоретических вопросов по предлагаемой литературе и решения индивидуальных задач повышенной сложности.

Фонд оценочных средств дисциплины состоит из средств текущего контроля (см. примерные варианты контрольных работ, индивидуальных заданий, задач и вопросов) и итоговой аттестации (зачета).

В качестве оценочных средств, используемых для текущего контроля успеваемости, предлагается перечень вопросов, которые прорабатываются в процессе освоения курса. Данный перечень охватывает все основные разделы курса, включая знания, получаемые во время самостоятельной работы.

Оценка успеваемости осуществляется по результатам: самостоятельного выполнения лабораторных работ, устного опроса при сдаче выполненных самостоятельных заданий, ответов на зачете.

Аттестация по учебной дисциплине проводится в виде зачета. Зачетный билет содержит два теоретических вопроса и одну задачу. Студент готовит ответы на билет в письменной форме в течение установленного времени. Далее зачет протекает в форме собеседования.

Примерное содержание самостоятельных работ

Самостоятельная работа студента заключается в самостоятельном выборе темы проекта-программы, его архитектуры и полное создание его с основными функциональными возможностями.

4.2 Фонд оценочных средств для проведения промежуточной аттестации

Перечень вопросов к зачету

1. Основные типы данных. Константы.
2. Выражения и операторы Python. Операции логического сдвига.
3. Условные операторы.
4. Операторы циклов.
5. Указатели и ссылки. Разыменование указателей.
6. Адресная арифметика.
7. Одномерные массивы.
8. Многомерные массивы.
9. Функции в Python.
10. Область действия имен.
11. Способы передачи параметров в функции.
12. Параметры командной строки.
13. Структуры. Массивы структур.
14. Указатели на структуры. Массивы структур с использованием указателей.
15. Объединения.
16. Макросы.
17. Функции ввода-вывода строк.
18. Форматированный ввод-вывод.
19. Файловый ввод-вывод. Функции ввода-вывода строк.
20. Файловый форматированный ввод-вывод.
21. Неформатированный ввод-вывод в нестандартные файловые потоки.
22. Определение классов в Python. Свойства и методы.
23. Работа с объектами через указатели.
24. Конструкторы и деструкторы.
25. Конструкторы с параметрами. Инициализация.

26. Конструкторы по умолчанию.
27. Статические данные и функции.
28. Массивы объектов.
29. Контейнеры. Векторы. Итераторы. Алгоритмы.
30. Дружественные функции и классы.
31. Наследование.
32. Перегрузка функций
33. Перегрузка операций.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

5.1 Основная литература:

1. Астапчук, В. А. Корпоративные информационные системы: требования при проектировании : учебное пособие для вузов / В. А. Астапчук, П. В. Терещенко. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2017. [Электронный ресурс] <https://www.biblio-online.ru/viewer/2B43246F-E60F-4B3C-9295-B4E4F872878B>, 05.10.2017.

2. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для прикладного бакалавриата / А. Ф. Тузовский. — М. : Издательство Юрайт, 2017. [Электронный ресурс] <https://www.biblio-online.ru/viewer/BDEEFB2D-532D-4306-829E-5869F6BDA5F9#/>

3. Казанский, А. А. Программирование на visual C# 2013 : учебное пособие для прикладного бакалавриата / А. А. Казанский. — Москва : Издательство Юрайт, 2019. — 191 с. — (Высшее образование). — ISBN 978-5-534-00592-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/434085> (дата обращения: 05.09.2019).

5.2 Дополнительная литература:

1. Трофимов, В. В. Алгоритмизация и программирование : учебник для академического бакалавриата / В. В. Трофимов, Т. А. Павловская ; под ред. В. В. Трофимова. — М. : Издательство Юрайт, 2017. [Электронный ресурс] <https://www.biblio-online.ru/viewer/B08DB966-3F96-4B5A-B030-E3CD9085CED4#/>, 05.10.2017.
2. Черпаков, И. В. Основы программирования : учебник и практикум для прикладного бакалавриата / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — (Серия : Бакалавр. Прикладной курс). — ISBN 978-5-9916-9983-9. [Электронный ресурс] <https://biblio-online.ru/book/7C1774D9-F5B5-4B45-85E1-BDE450DCC3E2>, 14.11.2017.
3. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для прикладного бакалавриата / А. Ф. Тузовский. — М. : Издательство Юрайт, 2017. [Электронный ресурс] <https://www.biblio-online.ru/viewer/BDEEFB2D-532D-4306-829E-5869F6BDA5F9#/>
4. Павловская, Татьяна Александровна. С/С++. Программирование на языке высокого уровня [Текст] : для магистров и бакалавров : учебник для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника" / Т. А. Павловская. - СПб. [и др.] : ПИТЕР, 2013. - 460 с.

5.3. Периодические издания:

1. "Алгебра и логика" / Институт математики им.Соболева СО РАН /Периодичность – 6 раз в год

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

№	Наименование электронного ресурса	Ссылка на электронный адрес
1.	Электронный каталог Научной библиотеки КубГУ	https://www.kubsu.ru/
2.	Электронная библиотечная система «BOOK.ru» ООО «КноРус медиа»	https://www.book.ru
3.	Электронная библиотечная система "Университетская библиотека ONLINE"	www.biblioclub.ru
4.	Электронная библиотечная система «ZNANIUM.COM» ООО «ЗНАНИУМ»	www.znanium.com
5.	Электронная библиотечная система издательства "Лань"	http://e.lanbook.com/
6.	Электронная библиотечная система "Юрайт"	http://www.biblio-online.ru

1. Российское образование, федеральный портал [Официальный сайт] — URL: <http://www.edu.ru>
2. Образовательный портал «Учеба» [Официальный сайт] URL: <http://www.ucheba.com/>
3. Портал «Российское образование» [Официальный сайт] URL: <http://www.edu.ru/>
4. Единое окно доступа к образовательным ресурсам «Единое окно» [Официальный сайт] URL: <http://window.edu.ru/>
5. Федеральная университетская компьютерная сеть России [Официальный сайт] URL: <http://www.runnet.ru/>
6. Служба тематических толковых словарей[Официальный сайт] URL: <http://www.glossary.ru/>
7. Образовательный портал [Официальный сайт] URL: «Академик» <http://dic.academic.ru/>

8. Web of Sciense (архив с 2002 года) рефераты [Официальный сайт] URL: <http://webofknowledge.com>.
9. Лекториум “(Минобрнауки РФ) единая Интернет-библиотека лекций [Официальный сайт] URL <http://www.lektorium.tv/>
10. Электронный архив документов КубГУ полнотекстов [Официальный сайт] URL: <http://docspace.kubsu.ru>

7. Методические указания для обучающихся по освоению дисциплины

Основной задачей промежуточной аттестации является повышение качества и прочности знаний обучающихся, приобретение и развитие навыков самостоятельной работы, укреплению обратной связи между преподавателем и обучающимся.

При самостоятельной работе рекомендуется изучить рекомендованные источники и усвоить полученную информацию. Необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Возможно использование литературы, подобранной самим обучающимся.

Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и информационно-телекоммуникационной сети «Интернет», является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у обучающихся свое отношение к конкретной проблеме.

Целесообразно также составление индивидуального терминологического словаря (глоссария) по теме вопросов, вынесенных на самостоятельное изучение, и словаря новых понятий, с которыми обучающийся впервые сталкивается в своей образовательной практике.

Для успешного освоения вопросов, вынесенных на самостоятельное изучение, необходимо законспектировать предложенные вопросы, проанализировать различные подходы на изложение предложенной проблемы. Возможно использование литературы, подобранной самим обучающимся.

Подготовка к практическим (лабораторным) занятиям.

Подготовку к каждому практическому занятию необходимо начать с ознакомления с планом практического занятия, который отражает содержание предложенной темы. Тщательное продумывание и изучение вопросов плана основывается на проработке текущего материала лекции, а затем изучения обязательной и дополнительной литературы, рекомендованной к данной теме. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса.

Подготовка к лабораторным занятиям и практикумам носит различный характер, как по содержанию, так и по сложности исполнения. Проведение прямых и косвенных измерений предполагает детальное знание измерительных приборов, их возможностей, умение вносить своевременные поправки для получения более точных результатов. Многие лабораторные занятия требуют большой исследовательской работы, изучения дополнительной научной литературы.

В процессе подготовки к практическим занятиям, необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. При всей полноте конспектирования лекции в ней невозможно изложить весь материал. Поэтому самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала.

Защита лабораторных работ должна происходить, как правило, в часы, отведенные на лабораторные занятия. Студент может быть допущен к следующей лабораторной работе только в том случае, если у него не защищено не более двух предыдущих работ.

Рекомендации по работе с литературой.

Работу с литературой целесообразно начать с изучения общих работ по теме, а также учебников и учебных пособий. Далее рекомендуется перейти к анализу монографий и статей, рассматривающих отдельные аспекты проблем, изучаемых в рамках курса, а также официальных материалов и неопубликованных документов (научно-исследовательские работы, диссертации), в которых могут содержаться основные вопросы изучаемой проблемы.

Работу с источниками надо начинать с ознакомительного чтения, т.е. просмотреть текст, выделяя его структурные единицы. При ознакомительном чтении закладками отмечаются те страницы, которые требуют более внимательного изучения.

В зависимости от результатов ознакомительного чтения выбирается дальнейший способ работы с источником. Если для разрешения поставленной задачи требуется изучение некоторых фрагментов текста, то используется метод выборочного чтения. Если в книге нет подробного оглавления, следует обратить внимание ученика на предметные и именные указатели.

Избранные фрагменты или весь текст (если он целиком имеет отношение к теме) требуют вдумчивого, неторопливого чтения с «мысленной проработкой» материала. Такое чтение предполагает выделение: 1) главного в тексте; 2) основных аргументов; 3) выводов. Особое внимание следует обратить на то, вытекает тезис из аргументов или нет.

Необходимо также проанализировать, какие из утверждений автора носят проблематичный, гипотетический характер, и уловить скрытые вопросы.

Понятно, что умение таким образом работать с текстом приходит далеко не сразу. Наилучший способ научиться выделять главное в тексте, улавливать проблематичный характер утверждений, давать оценку авторской позиции – это сравнительное чтение, в ходе которого Вы знакомитесь с различными мнениями по одному и тому же вопросу, сравниваете весомость и доказательность аргументов сторон и делаете вывод о наибольшей убедительности той или иной позиции.

Если в литературе встречаются разные точки зрения по тому или иному вопросу из-за сложности прошедших событий и правовых явлений, нельзя их отвергать, не разобравшись. При наличии расхождений между авторами необходимо найти рациональное зерно у каждого из них, что позволит глубже усвоить предмет изучения и более критично оценивать изучаемые вопросы. Знакомясь с особыми позициями авторов, нужно определять их схожие суждения, аргументы, выводы, а затем сравнивать их между собой и применять из них ту, которая более убедительна.

Следующим этапом работы с литературными источниками является создание конспектов, фиксирующих основные тезисы и аргументы..

Таким образом, при работе с источниками и литературой важно уметь:

- сопоставлять, сравнивать, классифицировать, группировать, систематизировать информацию в соответствии с определенной учебной задачей;
- обобщать полученную информацию, оценивать прослушанное и прочитанное;
- фиксировать основное содержание сообщений; формулировать, устно и письменно, основную идею сообщения; составлять план, формулировать тезисы;
- готовить и презентовать развернутые сообщения типа доклада;
- работать в разных режимах (индивидуально, в паре, в группе), взаимодействуя друг с другом;
- пользоваться реферативными и справочными материалами;
- контролировать свои действия и действия своих товарищей, объективно оценивать свои действия;

- обращаться за помощью, дополнительными разъяснениями к преподавателю, другим студентам;
- пользоваться лингвистической или контекстуальной догадкой, словарями различного характера, различного рода подсказками, опорами в тексте (ключевые слова, структура текста, предваряющая информация и др.);
- использовать при говорении и письме перифраз, синонимичные средства, слова-описания общих понятий, разъяснения, примеры, толкования, «словотворчество»;
- повторять или перефразировать реплику собеседника в подтверждении понимания его высказывания или вопроса;
- обратиться за помощью к собеседнику (уточнить вопрос, переспросить и др.);
- использовать мимику, жесты (вообще и в тех случаях, когда языковых средств не хватает для выражения тех или иных коммуникативных намерений).

Подготовка к промежуточной аттестации.

При подготовке к промежуточной аттестации целесообразно:

- внимательно изучить перечень вопросов и определить, в каких источниках находятся сведения, необходимые для ответа на них;
- внимательно прочитать рекомендованную литературу;
- составить краткие конспекты ответов (планы ответов).

При изучении курса «Язык программирование Python» необходимо активизировать остаточные знания студентов по таким дисциплинам как системное программное обеспечение, языки программирования и методы трансляции, программирование в ОС MS Windows.

При чтении лекционного курса представляется целесообразным обратить внимание на физические приложения излагаемых математических фактов. Чтобы изложение было понятным, следует акцентировать внимание не столько на формальных моментах доказательств, сколько на движущих ими идеях.

Необходимо отметить практическую значимость соответствующих проблем, обратить внимание на требования, предъявляемые к современному специалисту – прикладному математику, пояснить необходимость использования полученных знаний при изучении последующих специальных курсов.

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)

8.1 Перечень информационных технологий.

- Компьютерное тестирование по итогам изучения разделов дисциплины.
- Проверка домашних заданий и консультирование посредством электронной почты.
- Использование программного обеспечения при проведении лабораторных занятий.

8.2 Перечень необходимого программного обеспечения.

- ОС Windows XP, Архиватор WinRAR, Браузер Internet Explorer. Пакет программ Microsoft Office 2003, 2007, Oracle Database 11g Express Edition бесплатная для разработки, развертывания;

8.3 Перечень информационных справочных систем:

Не требуется

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине .

№	Наименование специальных помещений и помещений для самостоятельной работы	Номера аудиторий / кабинетов
---	---	------------------------------

1.	учебные аудитории для проведения занятий лекционного типа	501,502,503,505,506,507,508, 509, 510,513,514
2.	учебные аудитории для проведения занятий семинарского типа	501,502,503,505,506,507,508, 509, 510,513,514
3.	Компьютерные классы с выходом в Интернет	503,509,510
4.	учебные аудитории для выполнения научно – исследовательской работы (курсового проектирования)	Кабинет курсового проектирования (выполнения курсовых работ) - № 503 Оборудование: мультимедийный проектор, экран, персональные компьютеры, учебная мебель, доска учебная, выход в Интернет, учебно-наглядные пособия (тематические иллюстрации), принтер, презентации на электронном носителе, сплит-система
5.	учебные аудитории для самостоятельной работы, с рабочими местами, оснащенными компьютерной техникой с подключением к сети «Интернет» и обеспечением неограниченного доступа в электронную информационно-образовательную среду организации для каждого обучающегося, в соответствии с объемом изучаемых дисциплин	Кабинет для самостоятельной работы - № 504 Оборудование: персональные компьютеры, учебная мебель, доска учебная, выход в Интернет
6.	Исследовательские лаборатории (центров), оснащенные лабораторным оборудованием	Компьютерный класс № 510 : мультимедийный проектор, экран, персональные компьютеры, учебная мебель, доска учебная, выход в Интернет, наглядные пособия. Сетевое оборудование CISCO (маршрутизаторы, коммутаторы, 19-ти дюймовый сетевой шкаф) сплит-система, стенд «Архитектура ПЭВМ»
7.	Кабинет групповых и индивидуальных консультаций	№508 Оборудование: персональный компьютер, учебная мебель, доска учебная, учебно-наглядные пособия (тематические иллюстрации), сканер, доска магнитно-маркерная, стеллажи с учебной и периодической литературой
8.	Помещение для хранения и профилактического обслуживания учебного оборудования	Помещение № 511, Помещение № 516, Помещение № 517, Помещение № 518

9.	Помещение для проведения текущей и промежуточной аттестации	501,502,503,505,506,507,508, 509, 510,513,514
----	---	---

Согласно письма Министерства образования и науки РФ № МОН-25486 от 21.06.2017г «О разработке адаптированных образовательных программ» -Разработка адаптивной программы необходима в случае наличия в образовательной организации хотя бы одного обучающегося с ограниченными возможностями здоровья

Для обучающихся из числа инвалидов обучение проводится организацией с учетом особенностей их психофизического развития, их индивидуальных возможностей и состояния здоровья (далее - индивидуальные особенности).

При проведении обучения инвалидов обеспечивается соблюдение следующих общих требований:

- проведение обучения для инвалидов в одной аудитории совместно с обучающимися, не имеющими ограниченных возможностей здоровья, если это не создает трудностей для обучающихся;

- присутствие в аудитории ассистента (ассистентов), оказывающего обучающимся инвалидам необходимую техническую помощь с учетом их индивидуальных особенностей;

- пользование необходимыми обучающимся инвалидам техническими средствами с учетом их индивидуальных особенностей;

- обеспечение возможности беспрепятственного доступа обучающихся инвалидов в аудитории, туалетные и другие помещения, а также их пребывания в указанных помещениях;

В зависимости от индивидуальных особенностей обучающихся с ограниченными возможностями здоровья, организация обеспечивает выполнение следующих требований при проведении занятий:

а) для слепых:

- на компьютере со специализированным программным обеспечением для слепых, либо надиктовываются ассистенту;

б) для слабовидящих:

- задания и иные материалы оформляются увеличенным шрифтом;

- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;

- при необходимости обучающимся предоставляется увеличивающее устройство, допускается использование увеличивающих устройств, имеющихся у обучающихся;

в) для глухих и слабослышащих, с тяжелыми нарушениями речи:

- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающимся предоставляется звукоусиливающая аппаратура индивидуального пользования;

г) для лиц с нарушениями опорно-двигательного аппарата (тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания выполняются обучающимися на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;

Обучающийся инвалид при поступлении подает письменное заявление о необходимости создания для него специальных условий при проведении обучения с указанием особенностей его психофизического развития, индивидуальных возможностей и состояния здоровья (далее - индивидуальные особенности). К заявлению прилагаются документы, подтверждающие наличие у обучающегося индивидуальных особенностей (при отсутствии указанных документов в организации).