УДК 004.043

ИСПОЛЬЗОВАНИЕ МАССИВОВ В ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ

Струповец И.А.

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники» филиал «Минский радиотехнический колледж», г. Минск, Республика Беларусь

Научный руководитель: Шумчик Ф.С. – заместитель директора по учебной работе, преподаватель высшей категории, канд. филол. наук, доцент

Аннотация. Обладая способностью эффективно хранить и обрабатывать большие объемы данных, массивы стали фундаментальной концепцией программирования, они необходимы для понимания алгоритма работы с большими объёмами информации.

Ключевые слова: массив, элементы, индекс, переменная, структура данных.

Введение. Массивы – это фундаментальная структура данных в программировании, позволяющая эффективно хранить наборы связанных данных и управлять ими. Понимание того, как работают массивы и как их можно эффективно использовать, важно для любого программиста. Рассмотрим использование массивов в образовательном процессе колледжа.

Основная часть. В программировании массивы могут быть одномерными или двумерными, это зависит от количества индексов, необходимых для доступа к его элементам. Одномерный массив, также известный как вектор, представляет собой список элементов одного типа данных, расположенных в линейной последовательности. Доступ к нему можно получить с помощью одного индекса или нижнего индекса, который представляет позицию элемента в массиве. Одномерные массивы обычно используются для хранения списков данных, к примеру, список имен, список чисел и т.д. [1].

Одномерный массив выглядит так:

a = [1, 2, 3, 4, 5, 6].

Но как он используются?

Разберём на примере электронного журнала. У нас есть ученик, и нам нужно посчитать его средний балл. Его оценки – это элементы массива, значит, чтобы посчитать средний балл, нужно сложить оценки и поделить на их количество. Выглядеть это будет так:

```
Student = [9, 9, 8, 9, 7, 9, 9, 10]
Sr = len(Student)
Sum = 0
for i in range(len(Student)):
Sum = Sum + Student [i]
Mark = Sum/Sr print(Mark)
```

Окно вывода

8.75.

Массивы можно использовать и для других целей в рамках образовательного процесса колледжа. Например, отслеживание посещаемости учащимися: массив можно использовать для отслеживания посещаемости учащимися занятий. Каждый элемент массива будет представлять день семестра, и массив можно использовать для подсчета количества дней, которые посещал учащийся, или для определения часто отсутствующих учащихся:

```
attendance = [] students = ['Ира', 'Стас', 'Влад', 'Федя', 'Яна']
for student in students:
status = int(input(f"{student}: Enter 1 if present or 0 if absent: "))
attendance.append(status) print("Attendance: ", attendance)
```

Окно вывода
Ира: Введите 1 если присутствует или 0 если отсутствует: 1
Стас: Введите 1 если присутствует или 0 если отсутствует: 1
Влад: Введите 1 если присутствует или 0 если отсутствует: 0
Федя: Введите 1 если присутствует или 0 если отсутствует: 0
Яна: Введите 1 если присутствует или 0 если отсутствует: 1
Посещаемость: [1, 1, 0, 0, 1]

Создав базу данных учащихся, можно под конец месяца подсчитывать количество пропусков или узнать, присутствовал ли человек в этот день.

Массивы можно использовать для учета финансовой деятельности колледжа, к примеру, определения доходов или расходов на обучение. Каждый элемент массива будет представлять конкретную финансовую транзакцию, а массив можно использовать для расчета общего дохода или расходов за определенный период времени:

```
expenses = []
num_categories = 5
for category in range(num_categories):
amount = float
(input(f"Введите сумму для категории расходов {category+1}:")
expenses.append(amount) total_expenses = sum(expenses)
revenue = []
num_categories = 3 for category in range(num_categories):
amount = float(input(f"Введите сумму для категории дохода {category+1}: "))
revenue.append(amount)
total_revenue = sum(revenue) net_income = total_revenue – total expenses
print("Общие расходы: ", total_expenses)
print("Общий доход: ", total_revenue)
print("Чистый доход: ", net_income)
```

Окно вывода
Введите сумму для категории расходов 1: 100
Введите сумму для категории расходов 2: 200
Введите сумму для категории расходов 3: 300
Введите сумму для категории расходов 4: 400
Введите сумму для категории расходов 5: 500
Введите сумму для категории дохода 1: 1000
Введите сумму для категории дохода 2: 2000
Введите сумму для категории дохода 3: 3000
Общие расходы: 5100.0
Общий доход: 6000.0
Чистый доход: 900.0

Двумерный массив, также известный как матрица, представляет собой набор элементов, расположенных в строках и столбцах. Доступ к нему можно получить с помощью двух индексов или нижних индексов, которые представляют позицию строки и столбца элемента в массиве. Двумерные массивы обычно используются для представления сеток данных, например, электронных таблиц или изображений. Выглядит это так:

a = [[1, 2, 3][4, 5, 6][7, 8, 9]].

Гораздо проще представить двумерный массив в виде таблицы. Чтобы выбрать нужный элемент, нужно ввести столбец и строку.

1 2 3
1.[1, 2, 3]
2.[4, 5, 6]
3.[7, 8, 9]

Теперь, чтобы вывести элемент 5, нужно написать:

```
print(a[2][2])
```

Например, при обработке изображений двумерный массив можно использовать для представления пикселей изображения, где каждый пиксель представлен индексами строки и столбца, а также значением цвета или интенсивности.

Массив можно использовать для хранения расписания пар, предлагаемых в течение семестра. Каждый элемент массива будет представлять дату и определенную пару, и массив можно использовать для определения того, какие пары предлагаются в определенный день:

```
couples_schedule = []
print('Сколько рабочих дней?')
num_days = int(input())
num_slots = 4
for day in range(num_days):
day_schedule = []
for time_slot in range(num_slots):
couple_name = input(f"Введите имя пары на день {day+1} номер пары {time_slot+1}: ")
day_schedule.append(couple_name)
couples_schedule.append(day_schedule)
print("Расписание пар: ", couples_schedule)
print('В какой день вас интересуют пары?')
num_day = int(input())
print('Расписание на день ', couples_schedule[num_day-1])
```

Окно вывода

```
Сколько рабочих дней?
2
Введите имя пары на день 1 номер пары 1: русский
Введите имя пары на день 1 номер пары 2: английский
Введите имя пары на день 1 номер пары 3: математика
Введите имя пары на день 1 номер пары 4: черчение
Введите имя пары на день 2 номер пары 1: информатика
Введите имя пары на день 2 номер пары 2: физра
Введите имя пары на день 2 номер пары 3: физика
Введите имя пары на день 2 номер пары 4: география
Расписание пар: [['русский', 'английский', 'математика', 'черчение'], ['информатика', 'физра', 'физика', 'география']]
В какой день вас интересуют пары?
2
Расписание на день ['информатика', 'физра', 'физика', 'география']
```

Также массивы используются для фотошопа. Рассмотрим черно-белое изображение шириной 10 пикселей и высотой 5 пикселей. Двумерный массив можно использовать для представления данного изображения следующим образом:

```
int image[5][10] = {
{1, 0, 1, 1, 0, 1, 0, 0, 1, 0},
{0, 1, 1, 0, 1, 0, 1, 1, 0, 1},
{1, 0, 0, 1, 1, 0, 1, 0, 1, 1},
{0, 1, 0, 0, 1, 1, 0, 1, 1, 0},
{1, 0, 1, 1, 0, 0, 1, 0, 1, 1}
};
```

В этом примере каждый элемент массива равен 0 или 1, что соответствует цвету соответствующего пикселя: 0 представляет черный пиксель, а 1 представляет белый пиксель.

Двумерные массивы обычно используются для представления изображений в компьютерной графике и приложениях для обработки изображений. Представляя изображения в виде массивов, становится проще манипулировать ими и обрабатывать их с помощью алгоритмов и программного обеспечения.

На практике алгоритмы обработки изображений очень сложные, включая такие операции, как фильтрация, изменение размера и обнаружение признаков. Однако основная идея та же: изображения представлены в виде массивов пикселей, и их обработка включает в себя доступ к каждому элементу массива и манипулирование им [2].

Заключение. Таким образом, массивы являются важной структурой данных, используемой в различных областях обучения и языках программирования. Одномерные массивы могут хранить ряд данных, к примеру, информацию об успеваемости учащихся, а двумерные массивы полезны для представления сложных структур: сеток данных, электронных таблиц или изображений. Их можно использовать для анализа данных, научных исследований и управления базами данных. Обладая способностью эффективно хранить и обрабатывать большие объемы данных, массивы стали фундаментальной концепцией программирования, они необходимы для понимания алгоритма работы с большими объёмами информации.

Список литературы

1. Информатика : учеб. пособие. для 10 класса / В.М. Котов [и др.] – Минск: Народная асвета, 2020 – 121с.;

2. Многомерные массивы [Электронный ресурс]. – Электронные данные. – Режим доступа: https://apmath-spbu.github.io/basics/lecture3/multiplearrays/. Дата доступа: 07.03.2023.

UDC 004.043

USING ARRAYS IN THE EDUCATIONAL PROCESS

Strupovets I.A.

Belarusian State University of Informatics and Radioelectronics affiliate
Minsk Radioengineering College, Minsk, Republic of Belarus

Shumchik F.S. – Deputy Director for Academic Affairs, teacher of the highest category, PhD, associate professor

Annotation. With the ability to efficiently store and process large amounts of data, arrays have become a fundamental programming concept, they are necessary to understand the algorithm for working with large amounts of information.

Keywords: array, elements, index, variable, data structure.