Федеральное государственное бюджетное образовательное учреждение высшего образования
«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»

ИНСТИТУТ ЭКОНОМИКИ, МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ
КАФЕДРА СИСТЕМНОГО АНАЛИЗА И ИНФОРМАТИКИ

УТВЕРЖДЕНА
решением кафедры Системного анализа и информатики
Протокол от «03» сентября 2020 г. № 01

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.ДВ.05.03 Углубленное программирование на Python
(индекс и наименование дисциплины (модуля), в соответствии с учебным планом)

09.03.03 Прикладная информатика _
(код и наименование направления подготовки (специальности))

"Веб-разработка"
направленность (профиль/специализация)

бакалавр
квалификация

очная форма обучения
форма(ы) обучения

Год набора - 2021

Москва, 2020 г.

Автор составитель:

Профессор кафедры Системного анализа и информатики
д-р техн. наук, профессор Ромашкова О.Н.

Заведующий кафедрой
Системного анализа и информатики Маруев С.А.
(*наименование кафедры)* *(ученая степень и(или) ученое звание)* *(подпись)* *(Ф.И.О.)*

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы…….....

2. Объем и место дисциплины в структуре образовательной программы…….....

3. Содержание и структура дисциплины …….

4. Материалы текущего контроля успеваемости обучающихся и фонд оценочных средств промежуточной аттестации по дисциплине …….

5. Методические указания для обучающихся по освоению дисциплины…….....

6. Учебная литература и ресурсы информационно-телекоммуникационной сети "Интернет", учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине……..…….

6.1. Основная литература……..……..……..……

6.2. Дополнительная литература……...……..……..

6.3. Учебно-методическое обеспечение самостоятельной работы……

6.4. Нормативные правовые документы…….……..……

6.5. Интернет-ресурсы…….……..……

6.6. Иные источники…….…….…….…

7. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы …..…….

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения программы

1.1. Дисциплина Б1.В.ДВ.05.03 «Углубленное программирование на PYTHON» обеспечивает овладение следующими компетенциями с учетом этапа:

1.1.

Код компетенции	Наименование компетенции	Код этапа освоения компетенции	Наименование этапа освоения компетенции
ПКр – 2	Способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение	ПКр-2.1 ПКр-2.2 ПКр-2.3	Знает принципы, методы и стандарты разработки архитектуры ИС Умеет профессионально выполнять разработку баз данных ИС Владеет навыками разработки архитектуры и баз данных ИС
УК ОС – 9	Способность использовать основы экономических знаний в различных сферах деятельности	УК ОС-9.1 УК ОС-9.2 УК ОС-9.3	Обосновывает собственную точку зрения по различным аспектам социально-экономической политики государства Обосновывает собственную точку зрения по различным аспектам социально-экономической политики государства Владеет навыками использования экономических знаний в различных сферах деятельности

1.2. В результате освоения дисциплины у обучаемых должны быть сформированы:

ОТФ/ТФ *(при наличии профстандарта)*/ трудовые или профессиональные действия	Код этапа освоения компетенции	Результаты обучения
06.015 СПЕЦИАЛИСТ ПО ИНФОРМАЦИОННЫМ СИСТЕМАМ С Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы С/14.6 Разработка архитектуры ИС С/17.6 Разработка баз данных ИС	ПКр-2.1 ПКр-2.2 ПКр-2.3	ПКр-2.1.1 Знает принципы, методы и стандарты разработки архитектуры ИС ПКр-2.1.2 Умеет применять принципы, методы и стандарты разработки архитектуры ИС ПКр-2.2.1 Знает основные методологии и средства разработки баз данных ИС ПКр-2.2.2 Умеет профессионально выполнять разработку баз данных ИС ПКр-2.3.1 Знает основные методологии и средства разработки архитектуры и баз данных ИС ПКр-2.3.2 Умеет профессионально выполнять разработку архитектуры и баз данных ИС
Основы экономических знаний	УК ОС-9	на уровне знаний: основ экономических знаний в различных сферах деятельности
		на уровне умений: Умеет обосновывать собственную точку зрения по различным аспектам социально-экономической политики государства
		на уровне навыков: использования основ экономических знаний в различных сферах деятельности

2. Объем и место дисциплины в структуре ОП ВО

Объем дисциплины

Объем дисциплины Б1.В.ДВ.05.03 «Углубленное программирование на PYTHON» в ЗЕ и академических/астрономических часах составляет 3 ЗЕ (108/81 ч). Дисциплина реализуется с применением дистанционных образовательных технологий.

Количество академических/астрономических часов по очной дистанционной форме обучения, выделенных на контактную работу 48/36 часов (в т.ч. лекц. – 16/12 ч., практ. – 32/24 ч.); на самостоятельную работу обучающихся - 24/18 часов.

Место дисциплины в структуре ОП ВО

В соответствии с учебным планом дисциплина Б1.В.ДВ.05.03 «Углубленное программирование на PYTHON» относится к вариативной части дисциплин по выбору учебного плана по направлению подготовки 09.03.03 Прикладная информатика. и изучается на 4 курсе в 7 семестре.
Дисциплины опирается на объём знаний информатики, вычислительных машин, сетей и систем телекоммуникаций, программирования;
Форма промежуточной аттестации для очной дистанционной формы обучения – экзамен.

3. Содержание и структура дисциплины

№ п/п	Наименование тем и/или разделов	Объем дисциплины, час.						Форма текущего контроля успеваемости**, промежуточной аттестации ***
		Всего	Контактная работа обучающихся с преподавателем по видам учебных занятий				СР	
			Л/ЭО, ДОТ*	ЛР/ ЭО, ДОТ*	ПЗ/ ЭО, ДОТ*	КСР		
Тема 1	Объектно-ориентированное программирование	22/16,5	4/3	-	10/7,5	-	8/6	О, К, Т, Р
Тема 2	Разработка приложений с графическим интерфейсом	22/16,5	4/3	-	10/7,5	-	8/6	Т, О
Тема 3	Клиент-серверное программирование в Python	28/21	8/6	-	12/9	-	8/6	Т, О, КР
Промежуточная аттестация		36/27						Э
Всего:		**108/81**	16/12	-	32/24		24/18	

Примечание:
** – при применении электронного обучения, дистанционных образовательных технологий в соответствии с учебным планом;*
*** – формы текущего контроля успеваемости: опрос (О), тестирование (Т), контрольная работа (КР), коллоквиум (К), реферат (Р).*
**** - разработчик указывает необходимые формы промежуточной аттестации экзамен (Э).*

Содержание дисциплины

Тема 1. Объектно-ориентированное программирование

Объектно-ориентированное проектирование. Классы и объекты. Свойства и операторная перегрузка. Наследование. Классы и типы. Итераторы, генераторы и декораторы. Метапрограммирование. Многократное использование кода. Работа с базами данных.

Тема 2. Разработка приложений с графическим интерфейсом
Проектирование шаблонов в Python. Структуры данных. Проектирование и разработка пользовательских интерфейсов с помощью пакетов tkinter. PyQT5 и Kivy. Полезные функции и модули.

Тема 3. Клиент-серверное программирование в Python
Сетевые функции. Регулярные выражения. Запись и передача данных. Работа с сетевым оборудованием. Работа с базами данных. Использование виртуальной машины. Реализация Git и GitHub. Работа с файлами. Шаблоны конфигураций с Jinja2, обработка вывода команд TextFSM. Разработка скриптов для автоматизации рабочих процессов.

4. Материалы текущего контроля успеваемости обучающихся и фонд оценочных средств промежуточной аттестации по дисциплине

4.1. Формы и методы текущего контроля успеваемости обучающихся и промежуточной аттестации

4.1.1. В ходе реализации дисциплины Б1.В.ДВ.05.02 «Углубленное программирование на PYTHON» используются следующие методы текущего контроля успеваемости обучающихся[1]:

Тема и/или раздел		Методы текущего контроля успеваемости в аудитории	Методы текущего контроля успеваемости с применением ДОТ
Тема 1	Объектно-ориентированное программирование	Опрос, коллоквиум	Тестирование, Реферат
Тема 2	Разработка приложений с графическим интерфейсом	Опрос	Тестирование
Тема 3	Клиент-серверное программирование в Python	Опрос	Тестирование, Контрольная работа

4.1.2. Экзамен проводится с применением следующих методов (средств):

Экзамен проводится с применением следующих методов (средств): без использования дистанционных образовательных технологий в форме выполнения практических заданий на компьютере с использованием инструментария веб-технологий, а именно LAMP/LEMP/FAMP/FEMP. HTML, CSS, JavaScript, TypeScript. PHP 7, MySQL. Symfony, Laravel, Zend Framework, Yii, Kohana. LESS, SASS, PostCSS. Gulp, Webpack, Babel. БЭМ. React, Vue, Angular. Git, Mercurial. Jenkins,

[1] *Разработчик программы самостоятельно выбирает форму заполнения пункта 1.1.*

Gitlab.

4.2. Материалы текущего контроля успеваемости обучающихся.

Преподаватель оценивает уровень подготовленности обучающихся к занятию по следующим показателям:

- опрос - устные ответы на вопросы преподавателя по теме занятия. Критерии оценивания устных ответов на вопросы преподавателя по теме занятия - правильность и полнота ответа;

- выступление с презентациями по реферату темы №1 отчетные доклады с презентациями по предлагаемым темам. Критериями оценивания реферата:

степень усвоения понятий и категорий по теме;
умение работать с документальными и литературными источниками;
грамотность изложения материала;
самостоятельность работы, наличие собственной обоснованной позиции;
Критерии оценивания презентации
оформлен титульный лист;
количество слайдов соответствует требованиям (мин.10 слайдов);
использование дополнительных эффектов в Power point, инфографики;
слайды представлены в логической последовательности.

- выполнение практических заданий на компьютере. Критерии оценивания выполнения заданий:

правильность выполнения задания;
оригинальность решений;
полнота выполнения задания.

- количество правильных ответов при тестировании. Критерии оценки выполненных студентами тестов определяются преподавателем самостоятельно. Рекомендуются следующие критерии оценки:

1. 85% – 100% правильных ответов – «отлично»;
2. 66% – 84% правильных ответов – «хорошо»;
3. 50% – 65% правильных ответов – «удовлетворительно»;
4. менее 50% правильных ответов – «неудовлетворительно».

- выполнение контрольной работы на компьютере. Критерии оценивания обучаемого в ходе выполнения:

3 балла – задание выполнено полностью, студент показал отличные знания, проявил активную позицию при ответах на вопросы, отлично владеет инструментарием

2 балла – задание выполнено с недочетами, студент показал хорошие знания, хорошо ответил на вопросы, владеет инструментарием;

1 балл – задание выполнено не полностью, студент показал слабые знания, не полностью ответил на вопросы, слабо владеет инструментарием;

0 баллов – задание не выполнено, студент не знает, не может ответить на вопросы, не владеет инструментарием.

Оценка знаний, умений, навыков проводится на основе балльно-рейтинговой системы 70% из 100% (70 баллов из 100) - вклад по результатам посещаемости занятий, активности на занятиях, выступления с докладами, выполнение всех практических заданий и контрольных работ, правильность ответов в ходе

тестирования и контрольных практических заданий.

Детализация баллов и критерии оценки текущего контроля успеваемости утверждается на заседании кафедры.

Примеры типовых оценочных материалов исходя из методов текущего контроля успеваемости

Типовые вопросы для проведения опроса

1. Встроенный тип str. Методы объекта str.
2. print() и форматирование вывода.
3. Работа с файловой системой средствами Python.
4. Работа с файлами. Методы open(), close(), read(), write().
5. Модуль re. Синтаксис регулярных выражений, метасимволы. Методы compile(), match(),search(),findall(),split(),sub(),subn(). Нумерованные и именованные группы в шаблонах поиска.
6. Unicode- строки, типы bytes и bytearray.
7. Встроенные типы последовательностей list, tuple, range и их методы.
8. Встроенный объект dict и его методы.
9. Встроенные типы чисел — int, float, complex. Машинное представление чисел с плавающей точкой и целых. Преобразование типов при сравнении чисел.
10. Рациональные числа. Модуль fractions.
11. Двоичное представление чисел. Неассоциативность операций в арифметике с плавающей запятой. Целые числа с произвольной точностью.
12. Множества. Встроенные типы set и frozenset.

Вварианты тестовых заданий

Как расшифровывается LEGB?
Local, Enclosed, Global, Built-in Light Edged Great Button Local Environment of Global Bench Line of Energy to Get Build
Что такое функция-декоратор?
Функция, которая оборачивает другую функцию для расширения ее функциональности Функция, которая служит для улучшения читаемости кода, но не влияет на его функционал явно Функция, которая задает внешний вид программы/сайта Функция из модуля decorator, которая нужна для подсветки синтаксиса
Что выведет следующий код?

```
pythonist.py    saved
1   squares = map(lambda x: x*x, [0,1,2,3,4])
2   print(list(squares))
3
```

IndexError [0, 1, 4, 9, 16] 0, 1, 4, 9, 16 0, 1, 2, 3 ,4
Сколько строк кода можно вписать в lambda-функцию
Бесконечно много Одну Две Лямбда-функция принимает только импортированные функции
Для чего служит функция reduce?
Для уменьшения значения всех элементов коллекции на переданное значение Для применения функции ко всем элементам последовательности Для комбинирования всех значений в наборе определенной функцией Для слияния всех значений в наборе в одно целое

Хорошо, а что тогда выведет этот код?

```
pythonist.py    saved
from functools import reduce
sum = reduce(lambda a, x: a + x, [0, 1, 2, 3, 4])
print(sum)
```

24 10 [0, 1, 2, 3, 4] 0

В какой области видимости находится переменная, объявленная в теле функции?

Local Enclosed Global Built-in

Окей, а где мы мы можем использовать переменную, если она объявлена в теле функции fun1, которая находится в теле другой функции fun2 и перед ее объявлением стоит волшебное слово nonlocal?

В любой части программы Только в fun1 Только в fun2 Только в fun1 и fun2

Что выведет этот код?

```
pythonist.py    saved
pythonist = True
def fun1():
    global pythonist
    pythonist = None

    def fun2():
        nonlocal pythonist
        pythonist = 'Python dev'

    fun2()

fun1()
print(pythonist)
```

True None Python SyntaxError

Для чего служит функция map?

Она используется в программировании географических карт Она объявляет карту сайта в Django Для применения функции к каждому элементу переданной последовательности Для вывода загруженности ПК процессом python

Что такое ORM?

Object-Relational Memory Object-Relational Market Object-Relational Mapping Object-Relational Model

Для чего используется ORM в Django?

ORM выполняет большинство взаимодействий с базой данных Для лучшей читабельности кода и взаимодействия с ним ORM отправляет запросы (request) и получает ответы (response) ORM нужен для разметки полей

Что такое QuerySet?

С помощью QuerySet можно отправлять запросы Это набор каких-то запросов Это СУБД Это список объектов заданной модели

Для чего используется QuerySet?

QuerySet позволяет читать данные из базы данных, фильтровать и изменять их порядок QuerySet позволяет создавать базы данных С помощью QuerySet можно редактировать содержимое базы данных Такого понятия не существует

Для чего используется Pickling?

Pickling обычно используется в качестве предвестника кеширования Возвращает длину списка результатов Для выполнения запросов к базе данных Аннотирует каждый объект в QuerySet

Что делает метод all()?
Возвращает копию текущего QuerySet Возвращает все элементы из QuerySet Удаляет все данные из QuerySet Копирует все данные из QuerySey
Что делает метод union()?
Использует оператор SQL UNION для разъединения результатов двух или более QuerySet'ов Использует оператор SQL UNION для удаления результатов двух или более QuerySet'ов Использует оператор SQL UNION для редактирования результатов двух или более QuerySet'ов Использует оператор SQL UNION для объединения результатов двух или более QuerySet'ов
Что делает метод intersection()?
Использует оператор SQL INTERSECT для возврата различных элементов двух или более QuerySet'ов Использует оператор SQL INTERSECT для удаления различных элементов двух или более QuerySet'ов Использует оператор SQL INTERSECT для возврата общих элементов двух или более QuerySet'ов Использует оператор SQL INTERSECT для удаления общих элементов двух или более QuerySet'ов
Что делает метод difference()?
Использует оператор SQL EXCEPT для хранения только элементов, отсутствующих в QuerySet Использует оператор SQL EXCEPT для добавления элементов, отсутствующих в QuerySet Использует оператор SQL EXCEPT для хранения только элементов, присутствующих в QuerySet, но не в каких-либо других QuerySet'ах Использует оператор SQL EXCEPT для удаления элементов, присутствующих в QuerySet
Что делает метод using()?
Этот метод предназначен для указания, какую базу данных QuerySet будет использовать, если вы используете более одной базы данных Этот метод предназначен для выбора базы данных, если вы используете более одной базы данных Этот метод возращает все используемые базы данных
Этот метод предназначен для указания, какую базу данных QuerySet НЕ будет использовать, если вы используете более одной базы данных

Варианты контрольных работ

1. Приведите примеры работы функций filter и reduce с итерируемым объектом.
2. У вас есть функция, которая принимает другую функцию в качестве аргумента. Как проверить значение аргумента?
3. Расскажите, как поменять порядок работы генератора.
4. Вам нужно реализовать функцию, которая будет использовать статическую переменную (например, счетчик вызовов). Вы не можете писать никакого кода вне этой функции, к тому же у вас нет никакой информации о внешних переменных (вне вашей функции). Как это сделать?
5. У вас есть функция, которая принимает аргумент, находящийся в заданных рамках [a, b]. Установите параметры реализации теста так, чтобы увеличить скорость тестирования. Варианты данного задания:

- аргумент находится в списке [a, b];
- a <= argument <= b;
- a in range(a, b + 1);

1. Какие методы и в каком порядке вызываются при выполнении данного выражения: print (A() + B()) ?
2. Каким будет результат выполнения данной функции?
```
def Foo():
    yield 42;
    return 666
```
1. Что будет в результате выполнения данного кода?

```
a = [[]]*3
a[1].append(1)
print(a)  # [[1], [1], [1]
```

1. Расположите данные функции в порядке их быстродействия. Обоснуйте ответ.

```
def f1(lIn):
    l1 = sorted(lIn)
    l2 = [i for i in l1 if i<0.5]
    return [i*i for i in l2]
def f2(lIn):
    l1 = [i for i in lIn if i<0.5]
    l2 = sorted(l1)
    return [i*i for i in l2]
def f3(lIn):
    l1 = [i*i for i in lIn]
    l2 = sorted(l1)
    return [i for i in l1 if i<(0.5*0.5)]
```

1. Напишите однострочный код для вычисления всех заглавных букв в файле. Ваш код должен работать, даже если файл слишком велик, чтобы целиком поместиться в памяти.
2. Каким будет результат? Почему? И есть ли здесь наследование?

```
class C:
    pass
type (C ())
type (C)
```

1. Какой результат мы получим, запустив следующий код?

```
big_num_1   = 1000
big_num_2   = 1000
small_num_1 = 1
small_num_2 = 1
big_num_1 is big_num_2
small_num_1 is small_num_2
```

1. Как такое возможно?

```
_MangledGlobal__mangled = 23

class MangledGlobal:
     def test(self):
         return __mangled

>>> MangledGlobal().test()
23
```

1. Какой будет результат?

```
print(_)
```

1. Вы видите только следующий кусок кода. Что с ним не так? И для чего такой код нужен?

```
if __debug__:
    assert False, ("error")
```

Вопросы и задания для самостоятельной подготовки

1. Каким будет результат выполнения данного кода?

```
list = ['a', 'b', 'c', 'd', 'e']
print list[10:] # []
```

1. Как мне перезагрузить уже импортированный модуль? (Предполагается, что модуль это отдельный файл).
2. Каким будет результат выполнения данного кода?
```
a = [[]]*3
a[1].append(1)
print(a)  # [[1], [1], [1]]
```
1. Что не так в этом коде?
```
def foo():
    from .module import *
    print(f"{bar()}")
```
1. Файл с программой на языке Python находится здесь: /usr/lib/python/person.py. Он запускается следующим образом: python /usr/lib/python/person.py. Что получится в результате? Код программы ниже:
```
class Person:
    def __init__(self, name):
        __name__ = name

    def getAge(self):
        print(__name__)

p = Person("John")
p.getAge()
```
1. Напишите декоратор timeit для измерения времени работы функции.
2. Напишите настраиваемый декоратор repeater, который будет перехватывать повторяющиеся ошибки.
3. Какой будет результат выполнения данного кода?
```
class parent:
    def __init__(self, param):
        self.v1 = param

class child:
    def __init__(self, param):
        self.v2 = param

obj = child(11)
print(obj.v1 + " " + obj.v2)
```
1. Добавьте код, чтобы программа ниже заработала.
```
class Repeater:
    ...
class RepeaterIterator:
    ...

repeater = Repeater("Hello")
for i in repeater:
    print(i)  # hello
```
1. Напишите программу для получения списка с уникальными элементами из списка с различными типами данных.
2. У нас есть следующий код с неизвестной функцией f(). Мы не хотим использовать в функции f() return, а хотим, чтобы она была генератором.
```
for x in f(5):
    print x
```

1. Результат выполнения данного кода следующий: 0 1 8 27 64. Напишите функцию f(), чтобы в результате ее работы получался такой результат.
2. Каким будет результат выполнения данного кода?
```
x = [[0], [1]]
print(len(' '.join(list(map(str, x)))))
```
1. Напишите программу, которая выводит на экран числа от 1 до 20. Но для чисел, кратных трем, вместо числа надо вывести “Fizz”, а для чисел, кратных пяти, — “Buzz”. Если число кратно и трем, и пяти, то нужно вывести на экран “FizzBuzz”.
2. Каким будет результат выполнения данного кода? Что здесь не так и как это можно исправить?
```
class Developer:
    coffee_cups = 0

    def __init__(self, name):
        self.name = name
        self.coffee_cups += 1

    def speak(self):
        print(f"I'm {self.name} and I've had {self.coffee_cups} cups
            of coffee today")

rover = Developer("Steve")
spot = Developer("Bob")
```

Примерные темы для написания рефератов

1. Включение интерфейсов веб-приложения в компонентную модель организации.
2. Как вы создадите словарь, который может сохранять порядок элементов (пар ключ-значение)?
3. Что такое контекстный менеджер в Python? Чем он отличается от конструкции try ... finally?
4. Какие функции нужно переопределить в классе A, чтобы экземпляры этого класса могли реализовать протокол контекстного менеджера?
5. Что такое синхронный код? А асинхронный? Как написать асинхронное приложение?
6. Что такое unittest в Python?
7. Что такое проверка типов? Какие есть типы в Python?
8. Как осуществляется управление памятью в Python?
9. Как в Python работает сборщик мусора? Опишите вкратце его деятельность.
10. Как можно расширить зону видимости глобальных переменных на другие модули?
11. Что такое атрибут __slots__?
12. Что такое метаклассы в Python?
13. Как создать класс без инструкции class?

Шкала оценивания

Уровень освоения компетенций по дисциплине «Углубленное программирование на PYTHON» определяется:

- знанием содержания процессов самоорганизации и самообразования, их особенностей и технологий реализации, исходя из целей совершенствования профессиональной деятельности;

- умением найти необходимую информацию, самостоятельно решать стандартные задачи профессиональной деятельности, выполнять действия в изученной последовательности, в том числе в новых условиях, на новом содержании;
- навыками использования современных информационных визуальных цифровых технологий и способами их реализации;
- способностью самостоятельно строить процесс овладения информацией, отобранной и структурированной для выполнения профессиональной деятельности;
- умением обеспечивать сохранность здоровья себе и сотрудникам при выполнении профессиональных задач.

Низкий «неудовлетворительно/неэкзамен» - компетенция не освоена или освоена в недостаточной мере. Студент не знает, либо знает на слабом уровне теоретический материал по дисциплине. Не владеет терминологией и основными понятиями из профессиональной сферы или называет неуверенно, с ошибками.

Пороговый (базовый) «удовлетворительно/экзамен» - компетенция освоена удовлетворительно, но достаточно. Студент освоил основную базу теоретических знаний. Владеет терминологией и основными понятиями из профессиональной сферы.

Продвинутый «хорошо/экзамен» - компетенция освоена достаточно хорошо. Студент знает теоретический материал по дисциплине, умеет применить эти знания на практике. Чётко и ясно формулирует свои мысли. Знает специальную и публицистическую литературу по профессиональным вопросам.

Высокий «отлично/экзамен» - компетенция освоена в полной мере или на продвинутом уровне. Студент знает теоретический материал, умеет применить эти знания на практике и имеет опыт в профессионально-практической деятельности. Приводит актуальные примеры из сферы профессиональной деятельности; демонстрирует способности к нестандартной интерпретации поставленного вопроса.

Наименование темы (раздела)	**Показатели оценивания**	**Критерии оценивания**	**Оценка (баллы)**
Объектно-ориентированное программирование	Вопрос на экзамене, Р	Экзамен	Удовлетворительно -41 -60, жорошо 61-80, отлично 81-100, Не зачтено – 40-0
Разработка приложений с графическим интерфейсом	Вопрос на экзамене	Экзамен	Удовлетворительно -41 -60, жорошо 61-80, отлично 81-100, Не зачтено – 40-0
Клиент-серверное программирование в Python	Вопрос на экзамене, КР	Экзамен	Удовлетворительно -41 -60, жорошо 61-80, отлично 81-100, Не зачтено – 40-0

В начале каждого семестра до студентов доводится информация о структуре набора баллов за семестр:

Максимальный балл за посещение и активную работу в семестре – 60 баллов.

Максимальный балл за каждую контрольную работу – 7 баллов.

Максимальный балл за каждую решенную задачу – 7 баллов.

Максимальный балл за реферат по выбранной теме – 12 баллов.

Вес текущих контрольных работ различен и зависит от этапов формирования компетенций. Решение и постановка прикладных задач, формирующих компетенции, выше, чем простое владение информационными технологиями.

В результате каждый студент четко представляет свое число набранных баллов.

Текущая аттестация обучаемых. Текущая аттестация обучаемых по дисциплине «Углубленное программирование на PYTHON» проводится в соответствии с Уставом Академии, Положением о текущей аттестации студентов по программам ВО и является обязательной.

Текущая аттестация по дисциплине «Углубленное программирование на PYTHON» проводится в форме опроса и контрольных мероприятий по оцениванию фактических результатов обучения студентов и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость практических занятий по аттестуемой дисциплине);

уровень овладения практическими умениями и навыками по всем видам учебной работы, уровень освоения компетенций;

результаты самостоятельной работы.

Активность обучаемого на занятиях оценивается по скорости и правильности выполнения практических заданий на компьютерах.

Кроме того, оценивание обучаемого проводится на контрольной неделе в соответствии с распоряжением проректора по учебной работе. Оценивание обучаемого на контрольной неделе проводится преподавателем независимо от наличия или отсутствия обучаемого (по уважительной или неуважительной причине) на занятии. Оценка носит комплексный характер и учитывает достижения обучаемого по основным компонентам учебного процесса за текущий период.

Оценивание обучаемого на занятиях осуществляется с использованием балльно-рейтинговой системы в соответствии с технологической картой дисциплины. Оценивание обучаемого на контрольной неделе также осуществляется по балльно-рейтинговой системе с выставлением оценок в ведомости и указанием количества пропущенных занятий.

4.3. Оценочные средства для промежуточной аттестации

4.3.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы. Показатели и критерии оценивания компетенций с учетом этапа их формирования

1.2.

Код компетенции	Наименование компетенции	Код этапа освоения компетенции	Наименование этапа освоения компетенции

ПКр – 2	Способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение	ПКр-2.1 ПКр-2.2 ПКр-2.3	Знает принципы, методы и стандарты разработки архитектуры ИС Умеет профессионально выполнять разработку баз данных ИС Владеет навыками разработки архитектуры и баз данных ИС
УК ОС – 9	Способность использовать основы экономических знаний в различных сферах деятельности	УК ОС-9.1 УК ОС-9.2 УК ОС-9.3	Обосновывает собственную точку зрения по различным аспектам социально-экономической политики государства Обосновывает собственную точку зрения по различным аспектам социально-экономической политики государства Владеет навыками использования экономических знаний в различных сферах деятельности

ОТФ/ТФ *(при наличии профстандарта)*/ трудовые или профессиональные действия	Код этапа освоения компетенции	Результаты обучения

06.015 СПЕЦИАЛИСТ ПО ИНФОРМАЦИОННЫМ СИСТЕМАМ С Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы С/14.6 Разработка архитектуры ИС С/17.6 Разработка баз данных ИС	ПКр-2.1 ПКр-2.2 ПКр-2.3	ПКр-2.1.1 Знает принципы, методы и стандарты разработки архитектуры ИС ПКр-2.1.2 Умеет применять принципы, методы и стандарты разработки архитектуры ИС ПКр-2.2.1 Знает основные методологии и средства разработки баз данных ИС ПКр-2.2.2 Умеет профессионально выполнять разработку баз данных ИС ПКр-2.3.1 Знает основные методологии и средства разработки архитектуры и баз данных ИС ПКр-2.3.2 Умеет профессионально выполнять разработку архитектуры и баз данных ИС
Основы экономических знаний	УК ОС-9	на уровне знаний: основ экономических знаний в различных сферах деятельности
		на уровне умений: Умеет обосновывать собственную точку зрения по различным аспектам социально-экономической политики государства
		на уровне навыков: использования основ экономических знаний в различных сферах деятельности

Промежуточная аттестация обучаемых по дисциплине «Углубленное программирование на PYTHON» проводится в соответствии с Уставом Академии, Положением о промежуточной аттестации студентов по программам ВО. Промежуточная аттестация по дисциплине «Углубленное программирование на PYTHON» проводится в соответствии с Учебным планом в 7 семестре в форме экзамена. Обучаемые допускаются к экзамену по дисциплине в случае выполнения ими учебного плана по дисциплине: выполнения всех заданий и мероприятий, предусмотренных программой дисциплины по формам текущего контроля и выполненных рефератах. В случае наличия учебной задолженности обучаемый отрабатывает пропущенные занятия в соответствии с требованиями преподавателя. Экзамен и зачет принимает ведущий преподаватель.

Оценка знаний обучаемого на экзамене складывается из:

ответа на один вопрос экзамена;

выполнения всех текущих контрольных работ;

защиты рефератов по третьей теме;

Оценка знаний обучаемого на экзамене носит комплексный характер, является балльной и определяется его:

– ответом на экзамене;
– учебными достижениями в семестровый период.

Знания, умения, навыки обучаемого на зачете оцениваются оценками: «отлично» – 5, «хорошо» – 4, «удовлетворительно» – 3, «неудовлетворительно» – 2. Кроме того, обучаемому выставляется оценка в соответствии с балльно-рейтинговой системой. Основой для определения оценки служит уровень усвоения обучаемыми материала, предусмотренного данной рабочей программой.

4.3.2 Типовые оценочные средства

Темы коллоквиумов по дисциплине «Углубленное программирование на PYTHON»

Коллоквиум организуется для проверки самостоятельной работы по теме, предложенную преподавателем (обучаемым). Коллоквиум позволяет автору научиться четко и грамотно формулировать мысли, структурировать информацию, использовать основные категории анализа, выделять причинно-следственные связи, иллюстрировать понятия соответствующими примерами, аргументировать свои выводы; овладеть научным стилем речи. Коллоквиум содержит: четкое изложение сути поставленной проблемы, включать самостоятельно проведенный анализ этой проблемы с использованием концепций и аналитического инструментария, рассматриваемого в рамках дисциплины, выводы, обобщающие авторскую позицию по поставленной проблеме.

1. Что такое Python?
2. Что такое lambda в Python?
3. Что означает pass в Python?
4. Что обозначают *args и **kwargs при определении функции в Pyython?
5. Что такое docstring в Python?
6. Какие в Pyhton существуют встроенные типы данных? Какие из них изменяемые, а какие неизменяемые?
7. В чем разница между списками и кортежами?
8. Какие ключевые слова могут быть использованы вместе с ключевым словом for?
9. Какие могут быть ключи в словарях Python?
10. В чем разница между globals(), locals(), и vars()?
11. Что такое PEP8?
12. Как работают срезы в Python?
13. Что значит отрицательный индекс в Python?
14. Что такое модуль __init__.py? Для чего он нужен?
15. Как можно поменять значения двух переменных в Python?
16. Как можно посмотреть все методы объекта?
17. Как можно получить документацию по методам объекта Python?
18. Что такое модуль в Python? Что такое пакет в Python? В чем разница между пакетом и модулем?
19. Можете ли вы написать многопоточное приложение в Python? В чем разница между мнгопоточностью (multithreading) и многопроцессностью (multiprocessing)?

Критерии оценки коллоквиумов:

Оценка «удовлетворительно» предполагает, что полученные результаты в значительной степени соответствуют поставленной цели (цель работы достигнута в основном). Обоснована актуальность работы. В процессе анализа литературы отобраны наиболее важные источники, продемонстрировано понимание решаемой проблемы. Выбраны адекватные цели научный подход, методы, процедуры. Они в значительной степени реализованы в работе. Выводы имеют наглядный и проверяемый характер. Требования по оформлению работы в основном выполнены.

Оценка «хорошо» ставится, когда полученные результаты преимущественносоответствуют поставленной цели и задачам. Обоснована практическая и теоретическая актуальность работы. В процессе анализа литературы отобран и проанализирован широкий круг теоретических и эмпирических источников. Выбраны и обоснованы применяемые научные подходы, методы и процедуры. Полученные результаты в целом логичны, доказательны и систематизированы. Оформление работы в целом соответствует существующим требованиям.

Оценка «отлично» предполагает: полученные результаты полностью соответствуют поставленной цели. Обоснована практическая и теоретическая значимость работы. Проведен детальный анализ теоретических и эмпирических источников, выводы автора самостоятельны и аргументированы. Выбраны и подробно описаны применяемые в работе научные подходы, методы и процедуры. Содержание работы полностью отражает узловые проблемы темы, исследовательская часть (в курсовой работе) выполнена самостоятельно, методологически корректно и содержит достоверные и интересные выводы и положения. Оформление работы полностью отвечает всем требованиям

Вопросы к экзамену по дисциплине «Углубленное программирование на PYTHON»

1. Инструкции и синтаксис. Составные конструкции и обработка исключений
2. Инструкции if/else/elif, логические операторы и выражения сравнения
3. Циклы while и for в Python
4. Функции в Python. Основные понятия
5. Области видимости и пространство имен в Python.
6. Передача аргументов в функцию. Специальные режимы сопоставления аргументов.
7. Парадигма объектно-ориентированного программирования. Поддержка в Python функционального программирования.
8. Объекты. Динамическая типизация. Инкапсуляция.
9. Генерация объекта class. Новое пространство имен. Объект экземпляр класса.
10. Атрибуты класса. Атрибуты данных. Атрибуты-методы. Параметр self. Добавление атрибутов к классу во время исполнения программы.
11. Специальные методы и атрибуты классов. Методы __init__() и ___del__() в Python. Декораторы функций и декораторы классов. Инструменты интроспекции в Python. Метаклассы.
12. Абстрактные методы в Python. Классические классы и классы нового стиля.
13. Наследование. Базовый и производный класс. Построение производного класса.
14. Порождающие функции (функции-фабрики). Множественное наследование. Примеси (Mix-in)
15. Агрегация. Контейнеры. Иерархия наследования.
16. Полиморфизм. Подмена методов в производном классе. Доступ к методам базового класса.

17. Обработка исключений. Инструкция try… exept… else… finaly. Объект Менеджер контекста и конструкция with…as. Классы встроенных исключений. Пользовательские исключения. Генерация заданного исключения с помощью Отладочные проверки assert и возбуждение исключения AssertionError.raise. Поиск ошибок программирования на стадиях разработки и тестирования.
18. Генераторы Python. Специальный тип функций – generator function. Объект-итератор. Итерация и потоки данных. (Data Flow Programming). Объект-функтор. Анонимная генератор-функция.
19. Асинхронный код. Event Loop. Сопрограммы. Событийно-ориентированное программирование без функций обратного вызова и кооперативная многозадачность, асинхронный ввод-вывод.
20. Устойчивость объектов. Время жизни объектов. Их сохранение для следующего запуска программы и/или передачи на другой компьютер. Модуль pickle для хранения представлений объектов в виде байтовых последовательностей и их последующего восстановления (сериализация и десериализация). Модуль shelve – БД для объектов. Независимая от языка программирования сериализация и XML. Создание документов XML и обработка готовых документов средствами Python
21. Работа в реальном времени с большими объемами XML данных (больше ОЗУ). Высокопроизводительная XML-библиотека lxml для Python, с поддержкой XPath, XSLT, метода iterparse (расширение ElementTree API), класса целевого анализатора. Обработка XML документа с ошибками. Парсер с параметром recovery.
22. Модули и пакеты. Библиотеки сторонних разработчиков (Django, Flask, Pygame, PyQt, twisted).
23. Модули и пакеты. Графический интерфейс.
24. Взаимодействие Python с Интернетом. Структура и функционирование сети Интернет. Архитектура клиент-сервер. Пакетная передача данных
25. Взаимодействие Python с Интернетом. Unicode и строки байтов. Разбор URL-адреса. Кодирование и декодирование строки запроса. Преобразование относительного URL-адреса в абсолютный
26. Разбор HTML-эквивалентов. Обмен данными по протоколу HTTP с помощью модуля urllib.request. Обработка cookies.
27. Обработка данных из форм. Определение кодировки.
28. Взаимодействие Python с базой данных SQLite.
29. Генерация HTML документа на запрос клиента.
30. Сжатие и распаковка данных. Работа с датой и временем.
31. Структурирование модулей в пакеты.
32. Импорт модулей и их составляющих из пакета.
33. Ссылки в пакетах.
34. Пакеты и файловая система.
35. Класс File.
36. Открытие файла.
37. Методы класса для File ввода-вывода.
38. Взаимодействие с файловой системой.
39. Модуль path.
40. Объекты и файловый ввод-вывод.
41. Объявление класса
42. Управление атрибутами и методами класса
43. Объявление объектов
44. Множественное наследование
45. Заимствование свойств и методов у родительского класса

Критерии оценки компетенций

Промежуточная аттестация обучаемых по дисциплине «Углубленное программирование на PYTHON» проводится в соответствии с Уставом Академии, Положением о промежуточной аттестации студентов по программам ВПО. Промежуточная аттестация по дисциплине «Углубленное программирование на PYTHON» проводится в соответствии с Учебным планом в 7 семестре в форме экзамена. Обучаемые допускаются к экзамену по дисциплине в случае выполнения ими учебного плана по дисциплине: выполнения всех заданий и мероприятий, предусмотренных программой дисциплины по формам текущего контроля и выполненных заданий. В случае наличия учебной задолженности обучаемый отрабатывает пропущенные занятия в соответствии с требованиями преподавателя. Оценка знаний обучаемого на зачете носит комплексный характер, является балльной и определяется его:

- ответом на экзамене;
- учебными достижениями в семестровый период.

Знания, умения, навыки студента на экзамене оцениваются оценками: «отлично» – 5, «хорошо» – 4 , «удовлетворительно» – 3, «неудовлетворительно» – 2. Кроме того, обучаемому выставляется оценка в соответствии с балльно-рейтинговой системой. Основой для определения оценки служит уровень усвоения обучаемыми материала, предусмотренного данной рабочей программой.

Темы рефератов по дисциплине «Углубленное программирование на PYTHON»

1. Тема 1 Язык Python. Типы данных, операции, операторы. Особенности ввода/вывода.
2. Тема 2 Встроенные типы объектов: Числа Строки Кортежи Списки Словари Множества
3. Тема 3 Файловый ввод-вывод. Чтение строк с помощью итераторов файлов. Работа с двоичными файлами.
4. Решение задач по теме. Решаются задачи с использованием внешних источников данных в виде текстовых и бинарных файлов. Рассматриваются операции чтения и записи в файлы.
5. Тема 4 Библиотека numpy для реализации математических объектов и вычислений.
6. Использование объектов array и matrix. Решение задач матричной алгебры.
7. Тема 5 Создание приложений с GUI. Обзор графических библиотек: Tkinter, PyQT.
8. Разработка графических приложений для работы с графами(визуализация графа, решение задач по поиску кратчайшего пути, МОД и др. Отображение результата на экране.
9. Тема 6 Классы в Python. Определение данных, методов, операций. Наследование.
10. Множественное наследование. Композиция при разработке классов.
11. Создайте абстрактный класс Functions. Класс должен выполнять различные операции над функциями. В число операций класса включить такие абстрактные методы, как вычисление интеграла, нахождение максимума, минимума, корней функции на заданном интервале, вычисление значения производной функции в заданной точке. В классе-наследнике использующем сервисы класса Fnctions, определить абстрактные методы Типовое задание

12. Разработать класс "Отрезок на прямой". Определить конструкторы, свойства, методы ввода/вывода, операции пересечения и объединения отрезков.

Критерии оценки рефератов:

Оценка «удовлетворительно» предполагает, что полученные результаты в значительной степени соответствуют поставленной цели (цель работы достигнута в основном). Обоснована актуальность работы. В процессе анализа литературы отобраны наиболее важные источники, продемонстрировано понимание решаемой проблемы. Выбраны адекватные цели научный подход, методы, процедуры. Они в значительной степени реализованы в работе. Выводы имеют наглядный и проверяемый характер. Требования по оформлению работы в основном выполнены.

Оценка «хорошо» ставится, когда полученные результаты преимущественносоответствуют поставленной цели и задачам. Обоснована практическая и теоретическая актуальность работы. В процессе анализа литературы отобран и проанализирован широкий круг теоретических и эмпирических источников. Выбраны и обоснованы применяемые научные подходы, методы и процедуры. Полученные результаты в целом логичны, доказательны и систематизированы. Оформление работы в целом соответствует существующим требованиям.

Оценка «отлично» предполагает: полученные результаты полностью соответствуют поставленной цели. Обоснована практическая и теоретическая значимость работы. Проведен детальный анализ теоретических и эмпирических источников, выводы автора самостоятельны и аргументированы. Выбраны и подробно описаны применяемые в работе научные подходы, методы и процедуры. Содержание работы полностью отражает узловые проблемы темы, исследовательская часть (в курсовой работе) выполнена самостоятельно, методологически корректно и содержит достоверные и интересные выводы и положения. Оформление работы полностью отвечает всем требованиям

Шкала оценивания по результатам зачета с оценкой:

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения			
	неудовлетворительно	удовлетворительно	хорошо	отлично
ПКр – 2. Способен выполнять работы и управлять работами по созданию (модификации) и сопровождению информационных систем, автоматизирующих задачи организационного управления и бизнес-процессы				
на уровне знаний: принципов разработки планов коммуникаций с заказчиком в проектах	Не знает как проводить описание прикладных процессов и информационного обеспечения	Демонстрирует частичные знания в описании прикладных процессов и информационного обеспечения	Демонстрирует системные знания в описании прикладных процессов и информационного обеспечения	Проявляет высокий уровень знаний всего учебного материала

создания (модификации) ИС; Знает принципы, методы и стандарты разработки архитектуры ИС; Знает основные методологии и средства проектирования и дизайна ИС; Знает основные методологии и средства разработки баз данных ИС; Знает правила и стандарты разработки пользовательской документации к ИС; Знает принципы и приемы командообразования.	решения разработки планов коммуникаций с заказчиком в проектах создания (модификации) ИС; методологии и средства проектирования и дизайна ИС; правила и стандарты разработки пользовательской документации к ИС; принципы и приемы командообразования.	решения разработки планов коммуникаций с заказчиком в проектах создания (модификации) ИС; методологии и средства проектирования и дизайна ИС; правила и стандарты разработки пользовательской документации к ИС; принципы и приемы командообразования.	решения разработки планов коммуникаций с заказчиком в проектах создания (модификации) ИС; методологии и средства проектирования и дизайна ИС; правила и стандарты разработки пользовательской документации к ИС; принципы и приемы командообразования.	
на уровне умений: Умеет выполнять документирование существующих бизнес-процессов организации заказчика; Умеет применять принципы, методы и стандарты разработки	Не умеет применять знания на практике	Демонстрирует частичный уровень умений	Демонстрирует достаточный уровень умений	Демонстрирует высокий уровень умений для разработки алгоритмов решения задач

архитектуры ИС; Умеет профессионально выполнять разработку баз данных ИС; Умеет организовывать программы обучения и развития персонала.				
на уровне навыков: Владеет навыками выстраивания коммуникаций с заказчиком в проектах создания (модификации) и ввода ИС в эксплуатацию; Владеет навыками документирования и реинжиниринга бизнес-процессов организации; Владеет навыками проектирования и разработки дизайна ИС; Владеет навыками создания пользовательской документации к ИС; Владеет навыками организации	Не владеет практическими навыками работы с технологиями и программным инструментарие м	Владеет практическими навыками частично работы с технологиями и программным инструментарием	Владеет практическими навыками формирования требований к информационной системе достаточно свободно	Владеет практическим и навыками работы на компьютере и использования стандартных информационно-коммуникационных технологий на высоком уровне

заключения договоров на выполняемые работы, связанные с ИС				
УК ОС - 9. Способен использовать основы экономических знаний в различных сферах деятельности				
Обосновывает собственную точку зрения по различным аспектам социально-экономической политики государства	Не умеет применять знания на практике	Демонстрирует частичный уровень умений	Демонстрирует достаточный уровень умений	Демонстрирует высокий уровень умений для разработки алгоритмов решения задач

4.4. Методические материалы

4.4.1. Методические материалы, определяющие процедуру оценивания ответов обучающихся на вопросы на понимание лекционного материала

Критериями оценки ответа обучающихся на лекционном занятии выступают:

- правильность ответов на вопросы преподавателя по изученному
- материалу;
- полнота и лаконичность ответа;
- степень понимания тематики предмета;
- логика и аргументированность изложения материала;
- приведение примеров, демонстрирующих умение и владение
- полученными знаниями по темам предмета в раскрытии поставленных вопросов.

4.4.2. Методические материалы, определяющие процедуру оценивания при проведении опроса на практическом занятии

Оценки "отлично" заслуживает обучающийся, обнаруживший всестороннее, систематическое и глубокое знание вопроса, умение свободно ориентироваться в теме, усвоивший основную, и знакомый с дополнительной литературой, рекомендованной программой. Как правило, оценка "отлично" выставляется обучающимся, усвоившим взаимосвязь основных понятий в их значении для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала;

Оценки "хорошо" заслуживает обучающийся, обнаруживший полное знание темы, успешно выполняющий предусмотренные программой задания, усвоивший основную литературу, рекомендованную в программе. Как правило, оценка "хорошо" выставляется обучающимся, показавшим систематический характер знаний по пройденному материалу и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебы и профессиональной деятельности;

Оценки "удовлетворительно" заслуживает обучающийся, обнаруживший знание основного материала в объеме, необходимом для дальнейшего усвоения материала

и предстоящей работы по профессии, знакомый с основной литературой, рекомендованной программой.
Оценка "неудовлетворительно" выставляется обучающемуся, обнаружившему пробелы в знаниях основного материала темы, допустившему принципиальные ошибки в понимании и изложении учебного материала.

4.4.3. Методические материалы, определяющие процедуру оценивания промежуточной аттестации по дисциплине

Экзамен проводится путем распределения по времени в течение всего срока обучения проведения мониторинга уровня освоения компетенции по результатам выполнения заданий по подготовке к опросу. При условии положительных ответов на вопросы экзамена, компетенции считаются сформированными, и по данной дисциплине выставляется оценка. В ином случае, компетенции считаются несформированными, и по данной дисциплине выставляется оценка «неудовлетворительно».
Экзамен принимается в устной форме, по вопросам. Задание включает теоретический и практический вопрос. Оценка знаний обучающегося на экзамене носит комплексный характер и определяется его:

- ответом на экзамене;
- учебными достижениями в семестровый период.

Знания, умения, навыки обучающегося на экзамене оцениваются оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Основой для определения оценки служит уровень усвоения обучающимися материала, предусмотренного данной рабочей программой.

Оценивание студента на зачете с оценкой по дисциплине «Углубленное программирование на PYTHON»

Оценка	Требования к знаниям
Отлично	Оценка «отлично» выставляется обучающемуся, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает на экзамене, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение. Учебные достижения в семестровый период демонстрируют высокую степень овладения программным материалом.
Хорошо	Оценка «хорошо» выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения. Учебные достижения в семестровый период демонстрируют хорошую степень овладения программным материалом.
Удовлетворительно	Оценка «удовлетворительно» выставляется обучающемуся, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.

	Учебные достижения в семестровый период демонстрируют достаточную (удовлетворительную) степень овладения программным материалом.
Неудовлетворительно	Оценка «неудовлетворительно» выставляется обучающемуся, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится обучающимся, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине. Учебные достижения в семестровый период не демонстрировали достаточную степень овладения программным материалом на пороговом уровне.

Экзаменпроводится во время экзаменационной сессии по традиционной шкале оценивания.
Студентам, не выполнившим требования по промежуточной аттестации, по уважительным причинам, предоставляется возможность их выполнения и сдачи.

5. Методические указания для обучающихся по освоению дисциплины

Домашние задания соответствуют приведенным ранее темам для коллоквиумов и опросов при самостоятельной подготовке, и могут быть опубликованы:
1. На общих ресурсах академии (в сетевой папке);
2. Высланы на общую почту группы
Конкретный способ коммуникации со студентами определяется преподавателем. Для подготовки к практическим занятиям необходимо ознакомиться с лекциями, рекомендованной литературой, иными источниками, интернет ресурсами, повторить материал предыдущих практических занятий. Самостоятельно разобрать вопросы к практическим занятиям по темам. При возникновении вопросов - обратиться к преподавателю по электронной почте с указанием конкретной проблемы и (или) прийти к преподавателю на консультацию в установленное время.

5.1. Методические указания по вопросам на понимание лекционного материала

На лекциях рекомендуется слушать предлагаемый лектором материал, при этом параллельно конспектировать основные положения, поскольку это дает наибольший результат в усвоении материала. Предоставляется возможность задавать вопросы на уточнение понимания темы и принимать участие в ее обсуждении. Кроме этого, для лучшего освоения материала и систематизации знаний по дисциплине, необходимо постоянно разбирать материалы лекций по конспектам и учебным пособиям. Во время самостоятельной проработки лекционного материала особое внимание следует уделять возникшим вопросам, непонятным терминам, спорным точкам зрения. Все такие моменты следует выделить или выписать отдельно для дальнейшего обсуждения на семинарском занятии. В случае необходимости обращаться к преподавателю за консультацией. Полный список литературы по дисциплине приведен в разделе 6 программы.

5.2. Методические указания по подготовке вопросов для самостоятельного изучения к занятиям практического (семинарского) типов

Подготовка обучающегося к практическому занятию осуществляется на основании плана раскрытия темы практического занятия, которое разрабатывается преподавателем на основе рабочей программы и своевременно доводится до сведения обучающегося.

При подготовке к практическому занятию, обучающемуся необходимо внимательно изучить основные вопросы темы семинара. Важным условием успешной подготовки к практическому занятию является четкая организация самостоятельной работы студентов по изучению учебной и дополнительной литературы. Умение анализировать и применять для ответов на вопросы и решения задач и заданий полученные знания при самостоятельной подготовке в значительной степени определяет успешность освоения материала по дисциплине и формирование у обучающихся соответствующих компетенций.

Подготовка вопросов для самостоятельного изучения включает: изучение необходимой литературы (обязательной, дополнительной литературы, специальных периодических изданий, Интернет-ресурсов), подготовку конспекта ответа, ответы на вопросы.

При подготовке к практическим занятиям важно:

- использовать достаточно широкий диапазон массива информации, провести обзор литературы и специальных изданий, составить каталог Интернет-ресурсов;
- представить различные подходы, четко и полно определить рассматриваемые понятия, выявить взаимосвязи понятий и явлений, взаимозависимости и связи с другими вопросами;
- грамотно структурировать материал, ясно, четко и логично его излагать, приводить соответствующие примеры из практики, для иллюстрации положений, тезисов и выводов использовать таблицы, схемы, графики, диаграммы.

Вопросы для самостоятельной подготовки к занятиям практического (семинарского) типа указаны в разделе 4.2.

5.3. Методические рекомендации по подготовке к экзамену по дисциплине

Ответ на экзамене предусматривает устный ответ на теоретические вопросы. При подготовке к экзамену обучающийся обращается к пройденному материалу, сосредоточенному в конспектах лекций, учебниках и других источниках информации.

Повторяя, обобщая, закрепляя и дополняя полученные знания, поднимает их на качественно-новый уровень — уровень системы совокупных данных, что позволяет ему понять логику всего предмета в целом. Новые знания обучающийся получает в ходе самостоятельного изучения того, что не было изложено в лекциях и на семинарских занятиях.

Экзаменкак особая форма учебного процесса имеет свои особенности, специфические черты и некоторые аспекты, которые необходимо обучающемуся знать и учитывать в своей работе. Это, прежде всего:

• что и как запоминать при подготовке к зачету;
• по каким источникам и как готовиться;
• на чем сосредоточить основное внимание;
• каким образом в максимальной степени использовать программу курса;
• что и как записать, а что выучить дословно и т. п.

На экзамене, как правило, проверяется не столько уровень запоминания обучающимся учебного материала, сколько то, насколько успешно он оперирует теми или иными научными понятиями и категориями, систематизирует факты, как умеет мыслить, аргументировано отстаивать определенную позицию, объясняет и пересказывает заученную информацию.

Программу курса необходимо максимально использовать как в ходе подготовки, так и на самом экзамене. Ведь она включает в себя разделы, темы и основные проблемы, в рамках которых и формируются вопросы для экзамена.

Оптимальным для подготовки к экзамену является вариант, когда обучающийся начинает подготовку к нему с первых занятий по данному курсу.
При подготовке к экзамену по наиболее сложным вопросам, ключевым проблемам и важнейшим понятиям необходимо сделать краткие письменные записи в виде тезисов, планов, определений. Особое внимание в ходе подготовки к экзамену следует уделять конспектам лекций, ибо они обладают рядом преимуществ по сравнению с печатной продукцией. Как правило, они более детальные, иллюстрированные, что позволяет оценивать современную ситуацию, отражать самую свежую научную и оперативную информацию, отвечать на вопросы, интересующие аудиторию, в данный момент, тогда как при написании и опубликовании печатной продукции проходит определенное время, и материал быстро устаревает.
В то же время, подготовка по одним конспектам лекций недостаточна, необходимо использовать и иную учебную литературу. Не следует бояться дополнительных и уточняющих вопросов на экзамене. Они, как правило, задаются или помимо экзаменационного вопроса для выявления общей подготовленности, или в рамках билета для уточнения высказанной мысли.

6. Учебная литература и ресурсы информационно-телекоммуникационной сети "Интернет", включая перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Основная литература

1. Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2018. – 1280 с.
2. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2019. – 284 с.
3. Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2018. – 992 с.
4. Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2018. – 992 с.
5. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.
6. Лучано Рамальо Python. К вершинам мастерства. – М.: ДМК Пресс, 2020. – 768 с.

6.2. Дополнительная литература

7. Свейгарт, Эл. Автоматизация рутиных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, 2016. – 592 с.
8. Рейтц К., Шлюссер Т. Автостопом по Python. – СПб.: Питер, 2017. – 336 с.: ил. – (Серия «Бестселлеры O'Reilly»).
9. Любанович Билл Простой Python. Современный стиль программирования. – СПб.: Питер, 2016. – 480 с.: – (Серия «Бестсепперы O'Reilly»).
10. Федоров, Д.Ю. Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс). – ISBN 978-5-534-10971-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: https://urait.ru/bcode/437489 (дата обращения: 13.02.2020).

6.3. Учебно-методическое обеспечение самостоятельной работы

11. Положение об организации самостоятельной работы студентов федерального государственного бюджетного образовательного учреждения высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации» (в ред. приказа РАНХиГС от 11.05.2016 г. № 01-2211). http://www.ranepa.ru/images/docs/prikazy-ranhigs/Pologenie_o_samostoyatelnoi_rabote.pdf

Самостоятельная работа студентом осуществляется для закрепления изученного материала после практических занятий или лабораторных работ, для выполнения домашних заданий, для подготовки к контрольным работам, для изучения дополнительных материалов.

№ пп	Тип занятия	Указания
	Тема 1	Объектно-ориентированное программирование
1	СРС	Посмотреть в интернете понятие и особенности информации, данных, знаний; клиент-серверных технологий, определить, в чем различие этих понятий. Обратить внимание на протокол HTTP.
2	СРС	Выяснить роль технологий веб в системах организационно-экономического управления. Рассмотреть понятие Основы JavaScripta. Изучить уровни представления информационных объектов веб при описании предметной области.
3	СРС	Для подготовки к занятиям использовать лекции, интернет ресурсы, основную и дополнительную литературу.
	Тема 2	Разработка приложений с графическим интерфейсом
1	СРС	Изучить аспекты рассмотрения информационных процессов веб в системах управления: преобразование и движение. Процесс принятия решения как основной элемент преобразования информации. Знать основные фазы процесса принятия решений как информационного процесса. Подробно разобрать процессы передачи информации. Знать определение и свойства потока информации веб.
2	СРС	Знать уровни представления информационных процессов. Подробно изучить концептуальную модель информационного процесса. Разобрать структуры логического и физического уровней представления информационного процесса веб.
3	СРС	Для подготовки к занятиям использовать лекции, интернет ресурсы, основную и дополнительную литературу.
	Тема 3	Клиент-серверное программирование в Python
1	СРС	Разобрать понятие инструментария веб. Изучить место и роль интернет-приложений в общем контуре организационно-экономического управления. Подробно разобрать этапы развития веб. Выделить основные процессы, обеспечивающие работу веб приложений. Изучить основные свойства сайта. Знать преимущества внедрения веб в сферу управления и бизнеса.

2	СРС	Разобрать понятие Основы JavaScripta. Выделить и изучить подходы к разработке веб сайтов.
3	СРС	Для подготовки к занятиям использовать лекции, интернет ресурсы, основную и дополнительную литературу

Рекомендации по организации самостоятельной работы студентов:

Вид учебных занятий	Организация деятельности студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удается разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на практическом занятии.
Практические занятия	Проработка рабочей программы, уделяя особое внимание целям и задачам, структуре и содержанию дисциплины. Анализ литературы и/или Интернет – ресурсов, решение типовых задач, проведение консультаций с преподавателем по проблемным вопросам лекционного материала, подготовка к опросу и/или коллоквиуму, постановка задачи для практического задания, оформление отчета по заданию, работа в малой группе.
Контрольная работа / индивидуальное задание	Ознакомление с постановкой задачи, разработка алгоритма решения, выбор программно-аппаратных средств реализации. Реализация в выбранной инструментальной среде. Подготовка выводов и/или короткого отчета в виде презентации.
Подготовка к дискуссиям, докладам, написанию реферата	Поиск литературы и составление библиографии, использование от 3 до 5 научных работ, изложение мнения авторов и своего суждения по выбранному вопросу; изложение основных аспектов проблемы. Изучение научной, учебной, нормативной и другой литературы. Отбор необходимого материала; формирование выводов и разработка конкретных алгоритмов и/или рекомендаций по решению поставленной цели и задачи; проведение практических исследований по данной теме. Разработка презентации для доклада основных результатов проведенной работы. Желательно оформление результатов в виде статьи в сборник трудов конференции или тематический научный журнал.

6.4. Нормативные правовые документы

12. IEEE P1003.0 «Руководство по окружению открытых систем POSIX».

13. ГОСТ Р ИСО/МЭК 12207. Процессы жизненного цикла программных средств.

6.5. Интернет-ресурсы, справочные системы.

14. http://smarty.php.net – веб-технологии
15. http://yandex.ru/cgi/vesna - примеры скриптов веб-приложений
16. http://belitsoft.ru/ - инструментарий разработки интернет-приложений
17. http://www.virtech.ru – инструментарий разработки интернет-приложений
18. http://www.oracle.sun.com – инструментарий разработки интернет-приложений
19. 1. http://www.consultant.ru/ - Официальный сайт компании «Консультант Плюс».
20. http://www.cfin.ru/software/index.shtml - Сайт «Корпоративный менеджмент». Обзор информационных систем для основных функций управления и бизнеса.
21. http://citforum.ru/ - «Сервер информационных технологий» - on-line библиотека информационных материалов по компьютерным технологиям.
22. http://www.intuit.ru/ - Образовательный портал дистанционного обучения.
23. www.coursera.org - Платформа для бесплатных онлайн - лекций (проект по публикации образовательных материалов в интернете, в виде набора бесплатных онлайн - курсов).

6.6. Иные источники

24. Шелудько, В. М. Основы программирования на языке высокого уровня Python: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 146 с. – ISBN 978-5-9275-2649-9. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: http://www.iprbookshop.ru/87461.html (дата обращения: 13.02.2020). – Режим доступа: для авторизир. пользователей
25. Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 107 с. – ISBN 978-5-9275-2648-2. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: http://www.iprbookshop.ru/87530.html (дата обращения: 13.02.2020). – Режим доступа: для авторизир. пользователей
26. Доусон М. Программируем на Python. – СПб.: Питер, 2014. – 416 с.
27. Прохоренок Н.А. Python 3 и PyQt. Разработка приложений. – СПб.: БХВ-Петербург, 2012. – 704 с.
28. Прохоренок Н.А. Самое необходимое. — СПб.: БХВ-Петербург, 2011. — 416 с.

7. Материально-техническое и программное обеспечение дисциплины (модуля)

Название лаборатории/класса,	Наименование оборудования	Перечень лицензионного

оснащенного необходимым, в соответствии с требованиями ФГОС/ СУОС, оборудованием		программного обеспечения
учебная аудитория для проведения занятий лекционного типа, текущего контроля и промежуточной аттестации	Рабочие места студентов: столы и стулья – соответственно количеству студентов. Рабочее место для инвалида и лиц с ОВЗ: парта с телескопической столешницей на электромеханическом приводе - 1 шт., кресло-коляска для инвалидов 18" - 1 шт., индукционная петля - 1 шт., компьютер с версией для слабовидящих - 1 шт., кнопка вызова сотрудников – 1 шт. Рабочее место преподавателя: стол – 1 шт., стул – 1 шт, кафедра - 1 шт. Доска меловая и маркерная. Экран, ноутбук Lenovo ideapad 100/15, проектор	Мультимидийный проектор Консультант Плюс
информационно – аналитическая лаборатория – учебная аудитория для проведения занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальны х консультаций	Рабочие места: столы компьютерные – в соотвествии с количеством студентов, кресло Престиж Profi -B-20 Самба бордо в рубчик - 15 шт., подставка для ног Fellowes FS-48121 Standard черный - 15 шт. Рабочее место преподавателя: стол компьютерный - 1 шт., стул - 1 шт. Доска меловая или маркерная Персональные компьютеры	Мультимидийный проектор Консультант Плюс
библиотека - помещение для самостоятельной работы	Рабочие места: столы и стулья. Рабочее место преподавателя: стол – 1 шт., стул - 1 шт., кафедра библиотечная - 1 шт. Телефон – 1 шт., цифровой многофункциональный копир - 1 шт., копировальный аппарат МФУ – 1 шт., принтер - 1 шт., сканер – 1 шт. Шкаф – 7 шт, стеллаж-33 шт, библиотечная стойка – 2 шт., стенд – 2 шт. Меловая или маркерная доска. Персональные компьютеры	Мультимидийный проектор Консультант Плюс

Программное обеспечение:

В процессе лекционных и семинарских занятий используется следующее программное обеспечение:

- программы, обеспечивающие навигацию в сети Интернет: «Google chrome»;
- программы, демонстрации видео материалов: проигрыватель «Windows Media»;
- программы для демонстрации и создания презентаций: «Microsoft Power Point».

Информационные справочные системы:

1. www.nnir.ru / - Российская национальная библиотека
2. www.nns.ru / -Национальная электронная библиотека
3. www.rsi.ru / - Российская государственная библиотека
4. www.biznes-karta.ru / -Агентство деловой информации «Бизнес-карта»
5. www. rbs.ru / - Информационное агентство «РосБизнесКонсалтинг»
6. www.aport.ru / - Поисковая система
7. www.rambler.ru / - Поисковая система
8. www.yandex.ru / - Поисковая система
9. www.busineslearning.ru / - Система дистанционного бизнес образования
10. www.test.specialist.ru / - Центр компьютерного обучения МГТУ им. Н. Э. Баумана
11. http://www.consultant.ru/ - Консультант плюс
12. http://www.garant.ru/ - Гарант
13. Научная библиотека РАНХиГС. URL: http://lib.ranepa.ru/;
14. Научная электронная библиотека eLibrary.ru. URL: http://elibrary.ru/defaultx.asp;
15. Национальная электронная библиотека. URL: www.nns.ru; Российская государственная библиотека. URL: www.rsl.ru;
16. Российская национальная библиотека. URL: www.nnir.ru; Электронная библиотека Grebennikon. URL: http://grebennikon.ru/;
17. Электронно-библиотечная система Издательства «Лань». URL: http://e.lanbook.com;
18. Электронно-библиотечная система ЮРАЙТ. URL: http://www.biblio-online.ru/.