УДК 004.94 **Информационные технологии**

Зайцев Дмитрий Сергеевич, *аспирант 2 курс, институт математики, естествознания и техники, Елецкий государственный университет им. И.А. Бунина, Россия, г. Елец*

ИСПОЛЬЗОВАНИЕ PYTHON ДЛЯ ПОСТРОЕНИЯ СЕТЕВЫХ МОДЕЛЕЙ

Аннотация: В статье анализируются особенности использования Python для построения сетевых моделей. Рассматриваются понятие и сущность сетевых моделей, и ключевые свойства языка программирования Python. Выявляются типы данных, используемых в Python. Приводятся основные средства программирования языка, такие как операции с числами и строками, операторы ветвления и выбора, циклы, двумерные и ассоциативные массивы, обеспечивающие построение и визуализацию сетевых моделей.

Ключевые слова: язык программирования, Python, средства программирования, операторы, функции, сетевые модели, визуализация.

Annotation: The article analyzes the features of using Python to build network models. The concept and essence of network models and key properties of the Python programming language are considered. The data types used in Python are revealed. The main language programming tools are presented, such as operations with numbers and strings, branching and selection operators, loops, two-dimensional and associative arrays, which provide the construction and visualization of network models.

Keywords: programming language, Python, programming tools, operators, functions, network models, visualization.

Python представляет собой один из наиболее распространённых языков программирования общего назначения, входя в число самых

быстроразвивающихся языков программирования в мире [1]. Он является высокоуровневым, интерпретируемым и объектно-ориентированным языком, поддерживающим динамическую привязку и динамическую типизацию. Python содержит множество различных пакетов и библиотек для работы с данными, посредством которых можно проводить большое количество операций и преобразований с исходными сведениями. Одной из разновидностей таких операций является построение сетевых моделей, представляющих собой математическую модель определённого комплекса работ [2]. Такие модели представляются в виде сетевого графика (рис. 1), который связывает события, происходящие в ходе выполнения работ.

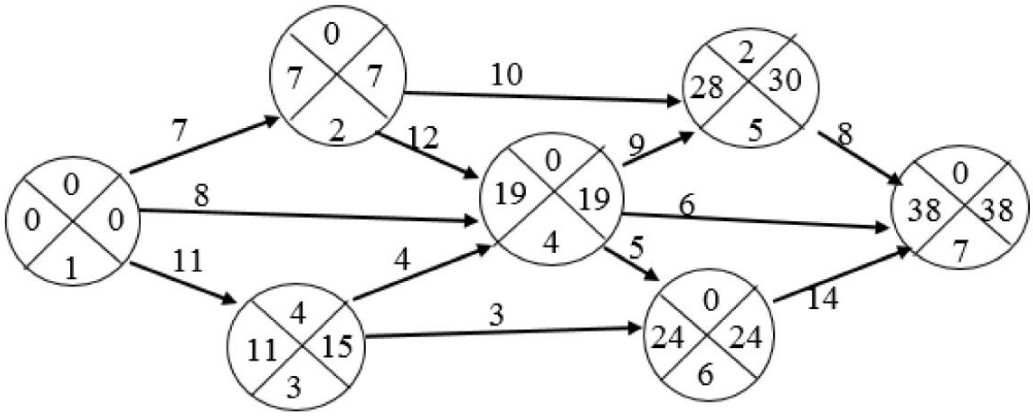

Рисунок 1. Пример сетевого графика, включающего 7 событий и 12 связывающих работ. Рядом со стрелками указана длительность каждой работы, внутри круга – временные параметры события.

Установление связи между событиями осуществляется при помощи промежуточных действий, определяющих выполнение всего проекта. Анализ сетевых моделей даёт возможность чётко выявить взаимосвязи всех этапов реализации проекта и определить оптимальный порядок их выполнения [3]. Значимость сетевых моделей для оптимизации управления проектами делает актуальным исследования применения языка Python для их построения.

Целью работы является изучение особенностей использования Python для построения сетевых моделей.

Популярность Python обусловлена такими его свойствами, как простота,

чёткость и последовательность синтаксиса, свободное распространение, отсутствие компиляции кода, поддержка обработки исключений, автоматическое управление памятью и ряд других [4]. Он содержит инструменты практически для всех аспектов анализа данных, позволяя обрабатывать и визуализировать данные любого типа посредством набора высокоуровневых математических функций. Python поддерживает несколько стилей программирования и процедурное программирование, при этом содержа ограниченную поддержку функционального программирования [5].

В Python реализована технология динамической типизации объектов, сущность которой сводится к тому, что каждый созданный объект является ссылкой [6]. Все данные в языке Python представляют собой объекты, меняющие свои типы при изменении ссылки. Это позволяет избежать необходимости заблаговременного объявления типа переменных, в отличие от других языков программирования.

— В языке Python возможно работать со следующими разновидностями данных:

— числовые типы, включая целые числа int() и числа с плавающей точкой (действительные float() и комплексные complex());

— строки – неизменяемые последовательности символов, заключаемые в одиночные либо двойные кавычки;

— списки – сложный тип данных в виде изменяемой упорядоченной последовательности различных объектов, заключаемой в квадратные скобки;

— кортежи – неизменяемые списки, заключаемые в круглые скобки;

— словари – изменяемые неупорядоченные наборы пар «ключ: значение», первая часть которых уникальная, заключаемые в фигурные скобки;

— множества – совокупность расположенных в случайном порядке неповторяющихся элементов, функционирующих как математическое множество.

В число основных средств программирования Python входят операции с числами и строками. К арифметическим операциям с числами относятся

сложение +, вычитание -, умножение *, деление /, деление нацело //, остаток от деления % и возведение в степень **[7]. Ключевыми встроенными функциями являются:

— abs(x) – модуль числа x;
— round(x) – округление;
— round(x, n) – округление числа x до n знаков после запятой;
— divmod(x, y) – вычисление частного и остатка;
— max(a, b, ...) – максимальное число из двух или более;
— min(a, b, ...) – минимальное число из двух или более;
— max(seq) – максимальный элемент последовательности;
— min(seq) – минимальный элемент последовательности;
— sum(seq) – сумма элементов последовательности;
— sorted(seq) – отсортированный список.

Для использования дополнительных действий применяется стандартная библиотека, представляющая собой набор модулей, поставляемых вместе с интерпретатором Python.

Операции над строками могут осуществляться при помощи следующих инструментов:

1. «Арифметические операции». Для строк, как и для чисел, определены операторы сложения и умножения.

2. Функция len(). С её помощью вычисляется длина строки.

3. Доступ по индексу. Позволяет обратиться к любому элементу строки по его номеру. Нумерация начинается с 0, первый элемент строки S имеет номер 0, последний – len(S)-1.

4. Срезы. Дают возможность скопировать либо использовать в выражениях часть строки. Оператор извлечения среза из строки имеет вид S[n1:n2], где n1 – индекс начала среза, n2 – индекс его окончания.

5. Оператор in. С его помощью можно узнать принадлежность подстроки к строке.

6. Функции min и max. max(s) определяет и выводит символ с наименьшим

кодом, min(s) – символ с наибольшим кодом.

Одним из элементов структурного программирования в Python является оператор ветвления if, представляющий собой выбор одной из двух либо более альтернатив в зависимости от условий [8]. После части if указывается логическое условие, которое может быть ложным либо истинным. Оператор ветвления может содержать другие операторы, такие как операторы дизъюнкции or, конъюнкции and, отрицания not, равенства == и неравенства !=.

Структуры с множественным выбором реализуются посредством оператора выбора switch-case [9]. Оператор switch осуществляет сравнение значения выражения, стоящего после него, со всеми метками, которые перечисляются посредством ключевых слов case. При совпадении значения выражения с какой-либо меткой управление передаётся следующему за этой меткой оператору.

Другим важным элементом структурного программирования в Python являются циклы, посредством которых организуется повторение выполнения определённых участков кода [10]. Используя циклы, можно компактно записать различные повторяющиеся действия. Выделяют следующие виды циклов:

1. Цикл обхода последовательности (for). Используется для перебора элементов последовательности. Для генерации индексов задействуется функция range().

2. Цикл с условием (while). Запускает выполнение инструкций, которое продолжается до момента истинности логического выражения. Опционально цикл while можно дополнить блоком else, выполнение инструкций в котором осуществляется по завершении цикла.

Python позволяет осуществлять сложные вычисления с использованием двумерных массивов, иначе называемых матрицами. Для объёмных вычислений с использованием массивов применятся библиотека NumPy, предоставляющая объект массива, который эффективнее подходит для математических вычислений по сравнению со стандартным списком Python [11].

Форма матрицы shape представляет собой совокупность её длин по

каждому измерению и задаётся в виде кортежа из нескольких чисел, в соответствии с размерностью. К двумерным массивам могут применяться операции sum, mean, max и другие, причём по умолчанию они применяются к матрице, как если бы она была линейным списком всех чисел, вне зависимости от её формы. Параметр axis позволяет применить операцию для указанной оси массива, причём axis=0 работает с отдельным столбцом, а axis=1 – с отдельной строкою. Для объединения массивов по начальному индексу используются функции hstack и vstack: hstack(), по последнему – vstack().

Python даёт возможность работать с ассоциативными массивами, представляющими собой последовательности значений с доступом по одному либо нескольким ключам [12]. В качестве ключей могут использоваться любые типы данных, однако для внутренней реализации массивов используются их хеши. Ассоциативные массивы используются преимущественно для сохранения истории состояний или событий.

Таким образом, средства программирования Python позволяют использовать этот язык для визуализации алгоритмов и различных структур данных, демонстрации рекурсии, синтаксических конструкций различной сложности и построения сетевых моделей. Средства данного языка программирования обеспечивают возможность эффективного составления сетевых графиков, связывающих события, происходящие в процессе выполнения определённого комплекса работ, что позволяет качественно планировать, организовывать, контролировать и управлять осуществляемыми работами.

Библиографический список:

1. Красочкин С.Г. Изображения и визуализация данных в Python // Научный журнал. – 2022. – № 2 (64). – С. 5-8.

2. Двоерядкина Н.Н. Расчёт временных параметров событий в сетевой модели // Вестник Амурского государственного университета. Серия: Естественные и экономические науки. – 2020. – № 89. – С. 22-27.

3. Будникова И.К., Приймак Е.В. Моделирование управляемых процессов с применением методов сетевого планирования // Вестник Технологического университета. – 2018. – Т. 21, № 1. – С. 115-118.

4. Гришков Д.Ю., Аусилова Н.М. Язык высокого уровня программирования Python // Наука и реальность. – 2022. – № 1 (9). – С. 114-117.

5. Бухаров Т.А., Нафикова А.Р., Мигранова Е.А. Обзор языка программирования Python и его библиотек // Colloquium-journal. – 2019. – № 3-1 (27). – С. 23-25.

6. Макаренко Л.Ф. Программирование на языке Python: учеб. пособие / Л.Ф. Макаренко, И.С. Шувалова. – М.: МАДИ, 2022. – 88 с.

7. Сысоева М.В., Сысоев И.В. Программирование для «нормальных» с нуля на языке Python: учебник. В двух частях. Часть 1 / отв. ред.: В.Л. Черный: – М.: Базальт СПО; МАКС Пресс, 2018. – 176 с.

8. Григорьев С.Г., Родионов М.А., Акимова И.В. Программирование на языке Python»: метод. пособие / под ред. С.Г. Григорьева. – М., 2021. – 123 с.

9. Гниденко И.Г. Технологии и методы программирования: учеб. пособие для вузов / И.Г. Гниденко, Ф.Ф. Павлов, Д.Ю. Федоров. – М.: Изд-во Юрайт, 2022. – 235 с.

10. Задорожный С.С., Фадеев Е.П. Объектно-ориентированное программирование на языке Python. – М.: Физический факультет МГУ им. М.В. Ломоносова, 2022. – 49 с.

11. Федоров Д.Ю. Программирование на языке высокого уровня Python: учеб. пособие для вузов / Д.Ю. Федоров. – 3-е изд., перераб. и доп. – М.: Изд-во Юрайт, 2022. – 210 с.

12. Кляус С.М. Инструменты динамической трассировки DTrace и SystemTap: метод. пособие. – 2020. – 220 с.