

Департамент информатизации Тюменской области
Государственное автономное учреждение дополнительного образования
Тюменской области
«Региональный информационно-образовательный центр»

СОГЛАСОВАНО

УТВЕРЖДАЮ

Директор
ГАУ ТО «Западно-Сибирский
инновационный центр»

Директор
ГАУ ДО ТО «РИО-Центр»

А.В. Сакевич

2019

Т.А. Беляева

« 2019



УЧЕБНАЯ ПРОГРАММА

«Основы анализа данных с использованием языка Python»

Трудоемкость программы – 42 академических часа

Форма обучения – очная

Режим занятий –6 академических часов в неделю

Начальные навыки: Базовые навыки работы на персональном компьютере

Цель обучения: сформировать у слушателей комплекс знаний, умений, принципов и практических навыков программирования для сбора и анализа данных на языке Python.

Компетенции на выходе:

слушатели обладают всеми необходимыми знаниями, умениями и навыками для программирования для сбора и анализа данных на языке Python.

Знания:

- знать место языка Python среди языков программирования высокого уровня;
- знать особенности структуры программы, представленной на языке Python;
- знать возможности и ограничения использования готовых модулей;
- знать, что такое операция, операнд и их характеристики;
- знать принципиальные отличия величин структурированных и неструктурированных;
- знать математические функции, входящие в Python;
- знать основные операторы языка Python, их синтаксис;
- знать правила описания процедур и функций в Python и построение вызова процедуры;
- знать принципиальные отличия между формальными, локальными и глобальными переменными;
- знать область действия описаний в процедурах;
- знать, как с помощью списков определять в программе тип «массив», «матрица»;
- знать свойства данных типа «массив», «матрица».

Умения:

- уметь записывать примеры арифметических и логических выражений всех атрибутов, которые могут в них входить;
- уметь разрабатывать программы обработки числовой и символьной информации;
- уметь разрабатывать программы (линейные, разветвляющиеся и с циклами);
- уметь воспроизводить алгоритмы сортировки массивов и матриц, поиска в упорядоченном массиве, распространять эти алгоритмы на сортировку и поиск в нечисловых массивах;
- уметь читать и записывать текстовые файлы в заданном формате;
- уметь решать основные алгоритмические задачи, представленные в пояснительной записке, в среде Python;
- уметь собирать, анализировать информацию и строить визуальные отчеты.

Приёмы работы:

- с модулями, входящими в состав среды Python;
- с величиной, ее характеристиками;
- с такими структурами данных, как число, текст, кортеж, список, словарь;
- с составом арифметического выражения;
- с логическими выражениями и входящих в них операндах, операциях и функциях;
- с процессами исполнения каждого из операторов;
- с значениями полноценных процедур и функций для структурно-ориентированного языка высокого уровня;
- с рекурсией, знать ее реализацию на Python;
- с основными приемами формирования процедуры и функций;
- с основными алгоритмическими задачами, представленными в пояснительной записке, в среде Python.

Программой обучения предусмотрены аудиторные занятия и самостоятельная работа слушателей. Под самостоятельной работой подразумевается самостоятельное выполнение слушателями практических заданий в учебной аудитории и в присутствии преподавателя.

УЧЕБНЫЙ ПЛАН

№ п/п	Наименование тем	Всего часов	в том числе:		Формы контроля
			аудиторн ые занятия	самос- тоятель- ная работа	
1.	Синтаксис и управляющие конструкции языка Python	16	16		Защита проекта
2.	Последовательности в Python	14	14		Защита проекта
3.	Анализ данных в Python	12	12		Защита проекта
Итого часов:		42	42		

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Наименование тем	Всего часов	в том числе:		Формы контроля
			аудиторн ые занятия	самосто- тельная работа	
1.	Синтаксис и управляющие конструкции языка Python	16	16		
1.1.	Компилируемые и интерпретируемые языки программирования. Примеры и сравнения	1	1		Выполнение практических заданий
1.2.	Общие сведения о языке Python и особенности его стиля программирования	1	1		
1.3.	Переменные, значения и их типы. Присваивание значения. Ввод значений с клавиатуры	1	1		
1.4.	Встроенные операции и функции. Основные алгоритмические конструкции	2	2		
1.5.	Условный оператор. Множественное ветвление. Условия равенства/неравенства	2	2		
1.6	Циклы и счетчики	3	3		
1.7	Определение функций. Параметры и аргументы. Вызовы функций. Оператор возврата return	3	3		
1.8	Определение класса. Методы класса	1	1		
1.9	Работа над проектом	2	2		Защита проекта
2.	Последовательности в Python	14	14		
2.1	Списки, кортежи и словари	2	2		Выполнение практических заданий
2.2	Операторы, общие для всех типов	2	2		

	последовательностей				ческих заданий
2.3	Специальные операторы и функции для работы со списками	2	2	-	
2.4	Работа со словарями	2	2	-	
2.5	Вложенные списки	2	2	-	
2.6	Работа над проектом	4	4	-	Защита проекта
3.	Анализ данных в Python	12	12	-	
3.1.	Работа с библиотекой Pandas	2	2	-	Выполне- ние практи- ческих заданий
3.2.	Объединение таблиц и аналитика	2	2	-	
3.3.	Визуализация базовых с помощью библиотеки Matplotlib	2	2	-	
3.4.	Парсинг сайтов и основы HTML	2	2	-	
3.5.	Работа над проектом	4	4	-	
	Всего часов:	42	42	-	

СОДЕРЖАНИЕ ПРОГРАММЫ

Раздел 1. Синтаксис и управляющие конструкции языка Python

1.1. Компилируемые и интерпретируемые языки программирования.

Примеры и сравнения

Классификация языков программирования, машинно-ориентированные языки, машинно – независимые языки.

1.2. Общие сведения о языке Python и особенности его стиля программирования

Объектноориентированный язык. Программные модули и структуры.

1.3 Переменные, значения и их типы. Присваивание значения.

Ввод значений с клавиатуры

Переменная в языке программирования, типы данных.

1.4. Встроенные операции и функции. Основные алгоритмические конструкции.

Основные алгоритмические конструкции, последовательность операторов, оператор условия и выбора.

1.5. Условный оператор. Множественное ветвление. Условия равенства/неравенства.

Логические операторы. Сложные логические выражения.

1.6. Циклы и счетчики.

Цикл while. Цикл for. Функция range. Вложенные циклы. Выход из цикла break и continue.

1.7. Определение функций. Параметры и аргументы. Вызовы функций. Оператор возврата return.

Функции, именные и анонимные, инструкции def, return и lambda, обязательные и необязательные аргументы функций, функции с произвольным

числом аргументов.

1.8. Определение класса. Методы класса.

Поддержка объектноориентированного программирования: определение собственных классов и базовых классов. Методология написания кода.

1.9. Работа над проектом.

Понятия «проект», «проектная деятельность», «проектная группа». Основы, особенности проектной деятельности. Этапы проектной деятельности. Работа над проектом. Защита проекта.

Раздел 2. Последовательности в Python

2.1. Списки, кортежи и словари

Перебор кортежей. Сложные кортежи Тип file, функция tuple, кортеж countries.

2.2. Операторы, общие для всех типов последовательностей

Арифметические операторы в Python. Операторы сравнения в Python. Операторы присваивания в Python. Побитовые операторы в Python

2.3. Специальные операторы и функции для работы со списками

Функция extend, insert. Операции над списками, генераторы списков и применение списков.

2.4. Работа со словарями

Методы словарей, работа со словарями, операции над ними, методы, генераторы словарей.

2.5. Вложенные списки

Матрицы. Обработка и вывод вложенных списков. Создание вложенных списков. Ввод двумерного массива. Вложенные генераторы двумерных массивов.

2.6. Работа над проектом

Разработка и создание программы, используя приобретенные знания, умения и навыки. Работа над проектом. Защита проекта.

Раздел 3. Анализ данных в Python

3.1. Работа с библиотекой Pandas

Изучение наиболее важных функций библиотеки Pandas.

3.2. Объединение таблиц и аналитика

Объединение нескольких таблиц в одну и анализ информации.

3.3. Визуализация данных с помощью библиотеки Matplotlib

Изучение основных функций библиотеки Matplotlib и построение визуальных отчетов.

3.4. Парсинг сайтов и основы HTML

Извлечение информации с WEB-страниц для анализа данных, знакомство с библиотекой BeautifulSoup.

3.5. Работа над проектом

Разработка и создание программы, используя приобретенные знания, умения и навыки. Работа над проектом. Защита проекта.