**Программа учебной дисциплины «Введение в программирование»**

Утверждена

Академическим советом ООП

Протокол № от «\_\_»\_\_\_\_\_20\_\_ г.

|  |  |
| --- | --- |
| Автор | Вознесенская Т.В., к. ф-м. н., доцент Департамента больших данных и информационного поиска ФКН ([tvoznesenskaya@hse.ru](mailto:tvoznesenskaya@hse.ru)) |
| Число кредитов | 5 |
| Контактная работа (час.) | 60 |
| Самостоятельная работа (час.) | 30 |
| Курс | 2-й |
| Формат изучения дисциплины | без использования онлайн курса |

1. ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ

Изучение дисциплины «Введение в программирование» нацелено на освоение языка программирования Python и восполнение пробелов в знаниях по информатике, полученных в школе.

Курс является является первым в майноре «Интеллектуальный анализ данных». Его задача научить студентов программировать на языке Python.

Целью курса также является ознакомление студентов с основными принципами проектирования и анализа алгоритмов и структур данных, развитие навыков оценки сложности алгоритмов, их практической реализации.

Предполагается, что в результате освоения курса студенты будут способны самостоятельно разработать и реализовать на языке программирования Python несложные алгоритмы.

В результате освоения дисциплины студент должен

* Знать
  + синтаксис и семантику основных конструкций языка программирования Python.
* Уметь
  + подбирать структуры данных для прикладных задач,
  + реализовывать алгоритмы на языке программирования Python,
  + работать в среде разработки PyCharm,
  + оценивать сложность реализуемых алгоритмов.

Для освоения учебной дисциплины, студенты должны владеть основными понятиями школьного курса информатики.

Основные положения дисциплины должны (могут) быть использованы в дальнейшем при изучении следующих дисциплин:

* Анализ данных
* Введение в машинное обучение
* Интеллектуальные системы

1. СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

**Тема 1. Особенности языка Python.**

Парадигмы программирования. История возникновения. Сильные и слабые стороны языка Python. Функциональные возможности. Типичные прикладные задачи. Архитектура программы.

**Тема 2. Введение в типы объектов языка Python.**

Числа, строки, списки, кортежи, словари, множества, файлы. Операции с объектами.

**Тема 3. Простые типы. Основные операторы.**

Числа. Динамическая типизация. Оператор присваивания. Условный оператор. Логические операторы. Операторы цикла.

**Тема 4. Работа с последовательностями.**

Строки, списки, кортежи.

**Тема 5. Отображения. Стиль программирования.**

Словари. Стиль программирования на Python (Стандарт PEP-8). Документирование кода. (Стандарт PEP-257).

**Тема 6. Функции.**

Назначение функций. Области видимости. Рекурсия.

**Тема 7. Работа с библиотеками.**

Обзор стандартной библиотеки Pyhton. Работа с текстовыми и бинарными файлами.

**Тема 8. Организация взаимодействия с пользователем.**

Интерфейсы. Web-разработка. GUI-приложения. Библиотека tkinter.

**Тема 9. Методы сортировки одномерных массивов.**

Метод пузырька, сортировка вставками, слиянием, Timsort, tree sort.

**Тема 10. Алгоритмы поиска в одномерных массивах.**

Линейный и бинарный методы.

**Тема 11. Структуры данных.**

Отображение абстрактных структур данных на структуры данных хранения. Стек, очередь, куча, дерево, граф.

**Тема 12. Алгоритмы на графах.**

Обход в глубину и в ширину.

**Тема 13. Интересные задачи.**

Примеры решения нестандартных задач. Использование особенностей языка Python.

**Тема 14. Основы объектно-ориентированного программирования.**

Классы. Основные парадигмы ООП. Примеры объектных моделей.

**Тема 15. Отладка и профилирование программы.**

Работа с отладчиком. Обработка исключений.

1. Оценивание

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тип  контроля | Форма контроля | 1 семестр | | Параметры \*\* |
| 1 модуль | 2 модуль |
| Текущий | Контрольная работа |  | 8 неделя | На темы 1-8 разделов |
| Текущий | Домашнее задание | 15 неделя |  | Решение задач в компьютерной системе на темы 1-14 разделов |
| Итоговый | Экзамен |  | \* | Устный экзамен |

В рамках курса студенты должны написать одну контрольную работу, выполнить одну домашнюю работу (далее ДР), состоящую из 6 частей, и сдать экзамен. Контрольная работа, ДР и экзамен оцениваются по 10-балльной шкале.

Для каждой части домашнего задания устанавливается срок сдачи.

Каждая задача имеет определенную «стоимость» – максимальное количество баллов, которое возможно получить за ее решение. Суммарная «стоимость» каждого домашнего задания – 10 баллов.

Решенную задачу (код) необходимо загрузить в систему автоматического приема задач. Решения, успешно прошедшие автоматические тесты, затем проверяются преподавателем. Преподаватель проверяет одну и ту же задачу не более двух раз, то есть у студента есть одна попытка для исправления замечаний.

За задание, сданное позже срока, можно получить не более половины от недостающего количества баллов: например, если в срок студент набрал 6 баллов, тогда позже срока он может набрать не более, чем (10-6)/2=2. Сроком сдачи считается время первой загрузки задачи в систему. При исправлении замечаний студент должен уложиться в общий срок приема задачи. Если исправленная задача не будет загружена в систему до истечения указанного срока, то оценивается исходный вариант. Задачи перестают приниматься через месяц после выдачи.

Итоговая оценка за все домашнее задание считается как среднее арифметическое по всем частям. Округление по стандартным правилам. 0.5 трактуется в пользу студента. Например, N=60. Студент набрал 40, то есть 0.67\*N. Оценка 7.

Работы промежуточного контроля заключаются в решении задач на одну или несколько тем. Экзамен состоит в ответе на один вопрос и решении двух задач.

Накопленная оценка за текущий контроль формируется из оценок за домашнее задание и контрольную работу следующим образом:

Накопленная = Средняя(КР, ДР)

Итоговая оценка за дисциплину рассчитывается следующим образом:

Итоговая = 0.6\*Накопленная+0.4\*экзамен

При накопленной оценке от 8 и выше возможно автоматическое получение итоговой оценки равной накопленной. Данное условие является необходимым, но не достаточным. Решение принимает преподаватель, ведущий семинарские занятия.

1. ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ

Образец домашнего задания

№111152. (1 балл). *Возрастает ли список?* Дан список. Определите, является ли он монотонно возрастающим(то есть верно ли, что каждый элемент этого списка больше предыдущего). Выведите YES, если массив монотонно возрастает и NO в противном случае. Решение оформите в виде функции IsAscending(A). В данной функции должен быть один цикл while, не содержащий вложенных условий и циклов — используйте схему линейного поиска.

№111155. (2 балла). *Противоположные элементы в массиве*. Дан список чисел. Определите, есть ли в нем два противоположных(то есть дающих в сумме 0) числа. Если такие числа есть в массиве, выведите их индексы в порядке возрастания. Если таких чисел в массиве нет, ничего не выводите. Гарантируется, что таких пар не больше одной.

№111156. (3 балла). *Сортировка выбором.* Дан список целых чисел.Выведите все элементы этого списка в порядке невозрастания значений.Выведите новый список на экран. Решите эту задачу при помощи алгоритма сортировки выбором.Решение оформите в виде функции SelectionSort(A). В алгоритме сортировки выбором мы находим наибольший элемент в списке и ставим его на первое место, затем находим наибольший элемент из оставшихся и ставим его на второе место и т.д.

*№*111157. (4 балла). *Сортировка вставкой.* Дан список целых чисел. Отсортируйте его в порядке неубывания значений. Выведите полученный список на экран. Решите эту задачу при помощи алгоритма сортировки вставкой.Решение оформите в виде функции InsertionSort(A). В этой задаче нельзя пользоваться дополнительным списком операциями удаления и вставки элементов. В алгоритме сортировки вставкой в каждый произвольный момент начальная часть массива уже отсортирована. В решении имеется цикл for i in range(1, len(A)), внутри которого предположении, что элементы списка A[0],A[1], ..., A[i-1]уже отсортированы, элемент A[i]добавляется в отсортированную часть списка.Для этого находится позиция, в которую необходимо вставить элемент A[i], затем осуществляется циклический сдвиг фрагмента уже отсортированной части.

*№*111158. (4 балла). *Сортировка пузырьком.* Дан список целых чисел. Отсортируйте его в порядке невозрастания значений. Выведите полученный список на экран. Решите эту задачу при помощи алгоритма пузырьковой сортировки. Решение оформите в виде функции BubbleSort(A). В алгоритме пузырьковой сортировки осуществляется проход по списку от начала к концу, и если два соседних элемента списка стоят в неверном порядке, то они переставляются в правильном порядке. В результате минимальный элемент массива окажется на последнем месте. Повторим эту процедуру еще несколько раз, чтобы поставить все элементы на свои места. Вспомогательным списком пользоваться нельзя.

Образец контрольной работы

№3463. (1 балл). *Число десятков*. Дано натуральное число. Найдите число десятков в его десятичной записи.

№3467. (1 балл). *Парты.* В некоторой школе решили набрать три новых математических класса и оборудовать кабинеты для них новыми партами. За каждой партой может сидеть два учащихся. Известно количество учащихся в каждом из трех классов. Выведите наименьшее число парт, которое нужно приобрести для них.

№3739. (2 балла). *Первое и последнее вхождение.* Дана строка. Если в этом числе буква f встречается только один раз, выведите её индекс. Если она встречается два и более раз, выведите индекс её первого и последнего появления. Если буква f в данной строке не встречается, ничего не выводите.

№3802. (2 балла). *Числа Фибоначчи.* Напишите функцию phib(n), которая по данному целому неотрицательному n возвращает *n*-e число Фибоначчи. *В этой задаче нельзя использовать циклы - используйте рекурсию.*

№3549. (3 балла). *Замечательные числа — 5.* Даны два четырёхзначных числа A и B. Выведите в порядке возрастания все четырёхзначные числа в интервале от A до B, запись которых содержит ровно три одинаковые цифры.

№3765. (3 балла). *Страны и города.* Дан список стран и городов каждой страны. Затем даны названия городов. Для каждого города укажите, в какой стране он находится. Входные данные. Программа получает на вход количество стран N. Далее идет N строк, каждая строка начинается с названия страны, затем идут названия городов этой страны. В следующей строке записано число M, далее идут M запросов— названия каких-то M городов, перечисленных выше. Выходные данные. Для каждого из запроса выведите название страны, в котором находится данный**.**

**Вопросы к экзамену.**

1. Основные характеристики языка Python.

Сильные и слабые стороны. Парадигма программирования. Области применения. Архитектура программы.

2. Python – интерпретируемый язык.

Интерпретатор Python. Виртуальная машина Python.Отличие процесса интерпретации от компиляции. Байт код. Плюсы и минусы интерпретации. Переносимость программ на Python.

3. Типы объектов в языке Python.

Числа, строки, кортежи, словари, множества, файлы. Рассказать коротко о каждом типе. Динамическая типизация.

4. Основные конструкции языка.

Оператор присваивания. Условный оператор. Логические операторы. Операторы цикла. Полиморфизм операций.

5. Числа в Python.

Изменяемые и неизменяемые объекты, отличия между ними в наборе операций. Типы int и float. Decimal. Операции с числами.

6. Строки в Python.

Изменяемые и неизменяемые объекты, отличия между ними в наборе операций. Тип str. Операции со строками. Встроенные методы.

7. Списки в Python.

Изменяемые и неизменяемые объекты, отличия между ними в наборе операций. Тип list. Операции со списками. Встроенные методы. Генераторы списков.

8. Кортежи в Python.

Изменяемые и неизменяемые объекты, отличия между ними в наборе операций. Тип tuple. Операции с кортежами. Встроенные методы.

9. Множества в Python.

Изменяемые и неизменяемые объекты, отличия между ними в наборе операций. Тип set. Операции с множествами. Генераторы множеств.

10. Словари в Python.

Изменяемые и неизменяемые объекты, отличия между ними в наборе операций. Тип dict. Операции со словарями. Генераторы словарей.

11. Функции.

Области видимости. Аргументы и способы их передачи..

12. Рекурсия.

Понятие рекурсии. Рекурсивные функции. Поиск чисел Фибоначчи с помощью рекурсии.

13. Работа с файлом в Python.

Изменяемые и неизменяемые объекты, отличия между ними в наборе операций. Режимы работы с файлами. Открытие и закрытие файлов.

14. Модули в Python.

Архитектура программы. Инструкция import.. Работа с библиотеками в Python.

15. Модули в Python.

Архитектура программы. Инструкция import.. Работа с библиотеками в Python.

16. Библиотека NumPy.

Архитектура программы. Инструкция import.. Основные возможности библиотеки NumPy.

17. Сотировка методом пузырька.

Метод сортировки одномерных массивов (метод пузырька)

18. Сортировка выбором.

Метод сортировки одномерных массивов (сортировка выбором)

19. Сортировка вставками.

Метод сортировки одномерных массивов (сортировка вставками.)

20. Сортировка слиянием.

Метод сортировки одномерных массивов (сортировка слиянием.)

21. Поиск.

Понятие поиска. Алгоритмы поиска в одномерные массивы. Линейный и бинарный методы.

22. Структура данных Стек.

Структуры данных. Отображение абстрактных структур данных на структуры данных хранения. Методы стека. Пример реализации на Python.

23. Структура данных Очередь.

Структуры данных. Отображение абстрактных структур данных на структуры данных хранения. Методы очереди. Пример реализации на Python.

24. Структура данных Дек.

Структуры данных. Отображение абстрактных структур данных на структуры данных хранения. Методы дека. Пример реализации на Python.

25. Структура данных Куча.

Структуры данных. Отображение абстрактных структур данных на структуры данных хранения. Пример реализации на Python.

26. Структура данных Граф.

Структуры данных. Отображение абстрактных структур данных на структуры данных хранения. Способы представления графа в памяти. Пример реализации на Python.

27. Структура данных Дерево.

Структуры данных. Отображение абстрактных структур данных на структуры данных хранения. Двоичное дерево. Пример реализации на Python.

28. Алгоритм обхода графа в ширину.

Идея алгоритма. Код. Сложность. Разбор на примере.

29. Алгоритм обхода графа в глубину.

Идея алгоритма. Код. Сложность. Разбор на примере.

30. Подсчет числа компонент связности с помощью алгоритма обхода в глубину.

Идея алгоритма. Код. Сложность. Разбор на примере.

1. Ресурсы

Основная литература

# Pooja Sharma. Programming in **Python,** BPB Publications, 2017

Дополнительная литература

1. Марк Лутц. Изучаем Python. 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011.
2. Марк Лутц. Python. Карманный справочник. – Вильямс, 2015.
3. Марк Лутц. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011.
4. Марк Лутц. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011.

**Программное обеспечение**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
| 1. | Microsoft Windows 10 | *Из внутренней сети университета (договор)* |
| 2. | JetBrains PyCharm Community Edition | *Свободно распространяемое лицензионнон соглашение* |

**Материально-техническое обеспечение дисциплины**

Учебные аудитории для лекционных занятий по дисциплине обеспечивают использование и демонстрацию тематических иллюстраций, соответствующих программе дисциплины в составе:

ПЭВМ с доступом в Интернет (операционная система, офисные программы, антивирусные программы);

мультимедийный проектор с дистанционным управлением.

Учебные аудитории для cеминарских и самостоятельных занятий по дисциплине оснащены ­­­­­­­­­­­­­­­­­­­­­­­­ ПЭВМ с доступом в Интернет (операционная система, офисные программы, антивирусные программы) и доступом к электронной информационно-образовательной среде НИУ ВШЭ.