



Наименование дисциплины и код: «Фрактальный анализ рыночных данных на языке Python» Б.3.2.20 гр.БИ-1-15

Лектор	Ногоева Г.Д., Шекеев К.
Контактная информация:	Кафедры «Прикладной информатики» каб. 102. тел.: раб.0312325120
Количество кредитов:	3 кредитов (45 часов)
Дата:	2018-19 учебный год, 7 семестр
Цель и задачи курса	Целью преподавания дисциплины «Фрактальный анализ рыночных данных на языке Python» является изучение основ языка высокого уровня, формирование базовых понятий структурного программирования, библиотеки стандартных модулей, принципов разработки программных систем и развитие логики студентов.
Описание курса	Предмет «Фрактальный анализ рыночных данных на языке Python» изучает: основы Python, программы в отдельном файле, строки и операции над ними, операторы отношений, условная инструкция if, модули в Python, строковые методы в Python, списки в Python, инструкции цикла в Python, множества, кортежи, словари, алгоритмы, обработка исключений в Python, работа с файлами в Python, объектно-ориентированное программирование в Python, разработка приложений с графическим интерфейсом, клиент-серверное программирование в Python.
Пре репреквизиты	Пре репреквизиты дисциплины Изучение курса «Фрактальный анализ рыночных данных на языке Python» опирается на полный цикл базовых знаний по информатике, основы программирования, интернет технологии, основы вычислительной математики, а также специальных дисциплин.
Пост репреквизиты	Пост репреквизиты дисциплины Знания по данной дисциплине необходимы для написания курсовых проектов, квалификационных работ, при подготовке презентаций, а также профессиональной подготовке.
Компетенции	<p>В результате освоения дисциплины бакалавр должен знать:</p> <ul style="list-style-type: none"> • основные приемы решения задач обработки текстовой и числовой информации; • основные способы и принципы представления структур данных; • синтаксис языка программирования Python; • приемы алгоритмизации; • основные этапы реализации программ на компьютере; • подходы структурного и объектно-ориентированного программирования; • реализацию вызова процедур в языках высокого уровня; • основные классы из библиотеки классов языка программирования Python для создания объектно-ориентированных приложений. <p>В результате освоения дисциплины бакалавр должен уметь:</p> <ul style="list-style-type: none"> • формализовать поставленную задачу; • разрабатывать программы на языке программирования Python, создавая собственные классы, а также использовать классы и модули из библиотек этого языка; • тестировать и отлаживать программы в интегрированной среде разработки; • создавать удобный интерфейс для использования созданных программных средств с помощью библиотеки Tkinter или др. <p>В результате освоения дисциплины бакалавр должен владеть:</p> <ul style="list-style-type: none"> • навыками практического программирования конкретных задач на языке программирования Python; • применять средства структурного и объектно-ориентированного

	<p>программирования для решения задач;</p> <ul style="list-style-type: none"> • навыками разработки консольных приложений в стиле объектно – ориентированного программирования на языке программирования Python; • навыками разработки приложений с внешними источниками данных (текстовыми файлами, xml-файлами, базами данных); • использовать набор библиотек языка Python для научных вычислений и научной визуализации; • применять полученные знания и навыки в процессе дальнейшего обучения, при написании курсовых и выпускных работ, а также в своей дальнейшей профессиональной деятельности.
Политика курса	<p>Для успешной работы преподавателя и студента надо соблюдать следующие правила:</p> <ul style="list-style-type: none"> • Не пропускать занятия; • отключить сотовый телефон; • активно участвовать в учебном процессе; • своевременно выполнять домашние задания.
Методы преподавания:	<p>Методы преподавания:</p> <ul style="list-style-type: none"> • лекции; • дискуссии;
Форма контроля знаний	<p>Оценка знаний будет проводиться на основе европейской системы ECTS. Система ECTS изначально делит студентов между группами «зачтено», «не зачтено», а затем оценивает работу этих двух групп по отдельности.</p> <p>Студенты, набравшие более 50 баллов, получают оценку «зачтено». Из групп получившие оценки «зачтено» на основании итогового контроля получают оценки «отлично» (от 85 до 100 баллов), «хорошо» (от 70 до 84 баллов), «удовлетворительно» (от 50 до 69 баллов).</p> <p><u>Баллы итоговой оценки распределяются следующим образом:</u></p> <p>Текущая контрольная работа – 40%</p> <p>Рубежная контрольная работа – 40%</p> <p>Итоговый контроль – 20%</p> <p><u>При выведении итоговой оценки будут учитываться активность студентов в решении задач, предлагаемых на занятиях.</u></p> <ul style="list-style-type: none"> • Текущая контрольная работа (домашние задания) необходимы для закрепления изученного материала, а также для проверки уровня понимания материала. Домашние задания будут содержать примерами, использующие основные факты и положения. Выполнение домашних заданий даст возможность студентам понимать на должном уровне пройденный материал. • Рубежная контрольная работа дается для проверки знаний по текущим материалам. Будут предложены практические и теоретические задания, раскрывающие понимание основных определений. Правильное выполнение контрольных работ, даст студентам приобрести высоких зачетных баллов. Одним из основных условий набора высоких баллов является владение студентом пройденного материала на достаточно высоком уровне. Контрольные работы будут проходить в установленное время. Передача контрольных работ не предусматривается. • Итоговый контроль – это компьютерное тестирование, чтобы студенты могли, надлежащим образом подготовиться к экзамену заранее дается перечень экзаменационных вопросов. Ответ считается наилучшим, если теоретические факты будут иллюстрированы конкретными примерами.
Литература: Основная Дополнительная	<p>Основная литература.</p> <ol style="list-style-type: none"> 1. Федоров, Д. Ю. «Программирование на языке высокого уровня Python»: учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. — М. : Издательство Юрайт, 2017. — 126 с. — (Серия : Бакалавр. Прикладной курс). 2. Николай Прохоренко, Владимир Дронов. «Python 3. Самое необходимое»:

	<p>учебное пособие для прикладного бакалавриата/ — Санкт-Петербург.: Издательство «БХВ-Петербург», 2016.</p> <p>3. Дронов В.А. «Django: практика создания Web-сайтов на Python». — СПб.: Издательство «БХВ-Петербург», 2016.</p> <p>4. С.К. Буйначев, Н.Ю.Боклаг. «Основы программирование на языке Python»: учебное пособие / С.К. Буйначев, Н.Ю.Боклаг. — Екатеринбург: Издательство Урал ин-та, 2014.</p> <p>5. МакГрат, Майк. «Программирование на Python для начинающих»: [перевод с англ. М.А. Райтмана] — Москва: Эксмо, 2015.</p> <p>6. Майкл Доусон. «Программируем на Python»: [перевод с англ.] — СПб.: Питер, 2014г. 416с.</p> <p>7. Васильев А. Н. «Python на примерах. Практический курс по программированию». - СПб.: Наука и Техника, 2016. - 432 с.: ил.</p> <p>8. Девид Бизли. «Python. Подробный справочник». [перевод с англ.] — СПб.: Символ-Плюс, 2010г. – 864с.</p> <p>Дополнительная литература.</p> <p>9. Томас Х. Кормен. Алгоритмы. Вводный курс.</p> <p>10. Томас Х. Кормен. Алгоритмы. Построение и анализ.</p> <p>11.3. Стивен С. Скиена. Алгоритмы. Руководство по разработке.</p> <p>12. Kenneth A. Lambert, Martin Osborne, Contributing Author, “Fundamentals of Python: First Programs” Printed in the United States of America. Print Year: 2017. P-498</p> <p>13. Summerfield, Mark. Programming in Python 3 : a complete introduction to the Python language / Mark Summerfield.—2nd ed. Text printed in the United States on recycled paper at RR Donnelley in Crawfordsville, Indiana. First printing, November 2009</p> <p>Online-сервисы и учебники для изучения Python</p> <p>1. Много интересных и полезных новостей из мира Python: https://pythondigest.ru/</p> <p>2. Игра для обучения программированию: https://checkio.org</p> <p>3. Online IDE: https://repl.it</p> <p>4. Визуализатор online на Python: http://pythontutor.com/visualize.html#mode=edit</p> <p>5. Задания по Python на основе рейтинга: https://www.hackerrank.com</p> <p>6. Видео лекции «Программирование на языке Python для сбора и анализа данных»</p> <p>7. Курс «Программирование на Python (Институт биоинформатики)»</p> <p>8. Курс «Python: основы и применение» (Институт биоинформатики)</p> <p>9. Видео лекции «Python 3 Basics Tutorial Series»</p> <p>10. Курс Программирование на Python от Mail.Ru Group</p>
СРС	<p style="text-align: center;">Задания лабораторных работ. Лабораторные работы по темам 2 главы.</p> <p>Функции</p> <p>Упражнение 2.1 Создайте в отдельном файле функцию, переводящую градусы по шкале Цельсия в градусы по шкале Фаренгейта по формуле: $T_F = 9/5 * T_C + 32$</p> <p>Упражнение 2.2 Создайте в отдельном файле функции, вычисляющие площадь и периметр квадрата.</p> <p>Упражнение 2.3 Напишите функцию в отдельном файле, вычисляющую среднее арифметическое трех чисел.</p> <p>Упражнение 2.4 Напишите функцию в отдельном файле, вычисляющую среднее арифметическое трех чисел. Задайте значения по умолчанию, в момент вызова используйте ключевые аргументы.</p>

Лабораторные работы по темам 3 главы.

Строки и операции над ними

Упражнение 3.1

Попросите пользователя ввести свое имя и после этого отобразите на экране строку вида: Привет, <имя>! Вместо <имя> должно указываться то, что пользователь ввел с клавиатуры.

Как тебя зовут?

Вася

Привет, Вася!

Упражнение 3.2

Напишите программу, определяющую сумму и произведение трех чисел (типа int, float), введенных с клавиатуры.

Пример работы программы:

Введите первое число: 1

Введите второе число: 4

Введите третье число: 7

Сумма введенных чисел: 12

Произведение введенных чисел: 28

Условная инструкция if

Упражнение 3.3

Напишите программу, которая запрашивает у пользователя значение pH (с плавающей точкой) и выводит на экран вещество, соответствующее введенному pH_{из} (“Яблочный сок”, “Шампунь”, “Мыло для рук”). Определение pH производится в отдельной функции, которая возвращает строку с названием вещества или фразу “Не найдено”.

Упражнение 3.4

Напишите собственную программу, определяющую максимальное из двух введенных чисел. Реализовать в виде вызова собственной функции, возвращающей большее из двух переданных ей чисел.

Упражнение 3.5

Напишите программу, проверяющую целое число на четность. Реализовать в виде вызова собственной функции.

Упражнение 3.6

Напишите программу, вычисляющую значение функции (на вход подается вещественное число):

$$f = \begin{cases} x^2 & \text{при } -2,4 \leq x \leq 5,7, \\ 4 & \text{в противном случае.} \end{cases}$$

Упражнение 3.7

Напишите программу, которая по коду города и длительности переговоров вычисляет их стоимость и результат выводит на экран: **город X**-код 343, 15 сом/мин; **город Y** -код 381, 18 сом/мин; **город G** -код 473, 13 сом/мин; **город W** -код 485, 11 сом/мин.

Упражнение 3.8

Напишите программу для определения индекса массы тела (BMI).

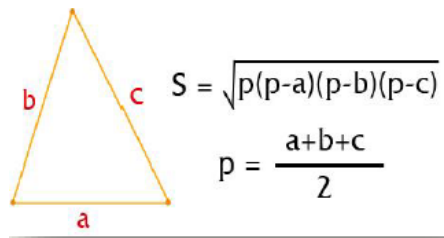
Лабораторные работы по темам 4 главы.

Модули в Python

Упражнение 4.1

Найдите площадь треугольника с помощью формулы Герона. Стороны задаются с клавиатуры. Реализовать вычисление площади в виде функции, на вход которой подаются три числа, на выходе – площадь. Функция находится в

отдельном модуле, где происходит разделение между запуском и импортированием. Описание математических функций можно найти в документации³⁵



Упражнение 4.2

Вывести число Пи с точностью до сотых.

Упражнение 4.3

Создайте в отдельном модуле функцию для вычисления выражения:

$$\sqrt{1 - \sin^2 x}$$

Упражнение 4.4

Напишите программу-игру в виде отдельного модуля. Компьютер загадывает случайное число, пользователь пытается его угадать. Программа запрашивает число ОДИН раз. Если число угадано, то выводим на экран «Победа», иначе – «Повторите еще раз». Для написания программы понадобится функция `randint()` из модуля `random`³⁶.

Упражнение 4.5

Напишите функцию, вычисляющую значение:

$$x^4 + 4^x$$

Автоматизируйте процесс тестирования функции с помощью модуля `doctest`.

Упражнение 4.6

Найдите значения выражений:

$$z = \frac{x + \frac{2+y}{x^2}}{y + \frac{1}{\sqrt{x^2+10}}} \text{ и } q = 2,8 \sin x + |y|$$

Упражнение 4.7

Напишите программу, вычисляющую значение функции (на вход подается вещественное число):

$$f = \begin{cases} \sin x & \text{при } 0,2 \leq x \leq 0,9, \\ 1 & \text{в противном случае.} \end{cases}$$

Упражнение 4.8

Напишите программу для моделирования бросания игрального кубика каждым из двух игроков. Определить, кто из игроков получил на кубике больше очков.

Строковые методы в Python.

Упражнение 4.9

`s = "У лукоморья 123 дуб зеленый 456"`

1. Определить, встречается ли в строке буква 'я'. Вывести на экран ее позицию (индекс) в строке.
2. Определить, сколько раз в строке встречается буква 'у'.
3. Определить, состоит ли строка только из букв, ЕСЛИ нет, ТО вывести строку в верхнем регистре.

4. Определить длину строки. ЕСЛИ длина строки превышает 4 символа, ТО вывести строку в нижнем регистре.
5. Заменить в строке первый символ на 'O'. Результат вывести на экран

Упражнение 4.10

Написать в отдельном модуле функцию, которая на вход принимает два аргумента: строку (s) и целочисленное значение (n).

ЕСЛИ длина строки s превышает n символов, ТО функция возвращает строку s в верхнем регистре, ИНАЧЕ возвращается исходная строка s.

Лабораторные работы по темам 5 главы.

Списки в Python

Упражнение 5.1

```
L = [3, 6, 7, 4, -5, 4, 3, -1]
```

1. Определите сумму элементов списка L. ЕСЛИ сумма превышает значение 2, ТО вывести на экран число элементов списка.

2. Определить разность между минимальным и максимальным элементами списка. ЕСЛИ абсолютное значение разности больше 10, ТО вывести на экран отсортированный по возрастанию список, ИНАЧЕ вывести на экран фразу «Разность меньше 10».

Упражнение 5.2

```
L = [3, 'hello', 7, 4, 'привет', 4, 3, -1]
```

Определите наличие строки «привет» в списке. ЕСЛИ такая строка в списке присутствует, ТО вывести ее на экран, повторив 10 раз.

Упражнение 5.3

```
L = [3, 'hello', 7, 4, 'привет', 4, 3, -1]
```

Исследуйте несколько примеров использования срезов (выполняются аналогично строкам).

```
>>> L[:3]
```

```
>>> L[:]
```

```
>>> L[::2]
```

```
>>> L[::-1]
```

```
>>> L[:-1]
```

```
>>> L[-1:]
```

Упражнение 5.4

```
L = [3, 'hello', 7, 4, 'привет', 4, 3, -1]
```

Определите наличие строки «привет» в списке. ЕСЛИ такая строка в списке присутствует, ТО удалить ее из списка, ИНАЧЕ добавить строку в список.

Подсчитать, сколько раз в списке встречается число 4, ЕСЛИ больше одного раза, ТО очистить список.

Упражнение 5.5

Напишите программу, которая запрашивает у пользователя две строки и формирует из этих строк список. Если строки состоят только из чисел, то программа добавляет в середину списка сумму введенных чисел, иначе добавляется строка, образованная из слияния двух введенных ранее строк. Итоговая строка выводится на экран.

Упражнение 5.6

Задан список слов. Необходимо выбрать из него случайное слово. Из выбранного случайного слова случайно выбрать букву и попросить пользователя ее угадать.

Задан список слов: ['самовар', 'весна', 'лето']
Выбираем случайное слово: 'весна'
Выбираем случайную букву: 'с'
Выводим на экран: ве?на
Пользователь пытается угадать букву.
Подсказка: используйте метод `choice()` модуля `random`.

Лабораторные работы по темам 6 главы.

Инструкции цикла в Python

Упражнение 6.1

Найдите все значения функции $y(x) = x^2 + 3$ на интервале от 10 до 30 с шагом 2.

Упражнение 6.2

`L = [-8, 8, 6.0, 5, 'строка', -3.1]`

Определить сумму чисел, входящих в список `L`. *Подсказка:* для определения типа объекта можно воспользоваться сравнением вида `type(-8) == int`.

Упражнение 6.3

Дан список числовых значений, насчитывающий `N` элементов. Поменяйте местами первую и вторую половины списка.

Упражнение 6.4

Напишите программу-игру. Компьютер загадывает случайное число, пользователь пытается его угадать. Пользователь вводит число до тех пор, пока не угадает или не введет слово «Выход». Компьютер сравнивает число с введенным и сообщает пользователю больше оно или меньше загаданного.

Упражнение 6.5

Дано число, введенное с клавиатуры. Определите сумму квадратов нечетных цифр в числе.

Упражнение 6.6

Найдите сумму чисел, вводимых с клавиатуры. Количество вводимых чисел заранее неизвестно. Окончание ввода, например, слово «Стоп».

Упражнение 6.7

Задана строка из стихотворения: «Мой дядя самых честных правил,
Когда не в шутку занемог,
Он уважать себя заставил И
лучше выдумать не мог»

Удалите из строки все слова, начинающиеся на букву «м». Результат вывести на экран в виде строки.

Подсказка: вспомните про модификацию списков.

Упражнение 6.8

Дан произвольный текст. Найдите номер первого самого длинного слова в нем.

Упражнение 6.9

Дан произвольный текст. Напечатайте все имеющиеся в нем цифры, определите их количество, сумму и найти максимальное.

Упражнение 6.10

Создайте матрицу (список из вложенных списков) размера `N x M` (фиксируются в программе), заполненную случайными целыми числами.

Упражнение 6.11

Создайте матрицу (список из вложенных списков) размера `N x N` (фиксируются в программе), заполненную случайными целыми числами.

Упражнение 6.12

Дана матрица (см. упражнение 6.10). Вывести номер строки, содержащей максимальное число одинаковых элементов.

Упражнение 6.13

Дана целочисленная квадратная матрица (см. упражнение 6.11). Найти произведение элементов матрицы, лежащих ниже главной диагонали.

Упражнение 6.14

Дана целочисленная квадратная матрица (см. упражнение 6.11). Найти сумму элементов матрицы, лежащих выше главной диагонали.

Лабораторные работы по темам 8 главы.

Алгоритмы

Упражнение 8.1

Напишите функцию, которая возвращает разность между наибольшим и наименьшим значениями из списка целых случайных чисел.

Упражнение 8.2

Напишите программу, которая для целочисленного списка из 1000 случайных элементов определяет, сколько отрицательных элементов располагается между его максимальным и минимальным элементами.

В упражнениях 8.3-8 список состоит из случайных элементов, в списке не менее 1000 элементов.

Упражнение 8.3

Найти элемент, наиболее близкий к среднему значению всех элементов списка.

Упражнение 8.4

Дан список, состоящий из чисел. Найти сумму простых чисел в списке.

Упражнение 8.5

Дан список целых чисел. Определить, есть ли в нем хотя бы одна пара соседних нечетных чисел. В случае положительного ответа определить номера элементов первой из таких пар.

Упражнение 8.6

Дан список целых чисел. Определить количество четных элементов и количество элементов, оканчивающихся на цифру 5.

Упражнение 8.7

Задан список из целых чисел. Определить процентное содержание элементов, превышающих среднеарифметическое всех элементов списка.

Упражнение 8.8

Задан список из целых чисел. Определить количество участков списка, на котором элементы монотонно возрастают (каждое следующее число больше предыдущего).

Упражнение 8.9

Дан список из 20 элементов. Найти пять соседних элементов, сумма значений которых максимальна.

Лабораторные работы по темам 9 главы.

Обработка исключений в Python.

Упражнение 9.1

Напишите программу, проверяющую четность числа, вводимого с клавиатуры. Выполните обработку возможных исключений.

Упражнение 9.2

Напишите программу, которая будет генерировать матрицу из случайных целых чисел. Пользователь может указать число строк и столбцов, а также диапазон целых чисел. Произведите обработку ошибок ввода пользователя.

Работа с файлами в Python.

Упражнение 9.3

Отсортированное по алфавиту содержимое файла plan.txt поместите в файл sort_plan.txt.

Следующий пример показывает, как можно напрямую обращаться к файлам, находящимся в сети Интернет:


```
import urllib.request
url = "http://dfedorov.spb.ru/python3/src/romeo.txt"
with urllib.request.urlopen(url) as webpage:
    for line in webpage:
        line = line.strip()
        line = line.decode('utf-8') # преобразуем тип bytes в utf-8
        print(line)
```

Упражнение 9.4

Напишите программу, которая создает (генерирует) полноценный HTML-документ, содержащий текст, приведенный по ссылке: <http://www.spb.ru/python/files/tutchev.txt> и под текстом размещает картинку: <http://www.spb.ru/python/files/tutchev.jpg>

Пример итогового HTML-документа: <http://www.spb.ru/python/files/p.html> (обратите внимание на код страницы, содержащей HTML-теги).

Выполните обработку ошибок.

PS. в момент чтения и записи используйте параметр функции `open()` `encoding='utf-8'`.

Регулярные выражения

Упражнение 9.5

Найдите в файле (файл находится в сети Интернет):

<http://www.spb.ru/python/files/mbox-short.txt> строки, содержащие почтовые адреса.

Запишите найденные строки в файл с именем `mail.txt`.

Упражнение 9.6

Очистите файл от HTML-тегов: <http://www.spb.ru/python/files/p.html>

Выведите на экран «чистый» текст. PS. можно использовать только стандартные модули

Python.

Упражнение 9.10

Определите частоту встречаемости всех слов для текста, находящегося в сети Интернет: <http://dfedorov.spb.ru/python3/src/romeo.txt> PS: используйте словари (`dict`).

Упражнение 9.11

Определите три наиболее популярных вида спорта в стране, исходя из количества построенных спортивных объектов для них.

Файл с данными находится по адресу: <http://www.spb.ru/python3/sport.txt>

Поля (столбцы) файла:

- порядковый номер строки;
- наименование спортивного объект;
- полный адрес спортивного объекта;
- виды спорта, для которых предназначен спортивный объект;
- пропускная способность объекта;
- вместимость объекта;
- площадь объекта (Га).

Разделитель между полями (столбцами) в файле: `'\t'`

Кодировка файла: `'cp1251'`

PS: для сбора статистики можно воспользоваться словарем (`dict`), после чего

провести его сортировку по значению.

Сортировка списка:

```
>>> sorted("This is a test string from Andrew".split())
['Andrew', 'This', 'a', 'from', 'is', 'string', 'test']
>>> str.lower("a")
'a'
>>> str.lower('Andrew')
'andrew'
```

Сортировка с предварительным применением к каждому элементу списка строкового метода `lower()`. Метод указывается в качестве значения параметра `key`.

```
>>> sorted("This is a test string from Andrew".split(),
key=str.lower)
['a', 'Andrew', 'from', 'is', 'string', 'test', 'This']
```

Создание и сортировка словаря по значению:

```
>>> d={"t1":2, "t6":5, "t9":1}
>>> d
{'t9': 1, 't6': 5, 't1': 2}
>>> sorted(d)
['t1', 't6', 't9']
>>> d.get('t1')
2
>>> sorted(d, key=d.get)
['t9', 't1', 't6']
```

Упражнение 9.12

Напишите функцию `stringCount()`, которая принимает два входных аргумента – имя файла и строку, а возвращает число повторений указанной строки в указанном файле.

Упражнение 9.13

Реализуйте функцию `myGrep()`, которая принимает два входных аргумента – имя файла и строку, а выводит на экран все строки указанного файла, содержащие заданную строку в качестве подстроки:

```
>>> myGrep('example.txt', 'line')
The 3 lines in this file end with the new line character.
There is a blank line above this line.
```

Упражнение 9.14

Реализуйте функцию `links()`, которая принимает на вход имя HTML-файла и возвращает количество гиперссылок в этом файле (тег `<a>`):

```
>>> links('twolinks.html')
```

Лабораторные работы по темам 10 главы.

Основы объектно-ориентированного подхода.

Упражнение 10.1

Создайте класс `Cat`. Определите атрибуты `name` (имя), `color` (цвет) и `weight` (вес). Добавьте метод под названием `meow` («мяуканье»). Создайте объект класса `Cat`, установите атрибуты, вызовите метод `meow`.

Упражнение 10.2

1. Напишите код, описывающий класс `Animal`:

добавьте атрибут имени животного

- добавьте метод `eat()`, выводящий «Ням-ням»
- добавьте методы `getName()` и `setName()`
- добавьте метод `makeNoise()`, выводящий «Имя животного говорит Гррр»
- добавьте конструктор классу `Animal`, выводящий «Родилось животное имя животного»

2. Основная программа:

- создайте животное, в момент создания определите его имя
- узнайте имя животного через вызов метода `getName()`
- измените имя животного через вызов метода `setName()`
- вызовите `eat()` и `makeNoise()` для животного

Упражнение 10.3

Создайте класс `StringVar` для работы со строковым типом данных, содержащий методы `set()` и `get()`. Метод `set()` служит для изменения содержимого строки, `get()` – для получения содержимого строки. Создайте объект типа `StringVar` и протестируйте его методы.

Упражнение 10.4

Создайте класс точка `Point`, позволяющий работать с координатами (x, y). Добавьте необходимые методы класса.

Упражнение 10.5

1. Напишите код, описывающий класс `Animal`:

- a) Добавьте атрибут имени животного.
- b) Добавьте метод `eat()`, выводящий «Ням-ням».
- c) Добавьте методы `getName()` и `setName()`.
- d) Добавьте метод `makeNoise()`, выводящий «Имя животного говорит Гррр».
- e) Добавьте конструктор класса `Animal`, выводящий «Родилось животное».

2. Пусть `Animal` будет родительским для класса `Cat`. Метод `makeNoise()` класса `Cat` выводит «Имя животного говорит Мяу». Конструктор класса `Cat` выводит «Родился кот», а также вызывает родительский конструктор.

3. Пусть `Animal` будет родительским для класса `Dog`. Метод `makeNoise()` для `Dog` выводит «Имя животного говорит Гав». Конструктор `Dog` выводит «Родилась собака», а также вызывает родительский конструктор.

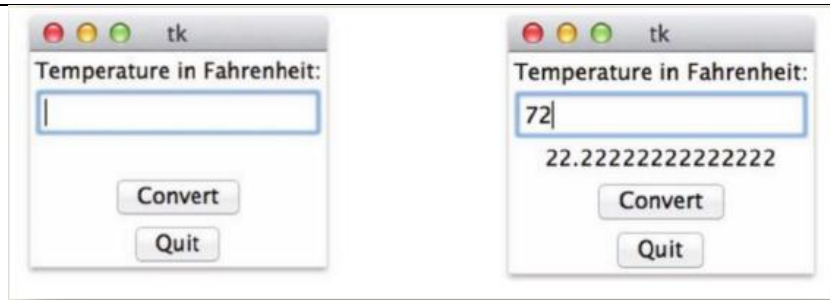
4. *Основная программа.* Код, создающий кота, двух собак и одно простое животное. Дайте имя каждому животному (через вызов методов). Код, вызывающий `eat()` и `makeNoise()` для каждого животного.

Лабораторные работы по темам 11 главы.

Разработка приложений с графическим интерфейсом.

Упражнение 11.1

Напишите программу, переводящую градусы по Фаренгейту в градусы по Цельсию. Интерфейс работы с программой представлен ниже.



Упражнение 11.2

Напишите программу, которая отображает случайное слово на русском языке (тип данных `dict`). Пользователь пытается угадать его на английском (или другом языке).

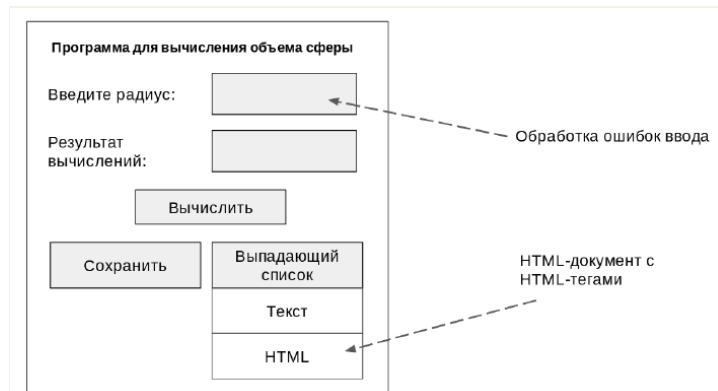
Дополнительно ограничить работу программы по числу неправильно угаданных слов.

Упражнение 11.3

Напишите программу, которая позволяет произвольный текст, введенный с клавиатуры, по нажатию кнопки сохранить в обычный текстовый файл либо в файл HTML-формата (тип файла указывается с помощью выпадающего меню).

Упражнение 11.4

Разработать программу со следующим графическим интерфейсом, предусмотреть обработку возможных ошибок.



Лабораторные работы по темам 12 главы.

Клиент-серверное программирование в Python

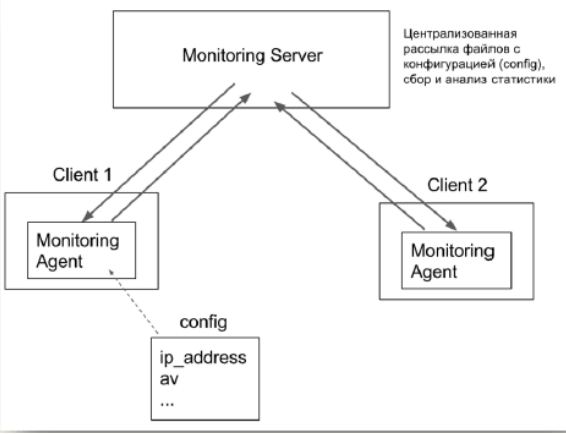
Упражнение 12.1

Разработайте программного бота, работающего по принципу клиент-серверного взаимодействия.

1. Идея бота: переводчик иностранных слов, бот-анекдотов и пр. (можно предлагать собственные идеи).
2. Разработайте систему команд для общения с ботом.
3. Реализацию необходимо построить с использованием шаблона MVC.
4. Оконный интерфейс tkinter.

Упражнение 12.2

Разработайте распределенную систему мониторинга удаленных хостов:

	 <p>Упражнение 12.3 Разработайте веб-форму (HTML+PHP) для запроса имени пользователя и пароля из базы данных (MySQL). Пароль состоит из цифр от 1 до 5. Используются GET-запросы. При правильном вводе пароля веб-сервис направляет на страницу, которая содержит «секретную» текстовую строку или ссылку на файл, содержащий «секретную» текстовую строку. Написать скрипт на языке Python, который создает текстовый файл, содержащий словарь возможных паролей, и на основании созданного словаря перебирает пароли («перебор по словарю») веб-формы. В случае подбора правильного пароля программа считывает и выводит на экран «секретную» текстовую строку. Построить график зависимости длины пароля от времени перебора.</p>
Примечание.	Домашние работы должны быть представлены в точно установленный преподавателем срок. В случае сдачи работ после установленного срока снимается 50% баллов, полученных студентом.

Календарно-тематический план распределения часов с указанием недели, темы

№	Дата	Тема	Кол-во час	Литература	Подготовительные вопросы по модулям
1	04.09.2018	Тема 1. Введение в Python. Язык программирования Python. Первая программа.	2	1. Федоров, Д. Ю. «Программирование на языке высокого уровня Python»: учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. — М. : Издательство Юрайт, 2017. — 126 с. — (Серия : Бакалавр. Прикладной курс).	1. История и тенденции развития языков программирования 2. Области применения языка программирования Python 3. Переменные в Python. Наименование. Модель памяти Python при работе с переменными
2	06.09.2018	PyCharm. Python в Visual Studio.	2		4. Функции в Python. Создание функций 5. Создание программ на языке Python в отдельном файле. Отличие от интерактивного режима
3	11.09.2018	Тема 2. Основы Python. Введение в написание программ. Переменные и типы данных. Операции с числами.	2	2. Николай Прохоренко, Владимир Дронов. «Python 3. Самое необходимое»: учебное пособие для прикладного	6. Строки и операции над строками в языке Python 7. Операторы отношений в Python. Логические операции над объектами 8. Условная инструкция if
4	18.09.2018	Условные выражения. Операции со строками. Условная конструкция if. Циклы.	2		9. Модули в Python
5	20.09.2018	Функции. Область видимости переменных. Модули.	2		

		Обработка исключений.		бакалавриата/ — Санкт-Петербург.: Издательство «БХВ-Петербург», 2016.	10. Создание собственных модулей в Python
6	25.09.2018	Тема 3. Списки, кортежи и словари. Списки.	2	3. Дронов В.А. «Django: практика создания Web-сайтов на Python». — СПб.: Издательство «БХВ-Петербург», 2016.	11. Строковые методы в Python. Отличие функций от методов
7	02.10.2018	Кортежи. Словари.	2		12. Списки в Python. Создание списка
8	04.10.2018	Множества.	2		13. Операции над списками в Python
9	09.10.2018	Тема 4. Работа с файлами. Открытие и закрытие файлов. Текстовые файлы. Файлы CSV.	2	4. С.К. Буйначев, Н.Ю.Боклаг. «Основы программирование на языке Python»: учебное пособие / С.К. Буйначев, Н.Ю.Боклаг. — Екатеринбург: Издательство Урал ин-та, 2014.	14. Пседонимы и копирование списков в Python
10	16.10.2018	Бинарные файлы. Модуль shelve. Модуль OS и работа с файловой системой.	2		15. Методы списка в Python
11	18.10.2018	Тема 5. Строки. Работа со строками. Основные методы строк.	2	5. МакГрат, Майк. «Программирование на Python для начинающих»: [перевод с англ. М.А. Райтмана] — Москва: Эксмо, 2015.	16. Преобразование типов в Python (списки, строки)
12	23.10.2018	Форматирование строк. Программа подсчета слов.	2		17. Вложенные списки в Python
13	30.10.2018	Тема 6. Основные встроенные модули. Модуль random. Модуль math.	2		18. Циклы в Python
14	01.11.2018	Модуль locale. Модуль decimal.	2		19. Цикл for для списков и строк в Python
15	06.11.2018	Тема 7. Объектно-ориентированное программирование. Классы и объекты. Инкапсуляция.	2	6. Майкл Доусон. «Программируем на Python»: [перевод с англ.] — СПб.: Питер, 2014г. 416с.	20. Функция range() и цикл for в Python
16	13.11.2018	Наследование. Полиморфизм.	2		21. Способы генерации списка в Python
17	15.11.2018	Класс object. Строковое представление объекта.	2		22. Цикл while в Python
18	20.11.2018	Тема 8. Работа с датами и временем. Модуль datetime.	2	7. Васильев А. Н. «Python на примерах. Практический курс по программированию». - СПб.: Наука и Техника, 2016. - 432 с.: ил.	23. Вложенные циклы в Python (на примере вложенных списков)
19	27.11.2018	Операции с датами.	2		24. Множества и операции над ними в Python
20	29.11.2018	Тема 9. Создание графического интерфейса. Tkinter. Создание окна приложения.	2		25. Кортежи и операции над ними в Python
21	04.12.2018	Кнопки. Изменение свойств элементов. Позиционирование элементов.	2		26. Словари и операции над ними в Python
22	11.12.2018	Текстовая метка Label. Поле ввода Entry. Checkbutton. Radiobutton.	2		27. Обработка исключений в Python
23	13.12.2018	Listbox. Меню.	1		28. Работа с файлами в Python. Менеджер контекста
		ИТОГО	45		29. Объектно-ориентированное программирование в Python. Классы, объекты
					30. Иерархия наследования в Python (класс object)
					31. Полиморфизм в Python
					32. Структура оконного приложения на примере модуля tkinter (обработка событий)
					33. Реализация шаблона «Модель-вид-контроллер» на примере модуля tkinter
					34. Структура клиент-серверного приложения (модуль socket)

График самостоятельной работы студентов

№	Недели Месяцы	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Суммы баллов
		сентябрь				октября				ноябрь				декабрь				
1	Текущий контроль	20								20								40 баллов
2	СРС	20								20								40 баллов
		22.10.18 – 27.10.18г								17.12.18- 22.12.18г								80 баллов