

Программа разработана экспертами
Федерального учебно-методического объединения
высшего образования по укрупненной группе
специальностей и направлений подготовки
45.00.00 Языкознание и литературоведение

Утверждена на заседании ФУМО
25 мая 2021 года

Примерная программа учебной дисциплины

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON

Уровень высшего образования:

БАКАЛАВРИАТ

Направление подготовки:

45.03.03 «ФУНДАМЕНТАЛЬНАЯ И ПРИКЛАДНАЯ ЛИНГВИСТИКА»

Раздел 1. Характеристики учебных занятий

1.1 Цели и задачи учебных занятий

Целью данного курса является ознакомление студентов с основами программирования на языке Python. Результатом занятий должно стать приобретение студентами навыков программирования на языке Python, применения методов программирования к задачам автоматической обработки текста.

1.2 Место дисциплины (модуля) в структуре образовательной программы, связь с другими дисциплинами (модулями) программы

Относится к вариативной части ОПОП ВО.

1.3 Требования подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты)

Устанавливаются образовательной организацией.

1.4 Перечень результатов обучения

По окончании курса студент должен знать основные алгоритмы и структуры данных; уметь формулировать алгоритм решения практической задачи, реализовывать его в виде программного кода на языке Python и применять регулярные выражения для извлечения информации из текста; владеть навыками тестирования и верификации программ, навыками применения регулярных выражений к извлечению информации из текста.

Набор компетенций, соотнесенных с результатами обучения, определяется образовательной организацией.

1.5 Перечень рекомендуемых образовательных технологий

В преподавании дисциплины «Программирование на языке Python» используются разнообразные образовательные технологии как традиционного, так и инновационного характера, учитывающие смешанный, теоретико- и практикоориентированный характер дисциплины:

- практические занятия;
- дискуссии;
- выступления с докладами и сообщениями;
- аудиторные контрольные работы;
- внеаудиторные контрольные работы;
- тестирование.

Степень необходимости образовательной среды и ее выбор определяется образовательной организацией. Формы текущей аттестации определяются образовательной организацией.

1.6 Объем дисциплины (модуля) в зачетных единицах

2 з.е.

Раздел 2. Организация, структура и содержание учебных занятий

2.1 Организация учебных занятий

Предусмотрены учебные занятия с использованием дистанционных технологий.

2.2 Краткая аннотация содержания дисциплины (модуля)

Наименование темы (раздела, части)	Вид учебных занятий	Кол-во часов
1. Введение. Установка, среды разработки. Основные приложения. Отличие Python от других языков программирования.	Практические занятия	2
2. Основные конструкции. Условный оператор. Оператор while.	Практические занятия	2
3. Цикл for. Списки в Python. Обращение по индексу, срезы. Генераторы списков.	Практические занятия	2
4. Строки. Основные операции со строками. Кортежи. Различие списков и кортежей.	Практические занятия	4
5. Словари. Методы словарей. Итераторы.	Практические занятия	2
6. Функции в Python. Позиционные и именованные параметры. Работа с файлами.	Практические занятия	2
7. Модули, импорт модулей. Основные компоненты стандартной библиотеки.	Практические занятия	2
8. Начала теории алгоритмов. Бинарный поиск. Сортировки. Сортировки выбором и слиянием. Их сложность.	Практические занятия	2
9. Основные структуры данных. Стек. Обратная польская запись. Стековый калькулятор. Связь стека и правильных скобочных последовательностей. Рекурсивные вычисления.	Практические занятия	4
10. Основные структуры данных. Очередь. Очередь с приоритетом.	Практические занятия	2
11. Графы, деревья. Обходы графов. Вычисление кратчайшего расстояния в графе.	Практические занятия	4
12. Классы. Примеры использования классов.	Практические занятия	2
13. Регулярные выражения, их приложения к задачам обработки текста.	Практические занятия	4
14. Извлечение информации из Интернета: парсинг http-страниц.	Практические занятия	2
ИТОГО		36

Раздел 3. Обеспечение учебных занятий

3.1 Методические указания по освоению дисциплины

Преподавание дисциплины осуществляется в форме практических занятий. Во время занятий обучающиеся выполняют практические задания, иллюстрирующие основные составляющие интерфейса программирования на языке Python. Для закрепления пройденного материала предлагаются домашние задания по каждой из тем. Успешное овладение содержанием дисциплины «Программирование на языке Python» предполагает работу обучающихся в группах в аудитории, а также их самостоятельную работу.

Дополнительные методические указания устанавливаются образовательной организацией.

3.2 Примерный перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине (модулю), в том числе примерный перечень учебной литературы и ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

Самостоятельная работа студентов должна включать усвоение теоретического материала, подготовку к практическим занятиям, выполнение творческих заданий, работу с электронным учебно-методическим комплексом, подготовку к текущему контролю знаний, к промежуточной аттестации (зачету).

Список рекомендованной литературы

Вабищевич П. Н. 2015. Численные методы: вычисл. практикум: [практ. применение числ. методов при использовании алгоритм. яз. Python]. М.: URSS: ЛЕНАНД.

Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. Издательский дом «Вильямс», 2005 (2-е изд.), 2013 (3-е изд.).

Прохоренок Н.А. Python 3. Самое необходимое / Николай Прохоренок, Владимир Дронов СПб.: БХВ-Петербург, 2019 и издания других годов

Описание материально-технической базы, рекомендуемой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория с мультимедийным комплексом.

Описание материально-технической базы (в т.ч. программного обеспечения), рекомендуемой для адаптации электронных и печатных образовательных ресурсов для обучающихся из числа инвалидов и лиц с ОВЗ

Устанавливается образовательной организацией

3.3 Методика проведения текущего контроля успеваемости и промежуточной аттестации и критерии оценивания

Для контроля усвоения данной дисциплины предусмотрен зачет. Мероприятия по текущему контролю знаний обучающихся проводятся в часы, отведенные для изучения дисциплины.

В течение семестра студентами выполняются практические и контрольные работы.

Порядок проведения зачета определяется ВУЗом.

3.4 Методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации (контрольно-измерительные материалы, оценочные средства)

Примерные практические задания:

1. Проверьте, является ли поданная на вход последовательность чисел монотонной (возрастающей или убывающей), в том числе нестрогой.
2. Проверьте, может ли шахматный ферзь добраться от одного поля до другого ровно за один ход. Программа получает на вход два обозначения поля, разделённые пробелом.

Каждое обозначение имеет вид <вертикаль><горизонталь>, где вертикаль принимает значение от **a** до **h**, а горизонталь от 1 до 8.

3. В футбольном турнире все пары команд должны либо сыграть по матчу дома у каждой из команд, либо не играть вообще. Вам дан список состоявшихся матчей в формате ХОЗЯЕВА-ГОСТИ. Вы должны вывести все матчи, которые осталось сыграть, чтобы это требование стало выполненным. Количество игр должно быть минимально возможным. Матчи печатаются в том же формате, что и во входных данных. Сортировку нужно произвести сначала по названию хозяев, потом по названию гостей.
4. Программа получает на вход натуральное число N ($0 < N \leq 100$) и N результатов футбольных матчей в формате <Команда_1>-<Команда_2> <ТАВ> <Голы_1>:<Голы_2>. Нужно для каждой команды вывести её название и после табуляции вывести число забитых и пропущенных мячей, разделённых двоеточием. Каждая новая команда печатается с новой строчки, команды отсортированы по разнице мячей, а в случае равной разницы – по алфавиту.
5. В фирме решили учесть рабочее время своих сотрудников. Для этого по каждому сотруднику собирается информация, показывающая время его прихода и ухода с работы. Напишите программу, получающую на вход строчку, состоящую из записей вида ВРЕМЯ_ПРИХОДА-ВРЕМЯ_УХОДА, разделённых пробелами, и возвращающую три числа: число дней, в которые работник выходил на работу, а также число часов и минут, которое он провёл в офисе. Время прихода и ухода задано в формате h:m, $0 \leq h < 24$, $0 \leq m < 60$, уход всегда не раньше прихода.

Примерный перечень заданий к зачету (экзамену) по всему курсу:

1. На вход программе поступают слова из букв латинского алфавита. Программа должна заменить каждую максимальную последовательность из гласных на строчку <VOWELS>. Гласными считать буквы a, i, u, e, o, y, знаниями английской фонетики пренебречь.
2. На вход программе поступают слова из букв латинского алфавита. Программа должна заменить каждую максимальную последовательность из гласных на строчку <VOWELS>. Гласными считать буквы a, i, u, e, o, а также последовательность букв y, если по бокам данной последовательности находятся согласные или края слова. Если слово заканчивается на e, которой предшествует согласная или y, то данная буква в ответ не печатается. Входные данные содержатся в файле input.txt, каждое слово на отдельной строчке. Ответ нужно вывести на стандартный вывод, каждое слово также печатается с новой строки.
3. Выделите с помощью регулярных выражений основу в паре лексема-словоформа. Основой считать наибольшее общее начало двух слов. На вход поступает файл input.txt со списком пар лексема-словоформа, каждая пара находится на новой строке, элементы пары разделены табуляцией. В ответе для каждой пары выведите запись $1+\langle x \rangle\#1+\langle y \rangle$, где 1 обозначает общую часть двух слов. В случае, если помимо основы слово не содержит букв, финальный плюс не печатается. Для поиска общей части обязательно использовать регулярные выражения.
4. Напишите программу, выделяющую основу в паре лексема-словоформа. Основа состоит либо из наибольшего общего префикса двух слов, либо из двух непрерывных компонент. Во втором случае вторая компонента состоит ровно из одной буквы, в начальной форме слова она отделена от первой части основы одной из букв o, e, ё, после второй компоненты идёт либо конец слова, либо мягкий знак, вслед за которым также идёт конец слова. В производной словоформе две компоненты либо непосредственно примыкают друг к другу, либо отделены от неё одной из букв ь, й. Вам нужно построить шаблон словоизменения, заменив первый фрагмент основы на

символ 1, а второй (если он присутствует) – на символ 2, при этом основа отделяется от соседних изменяемых частей символом `+`. Программа получает на вход файл input.txt, состоящий из пар лексема <tab> словоформа, записанных каждая на своей строке, в ответе нужно напечатать на стандартный выход шаблон словоизменения для каждой из пар. В шаблоне представления лексемы и изменённой словоформы разделяются символом #, для каждой новой пары ответ печатается с новой строки.

5. Напишите программу, получающую на вход предложение в формате CONLL-U (<http://universaldependencies.org/format.html>) и выделяющую в нём предложные группы. Предложной группой считать предлог и существительное, возможно, разделённые последовательностью прилагательных. Между прилагательными могут также встречаться союз "и" и запятая. Предлог и существительное определять по метке части речи (ADP и NOUN, соответственно).
6. Ваша программа читает предложение из файла input.txt и возвращает для каждой предложной группы строку <start >< end ><tab>preposition<tab>noun, где <start >и <end >— позиции предлога и существительного в предложении, а preposition и noun — сами эти слова. Позиции нумеруются с 1 и содержатся в первой колонке каждой строки. Каждая группа выводится в ответ на новой строчке.

3.5 Материально-техническое обеспечение

Минимально необходимый для реализации курса перечень материально-технического обеспечения включает лекционные аудитории (с компьютерным и видеопроекторным оборудованием для презентаций, средствами звуковоспроизведения и экраном, с выходом в Интернет). Количество индивидуальных рабочих станций должно соответствовать количеству студентов.

3.6 Информационное обеспечение

Рекомендуемая обязательная литература

Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. Издательский дом «Вильямс», 2005 (2-е изд.), 2013 (3-е изд.).

Рекомендуемая дополнительная литература

Вабищевич П. Н. 2015. Численные методы: вычисл. практикум: [практ. применение числ. методов при использовании алгоритм. яз. Python]. М.: URSS: ЛЕНАНД.

Прохоренок. Н.А. Python 3. Самое необходимое / Николай Прохоренок, Владимир Дронов СПб.: БХВ-Петербург, 2019 и издания других годов

Рекомендуемый перечень иных информационных источников

Документация на сайте python.org.

Раздел 4. Разработчики программы

Сорокин Алексей Андреевич, кандидат физико-математических наук, ассистент.

Рабочая группа ФУМО 45.00.00 по проблемам искусственного интеллекта в языкознании и литературоведении.