

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

СОГЛАСОВАНО

Исполняющий обязанности директора
ГАУ ТО «Западно-Сибирский
инновационный центр»

 А.Н.Ермолаев
« 20 » сентября 2022 г.

УТВЕРЖДАЮ

Директор
ГАУ ДО ТО «РИО-Центр»

 Т.А.Беляева
« 20 » сентября 2022 г.

УЧЕБНАЯ ПРОГРАММА
«Python-программирование»

Трудоемкость программы – 80 академических часов

Форма обучения – дистанционная, аудиторная

Режим занятий – 3 академических часа в день, до 5 дней в неделю

**Начальные навыки – базовые навыки работы на персональном
компьютере**

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

Компетенции на выходе:

слушатели обладают всеми необходимыми знаниями, умениями и навыками для программирования 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 не структурированных;
- 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.

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 функции;
- с основными алгоритмическими задачами, представленными в пояснительной записке, в среде Python.

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

УЧЕБНЫЙ ПЛАН

№	Темы	Дистанционные / аудиторные занятия	Самостоятельная работа, консультации	Контроль
1.	Синтаксис и управляющие конструкции	12	6	Практика / лабораторная работа
2.	Модули и пакеты в Python	4	8	Практика / лабораторная работа
3.	Типы в Python	10	4	Практика / лабораторная работа
4.	Внешние модули в Python	36		Практика
	Итого часов:	80		

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Темы	Дистанционные / аудиторные занятия	Самостоятельная работа, консультации	Контроль
1.	Синтаксис и управляющие конструкции			

1.1	Компилируемые и интерпретируемые языки программирования	1		
1.2	Общие сведения о языке Python и особенности его стиля программирования	1		
1.3	Переменные, значения и их типы. Присваивание значения. Ввод значений с клавиатуры.	1		
1.4	Встроенные операции и функции. Основные алгоритмические конструкции.	1		
1.5.	Условный оператор. Множественное ветвление. Условия равенства/неравенства.	2		
1.6	Циклы и счетчики.	2		
1.7	Определение функций. Параметры и аргументы. Вызовы функций. Оператор возврата return.	2	2	Практика / лабораторная работа
1.8	Определение класса. Методы класса.	2	4	Практика / лабораторная работа
2.	Модули и пакеты в Python.			
2.1.	Основные стандартные модули и пакеты в Python.	1	2	Практика / лабораторная работа
2.2.	Импортирование модулей.	1	2	Практика / лабораторная работа
2.3.	Создание собственных модулей и их импортирование.	1	2	Практика / лабораторная работа
2.4.	Специализированные модули и приложения.	1	2	Практика / лабораторная работа
3.	Типы в Python.			
3.1.	Списки, кортежи и словари.	2	1	Практика / лабораторная работа
3.2.	Операторы общие для всех типов последовательностей.	2	1	Практика / лабораторная работа
3.3.	Специальные операторы и функции для работы со списками.	2	1	Практика / лабораторная работа

3.4.	Работа со словарями.	2	1	Практика / лабораторная работа
3.5.	Вложенные списки.	2		
4.	Внешние модули в Python.			
4.1	Tkinter	10		Практика / лабораторная работа
4.2	MathPlotLib+SymPy	4		Практика
4.3	wxPython	4		Практика
4.4	PyGame	6		Практика
4.5	Flask	4		Практика
4.6	Django	4		Практика
4.7	OpenGL	4		Практика
	Всего часов:	80		

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

Объектно-ориентированный 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, именованные и анонимные, инструкции def, return и lambda, обязательные и необязательные аргументы функции, функции с произвольным числом аргументов.

1.8. Определение класса. Методы класса.

Поддержка объектно-ориентированного программирования: определение собственных классов и базовых классов. Методология написания кода

Раздел 2. Модули и пакеты в Python.

2.1. Основные стандартные модули и пакеты в Python.

Изучение наиболее важных модулей и пакетов стандартных библиотек Python

2.2. Импортирование модулей.

Подключение модуля из стандартной библиотеки Использование псевдонимов

2.3. Создание собственных модулей и их импортирование.

Повторное использование кода: такой код может быть загружен много раз во многих местах. Управление адресным пространством

2.4. Специализированные модули и приложения.

Пакетирование и дистрибуция приложений Python

Раздел 3. Последовательности в Python.

3.1. Списки, кортежи и словари.

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

3.2. Операторы общие для всех типов последовательностей.
Арифметические операторы в Python. Операторы сравнения в Python. Операторы присваивания в Python. Побитовые операторы в Python

3.3. Специальные операторы и функции для работы со списками.
Функция extend, insert. Операции над списками, генераторы списков и применение списков.

3.4. Работа со словарями.

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

3.5. Вложенные списки.

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

Раздел 4. Внешние модули в Python.

4.1 Tkinter.

Описание. Модель использование. Практика

4.2 MathPlotLib+SymPy.

Описание. Модель использование. Практика

4.3 wxPython.

Описание. Модель использование. Практика

4.4 PyGame.

Описание. Модель использование. Практика

4.5 Flask.

Описание. Модель использование. Практика

4.6 Django.

Описание. Модель использование. Практика

4.7 OpenGL.

Описание. Модель использование. Практика