УДК 37.04:372.8
DOI

АНАЛИЗ ФАЙЛОВЫХ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON

[1]Козлов С.В. ORCID ID 0000-0001-9945-2098, [2]Быков А.А.

[1]Федеральное государственное бюджетное образовательное учреждение высшего образования «Смоленский государственный университет», Смоленск, Российская Федерация, e-mail: svkozlov1981@yandex.ru;

[2] Филиал Федерального государственного бюджетного образовательного учреждения высшего образования «Национальный исследовательский университет «Московский энергетический институт» в г. Смоленске, Смоленск, Российская Федерация

Цель исследования – рассмотреть особенности организации работы по анализу файловых данных с использованием языка программирования Python. Изучение программирования в школе является одной из центральных тем, которой отводится особое место ввиду формирования навыков алгоритмического мышления. В связи с этим обучающиеся должны уметь оперировать данными, представленными в разном виде. Это относится как к типу данных исследуемого объекта, так и к виду структуры хранения данных. При этом акцент следует делать на записи информации в текстовых файлах, как одном из основных способов ее хранения и передачи. Вместе с этим при чтении информации из файла ее целесообразно сохранять в списке, имеющем линейную структуру и многообразный функционал для исследования свойств его элементов. Вид элементов списка обуславливает выбор специальных инструментов для работы с ними, таких как целочисленное деление и остаток от деления при выполнении действий над целочисленными переменными. При этом можно применять общие методы обработки компонентов линейной структуры, например такие, как функции определения длины списка или позиции элементов. В статье описаны возможности применения этих подходов при анализе последовательности целых чисел, хранящихся в текстовом файле. Рассмотрены особенности чтения данных из файла, вычисления статистических характеристик списка полученных элементов и определения пар соседних элементов с заданными свойствами. Авторами охарактеризованы методические подходы решения такого рода заданий. Представлены результаты организованной исследовательской работы по анализу файловых данных с использованием языка программирования Python. Описаны комплексные знания о структурах данных и методах работы с информацией в текстовых файлах, которые обучающиеся получают в ходе исследования.

Ключевые слова: программирование, школа, профильные классы, анализ данных, текстовый файл, язык Python

FILE DATA ANALYSIS USING THE PYTHON PROGRAMMING LANGUAGE

[1]Kozlov S.V. ORCID ID 0000-0001-9945-2098, [2]Bykov A.A.

[1]Federal State Budgetary Educational Institution of Higher Education "Smolensk State University", Smolensk, Russian Federation, e-mail: svkozlov1981@yandex.ru;

[2]Branch of the Federal State Budgetary Educational Institution of Higher Education National Research University "Moscow Power Engineering Institute" in Smolensk, Smolensk, Russian Federation

The purpose of this study is to examine the specifics of organizing file data analysis using the Python programming language. Programming is a central topic in schools, particularly relevant for developing algorithmic thinking skills. The purpose of the study is to evaluate the effectiveness of explaining to students methods for analyzing file data using the Python programming language. The study of programming at school is one of the central topics, which is given special attention due to the formation of algorithmic thinking skills. In this regard, students should be able to operate with data presented in different forms. This applies both to the data type of the object under study and to the type of data storage structure. In this case, the emphasis should be on recording information in text files, as one of the main ways to store and transfer it. At the same time, when reading information from a file, it is advisable to save it in a list that has a linear structure and diverse functionality for studying the properties of its elements. The appearance of the list items determines the choice of special tools for working with them, such as integer division and the remainder of the division when performing actions on integer variables. You can use general methods for processing linear structure components, such as list length or item position functions. The article describes the possibilities of using these approaches when analyzing a sequence of integers stored in a text file. The features of reading data from a file, calculating statistical characteristics of the list of obtained elements and determining pairs of neighboring elements with given properties are considered. The authors describe methodological approaches to solving these types of assignments. They present the results of an organized research project on file data analysis using the Python programming language. They describe the comprehensive knowledge of data structures and methods for working with information in text files that students gain while their research.

Keywords: programming, school, profile classes, data analysis, text file, Python language

Введение

Дидактическая линия «Алгоритмизация и программирование» является сквозной в школьном курсе информатики и определяет будущие базовые способности обучающихся как IT-специалистов [1, 2]. Формирование у них базовых навыков программирования закладывается в средней школе. При обучении в 7–9-х классах они изучают базовые алгоритмические конструкции – следование, ветвление и циклы. При этом основным типом обрабатываемой в задачах информации являются числовые данные. Обучающиеся знакомятся с классическими алгоритмами их обработки, такими как вычисление натуральных делителей заданного числа, поиск простых чисел, перевод чисел из одной системы счисления в другую систему. При изучении записи условий они учатся оперировать логическим типом данных, пробуют составлять простые и сложносоставные логические условия [3, 4]. Знакомство со структурами данных «массив» и «список» позволяет сформировать у обучающихся умения обрабатывать наборы однотипных данных. Они должны продемонстрировать навыки использования ветвлений и циклов при обработке последовательностей чисел – записи их для хранения в единую наиболее подходящую структуру, поиске характеристик элементов, свойственных указанной выборке, вычислению статистических значений, таких как сумма, количество, максимальное, минимальное и среднее арифметическое значения.

При обучении в профильных физико-математических и IT-классах базу знаний о методах программирования необходимо расширять. Обучающиеся должны научиться обрабатывать данные, записанные в файлах. Они должны уметь оперировать информацией, представленной в них, уметь использовать базовые навыки программирования для анализа всей ее совокупности и формирования выборок элементов, удовлетворяющих указанным условиям. Чтение данных из файла, их обработка, вычисление характеристических свойств, запись новых файлов и модификация существующих должны стать у обучающихся в профильных классах по информатике базовым умением, характеризующим их IT-профпригодность.

В то же время нередко можно встретить ситуации, когда в профильном обучении программированию на уроках обучающиеся еще раз проходят базовый материал предпрофильных классов средней школы, не расширяя и не углубляя его. Они оттачивают свое мастерство практически на тех же наборах заданий, которые они решали ранее, так как не у всех из них сформированы базовые компетенции о способах и приемах программирования. Ввиду этого файлы, как структура данных, позволяют обобщить пройденный ранее учебный материал – закрепить алгоритмы обработки числовых типов данных в новой ситуации, а также овладеть новыми методами и подходами исследования информации, представленной в текстовом виде.

Цель исследования – рассмотрение особенностей организации работы по анализу файловых данных с использованием языка программирования Python.

Материал и методы исследования

Педагогический эксперимент по обучению анализу данных последовательностей, записанных в файлы, с использованием инструментов среды программирования PyCharm и языка программирования Python осуществлялся на базе Смоленского физико-математического лицея ЯВИР при МИФИ и МЭИ и средней школы № 6 г. Смоленска. Экспериментальное исследование проводилось в 10-х классах. В физико-математическом лицее в эксперименте участвовали 19 обучающихся, в IT-классе школы № 6 – 14 обучающихся. Педагогический эксперимент проводился три месяца, в нем принимали участие 33 обучающихся.

Педагогический эксперимент предполагал использование таких методов исследования, как анализ педагогической и методической литературы, обобщение существующего педагогического опыта, формирующий эксперимент и констатирующий эксперимент.

Результаты исследования и их обсуждение

Изучению программирования в старшей школе в профильных классах отводится существенное время. Сначала обучающиеся актуализируют свои знания по основам алгоритмизации, рассматривая на занятиях алгоритмические конструкции в приложении к заданиям практики. Так, например, они учатся составлять рекурсивные алгоритмы, изучают основы динамического программирования [5, 6]. Затем значительная часть занятий отводится на исследование возможностей разных встроенных библиотек изучаемой среды программирования. Например, программируя на языке Python в среде PyCharm, обучающиеся пробуют оперировать инструментами таких библиотек, как «sys», «math», «itertools», «functools» [7, 8]. После этого они изучают линейные и нелинейные структуры данных, такие как стек, очередь, графы, деревья,

записи, словари, информацию о которых можно хранить в файлах, а обрабатывать с помощью встроенных и пользовательских программных методов [9]. При этом отметим, что изучение методов работы с файлами имеет огромное значение для формирования профессиональных навыков будущего программиста. Это обусловлено тем, что при решении задач из разных областей знаний требуется обрабатывать информацию, записанную в файлах, и передавать ее обратно пользователю также в файловом виде.

Остановимся более подробно на вопросах обработки последовательностей чисел, записанных в файлах. Так, файловые данные могут иметь разную структуру и способ представления. Среди разных видов файлов наиболее распространены текстовые файлы. В них информация, относящаяся к разным типам данных, как числовым, логическим, так и собственно символьным, записана в виде текста, то есть относится к символьным типам. В связи с этим при чтении данных из файла с ними доступны методы обработки текстовых типов данных. Например, такие как функции вычисления длины текстовой строки, определения позиции символа в строке, выделения подстроки текста, методы поиска и замены одного вхождения подстроки в строку на другую текстовую подстроку. При этом, если необходимо обрабатывать данные методами, которые относятся к инструментам работы с другими типами данных, при считывании информации из файла ее переводят в другой тип. Так часто поступают с числовой информацией, представленной в символьном виде и разделенной в файле пробелами или специальными символами или размещенной по отдельным строкам. Ее при чтении из файла записывают в список, переводя каждое строковое значение в числовой тип. После этого все данные становятся числовыми, относящимися к целым или дробным значениям, и получают возможность обработки с помощью соответствующих методов. Можно оперировать арифметическими операциями над числами, выделять из них цифры с использованием методов целочисленного деления, переводить числа из одной системы счисления в другую систему [10], использовать рекурсивные методы [11] и алгебру логики [12]. Кроме того, при работе с данными, записанными в список, как с числовыми, так и с символьными значениями, можно использовать дополнительные методы работы с линейными итерируемыми объектами. Так, можно вычислить длину списка, определив тем самым количество элементов в нем, или узнать позиции всех четных элементов списка. При этом необходимо помнить, что все элементы в языке программирования Python пронумерованы с нуля.

Рассмотрим пример (составлен авторами). Текстовый файл primer.txt содержит последовательность целых чисел, значения которых могут варьироваться от -100000 до 100000 включительно. Определите количество пар соседних элементов последовательности, в которых хотя бы одно число оканчивается на 5 и делится на 3, при этом сумма элементов пары четная, а также минимальное число в парах такого вида.

Приведем программное решение задачи.

```
f = open('primer.txt')
a = [int(i) for i in f]
f.close()
k = 0
mn = 100000
for i in range(len(a) - 1):
    x, y = a[i], a[i + 1]
    if ((abs(x) % 10 == 6 and abs(x) % 3 == 0)
or (abs(y) % 10 == 6 and abs(y) % 3 == 0)) and
((x + y) % 2 == 0):
        k += 1
        mn = min(mn, x, y)
print(k, mn)
```

Охарактеризуем особенности представленного программного решения. Во-первых, необходимо текстовый файл *primer.txt* разместить в той же директории, что и файл проекта, который будет содержать данное программное решение. В этом случае при открытии файла можно будет указать только его краткое имя, не прописывая полный путь к нему. Это более удобный вариант, не требующий знания точного расположения файла на диске. Команда *f = open('primer.txt')* позволяет связать содержимое текстового файла с файловой переменной *f*, открыть файл на чтение данных и разместить считывающее устройство в его начале. Затем разместим считываемые из текстового файла данные в списке a = []. При этом для каждого строкового значения *i* из файла *f* переведем данные в целочисленный формат с помощью команды *int(i)*. После прочтения всех данных из файла требуется его закрыть, использовав для этого команду *f.close()*. Такой подход является универсальным способом чтения данных из файла и размещения их в список, содержащий элементы заданного типа для дальнейшего удобства их обработки в соответствии с условиями задачи. Его можно повторять во всех заданиях, требующих чтения данных из текстового файла и пере-

вода их в другой отличный от символьного формат данных.

Во-вторых, необходимо задать исходные значения поиска. Так, в приведенном задании переменной *k*, которая отвечает за количество пар соседних элементов последовательности чисел, отвечающих условиям отбора, следует присвоить начальное значение 0. Переменной mn необходимо присвоить наибольшее значение из указанного в задании числового диапазона или больше него, так как в ходе поиска минимального значения из элементов пары такое значение может только уменьшаться. При этом назвать переменную min в языке программирования Python нельзя, так как в среде PyCharm существует одноименный метод, отвечающий за поиск минимального значения среди итерируемых объектов [13, 14]. Также отметим, что в других заданиях такого типа может понадобиться дополнительно найти какое-то значение. Например, вычислить минимальный элемент последовательности чисел, который заканчивается на 3. Для этого можно выполнить команду:

```
mn3 = min([i for i in a if abs(i) % 10 == 3])
```

Обратим внимание, что выполнить аналогичные действия можно было и в развернутой структуре:

```
mn3 = 100000
for i in a:
    if abs(i) % 10 == 3:
        mn3 = min(mn3, i)
```

В этом случае она будет подобна центральному циклу в задаче, в котором осуществляется поиск количества пар элементов и минимального числа среди найденных них. Также заметим, что при вычислении минимального значения последовательности целых чисел, оканчивающегося на 3, необходимо значение элемента брать по модулю. Абсолютное значение элемента позволяет отбросить знак числа и вычислить его последнюю цифру, как результат определения остатка от деления числа на 10, так как остатки имеют положительные, а не отрицательные значения.

В-третьих, необходимо организовать выбор нужных пар элементов. Для этого служит цикл с параметром:

for i in range(len(a) – 1):.

В нем осуществляется перебор пар всех соседних элементов. Обратим внимание, что количество элементов в списке можно вычислить с помощью команды *len(a)*, а перебор выполнять до предпоследнего элемента списка, так как у последнего элемента уже нет пары. В цикле необходимо задать условие отбора, используя логические связки *and* и *or*, для интерпретации одновременного выполнения условий или хотя бы одного из них соответственно. При этом следует помнить про приоритет логических операций, *and* выполняется раньше *or*, если скобки не меняют порядок их действия. Также заметим, что при проверке на кратность суммы двум ее необходимо заключить в скобки, иначе сначала выполнится вычисление остатка от деления на 2 второго элемента пары, а затем оно прибавится к первому элементу последовательности и сумма сравнится с нулем. Кроме того, для удобства проверки правильности записи условий и уменьшения их громоздкости элементы пары были записаны в переменные *x* и *y*. Это повышает читабельность программного кода алгоритма. Внутри оператора *if* был использован метод min, который позволяет выбрать наименьшее значение из предыдущего наименьшего значения и текущих значений пары соседних элементов последовательности, удовлетворяющих условию задачи.

В-четвертых, остается вывести найденные значения *k* и *mn* на экран и записать полученный в задаче ответ. Для этого была использована команда *print()*, вывод параметров которой можно выполнять через запятую.

В экспериментальной работе такие задания от простого уровня к сложному предлагались участникам исследования. На базовом уровне предлагались задания, в которых было необходимо использовать только один цикл для обработки условий выбора пар соседних элементов. На продвинутом уровне сложности следовало сначала вычислить дополнительные характеристики списка элементов, такие как минимальный или максимальный элемент, удовлетворяющий указанным критериям делимости, например, на 3 или оканчивающийся на заданную цифру. На высоком уровне сложности уже было необходимо обрабатывать тройки соседних элементов с указанными свойствами. При этом также использовался дополнительный просмотр элементов списка на поиск критерия отбора для искомой тройки элементов.

Система заданий, а также обработка результатов экспериментальной работы была проведена с использованием программного комплекса «Advanced Tester». Он позволяет осуществлять подбор индивидуальных и групповых заданий в ходе обучения [15] в автоматизированном виде в соответствии с заданными параметрами обучения. Это открывает возможности оценки уровня

учебных достижений как отдельно взятого ученика, так и группы обучения в целом.

В ходе эксперимента обучаемые знакомились с подходами организации данных в файлах, с возможностями их обработки методами языка Python среды программирования PyCharm. Они отрабатывали умения и навыки чтения информации из файла, работы со списком как объектом, который содержит итерируемые элементы. Обучающиеся учились поиску элементов, их пар и троек, которые отвечают заданным в условии задания параметрам. В завершение экспериментальной деятельности обучающимся была предложена итоговая диагностическая работа, которая состояла из восьми заданий – двух базового и по три продвинутого и высокого уровней сложности. Ее результаты свидетельствуют о достижении обучающихся в СФМЛ ЯВИР при МИФИ и МЭИ и средней школе № 6 г. Смоленска высоких показателей усвоения учебного материала.

Так, высокий уровень учебных достижений продемонстрировали 14 чел., продвинутый – 14 чел. и базовый уровень – 5 чел. Такие данные свидетельствуют о нормальном законе распределения с плато в области высоких результатов обучения по программированию в профильных классах старшей школы. Гипотеза исследования состояла в том, что в результате специальным образом организованной работы по анализу файловых данных с использованием языка программирования Python обучающиеся получают комплексные знания о структурах данных, методах работы с информацией в текстовых файлах, совершенствуют приемы обработки элементов списков, расширяют практику применения инструментов программных библиотек и большинство из них способны к решению задач высокого и продвинутого уровней сложностей.

Качественный анализ условий и результатов эксперимента. Анализ данных экспериментальной работы позволяет сделать следующие выводы. На формирующем этапе педагогического эксперимента обучаемые усвоили подходы модели обработки информации из текстового файла. Они изучили методы обработки данных числовых последовательностей. На практике проверили полученные знания в заданиях поиска элементов, которые отвечают простейшим условиям. При этом они развивали и пополняли багаж своих знаний об инструментах языка Python среды программирования PyCharm, используемых для работы с объектами символьного и числового типа. Кроме того, они совершенствовали навыки применения базовых алгоритмических конструкций и структур данных. На констатирующем этапе педагогического эксперимента обучающиеся выполняли задания повышенного и высокого уровней сложности, соответствующих показателям сформированности учебных действий, необходимых для государственной итоговой аттестации. При составлении заданий для решения были учтены типичные ошибки, которые совершают экзаменуемые на ЕГЭ по информатике при решении заданий данного типа. Это позволило выработать у обучаемых понимание логической последовательности отбора элементов, удовлетворяющих заданным условиям, правильной обработки результатов делимости, специфики определения последней цифры числа для отрицательных чисел, поиска минимального и максимального значений элементов с дополнительными свойствами. Кроме того, были изучены конструкции языка программирования Python, которые позволяют «сворачивать» алгоритмические конструкции обработки итерируемых объектов в запись команды в одну строку, использовать при необходимости и целесообразности «лямбда-функции». Это позволяет говорить о том, что гипотеза исследования находит свое подтверждение в практике проведения педагогического эксперимента.

Заключение

Таким образом, работа в ходе исследования показывает целесообразность использования языка Python среды программирования PyCharm при анализе данных числовых последовательностей. Обучающиеся получают комплексные знания о структурах данных, методах работы с информацией в текстовых файлах, совершенствуют приемы обработки элементов списков, расширяют практику применения инструментов программных библиотек. Это позволяет сформировать устойчивый профессиональный интерес к программированию у обучающихся профильных классов для подготовки их как будущих IT-специалистов.

Список литературы

1. Козлов С.В., Быков А.А. О применении методов математического моделирования при обучении алгоритмизации в вузе // Современные проблемы науки и образования. 2021. № 3. С. 92. DOI: 10.17513/spno.30946.

2. Кормилицына Т.В. Формирование цифровых компетенций и навыков в педагогическом образовании как современный тренд // Гуманитарные науки и образование. 2021. Т. 12. № 1 (45). С. 42–48. DOI: 10.51609/2079-3499_2021_12_01_42.

3. Корчажкина О.М. Вербально-визуальный метод при обучении булевой алгебре в курсе информатики для старшей школы // Вестник МГПУ. Серия: Информатика и информатизация образования. 2020. № 4 (54). С. 16–26. DOI: 10.25688/2072-9014.2020.54.4.02.

4. Козлов С.В., Быков А.А. Обучение анализу логических выражений с использованием языка программирования Python // Современные наукоемкие технологии. 2025. № 4. С. 121–126. DOI: 10.17513/snt.40375.

5. Попов В.С. Эмпирическое исследование времени выполнения рекурсивных функций и развитие STEM-компетенций // Информатика в школе. 2024. Т. 23. № 3. С. 5–14. DOI: 10.32517/2221-1993-2024-23-3-5-14.

6. Козлов С.В., Быков А.А. Обучение школьников выполнению рекурсивных алгоритмов с использованием систем программирования // Современные наукоемкие технологии. 2023. № 5. С. 49–54. DOI: 10.17513/snt.39616.

7. Каракозов С.Д., Маняхина В.Г. Python как базовый язык обучения программированию в школе // Информатика в школе. 2020. № 1 (154). С. 26–30. DOI: 10.32517/2221-1993-2020-19-1-26-30.

8. Маркелов В.К., Завьялова О.А. Язык программирования Python как альтернативный инструмент для решения заданий ЕГЭ по информатике // Информатика в школе. 2023. № 2 (181). С. 63–72. DOI: 10.32517/2221-1993-2023-22-2-63-72.

9. Родыгин Е.Ф. Методические рекомендации обучения программированию в школе // Вестник Марийского государственного университета. 2011. № 7. С. 20–22. EDN: RDCWEL.

10. Козлов С.В., Быков А.А. Обучение анализу записи чисел и проверке их делимости в языке программирования Python в школе // Современные наукоемкие технологии. 2024. № 7. С. 157–162. DOI: 10.17513/snt.40101.

11. Попов В.С., Абросимова-Романова Л.А. Реализация компетентностного подхода при решении заданий на рекурсию // Вестник Тверского государственного университета. Серия: Педагогика и психология. 2025. № 2 (71). С. 199–209. DOI: 10.26456/vtpsyped/2025.2.199.

12. Колобов А.Н. Математическая логика на современном этапе образования // Мир науки, культуры, образования. 2024. № 4 (107). С. 236–238. DOI: 10.24412/1991-5497-2024-4107-236-238.

13. Панова И.В., Коливнык А.А. Методические аспекты обучения программированию на языке Python в школьном курсе информатики // Информатика в школе. 2020. № 6. С. 47–50. DOI: 10.32517/2221-1993-2020-19-6-47-50.

14. Кондратьева В.А. Обучение основам программирования на языке Python в школьном курсе информатики // Вестник МГПУ. Серия: Информатика и информатизация образования. 2021. № 1 (55). С. 8–16. DOI: 10.25688/2072-9014.2021.55.1.01.

15. Киселева О.М. Использование математических методов для формализации элементов образовательного процесса // Концепт. 2013. № 2. С. 51–57. EDN: PVXPOV.

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare that there is no conflict of interest.