

САПР БИОЛОГИЧЕСКИХ ОБЪЕКТОВ

(программный комплекс)

В.А.Сапрыкин

Новый Международный Университет Виртуальных Компьютерных систем

mveo@yandex.ru

(09.01.26)

Abstract

Part1.py

This script allows you to create and visualize a three-dimensional data array. You can either enter the array values manually or generate a random array. Below are the steps for using the program.

Part2.py

This script allows you to create and visualize a three-dimensional data array by applying instructions from a file. The instructions define which cubes to modify and when. Below are the steps for using the program.

Parts 3, 4

newPart4.py

After the array is filled, the program starts propagating a wave from a starting point. The program saves all intermediate states of the array during wave propagation for subsequent animation.

Parts 5, 6

The program is an interactive application that allows the user to create a three-dimensional array (body), fill it with values via a graphical interface, propagate a wave from a specified starting point, and visualize the filling and wave propagation process using animation. The program also records information about the wave propagation process in a "DNA" format for further analysis.

Часть 1

Инструкция по эксплуатации системы проектирования J-Объектов

Part1.py

Этот скрипт позволяет создать и визуализировать трёхмерный массив данных. Вы можете либо ввести значения массива самостоятельно, либо сгенерировать случайный массив. Ниже приведены шаги для использования программы.

Шаг 1: Запуск программы

1. Запустите программу, выполнив скрипт на вашей системе
python part1.py

Шаг 2: Выбор режима ввода данных

После запуска программы вы увидите следующий вопрос:

«Хотите ввести массив самостоятельно? (да/нет):»

- Ввод самостоятельно: Введите `да`, если хотите ввести данные массива самостоятельно.
- Случайный массив: Введите `нет`, если хотите сгенерировать случайный массив.

Шаг 3: Ввод размеров массива (если выбран режим случайного массива)

Если вы выбрали создание случайного массива, вам нужно будет ввести размеры массива:

1. Глубина (z): Введите число от 1 до 10, представляющее глубину массива.
2. Строки (y): Введите число от 1 до 10, представляющее количество строк массива.
3. Столбцы (x): Введите число от 1 до 10, представляющее количество столбцов массива.

Шаг 4: Ввод значений массива (если выбран режим самостоятельного ввода)

Если вы выбрали самостоятельный ввод, вам нужно будет последовательно вводить значения для каждого элемента массива. Программа запросит значение для каждого элемента по его координатам (глубина, строка, столбец).

Шаг 5: Визуализация массива

После ввода всех данных или генерации случайного массива программа автоматически создаст и отобразит 3D-диаграмму массива. Каждый ненулевой элемент массива будет представлен как кубик с цветом, соответствующим значению элемента.

Дополнительные замечания

- Цветовая шкала: Значения от 0 до 10 представлены различными цветами на диаграмме.
- Отсутствие осей: На диаграмме оси отсутствуют для лучшей визуализации структуры массива.

Пример работы программы

1. Выберите режим ввода: `да`
2. Введите размеры массива:
Введите глубину (z) от 1 до 10: 2
Введите количество строк (y) от 1 до 10: 2
Введите количество столбцов (x) от 1 до 10: 2
3. Введите значения для элементов:
Введите значение для элемента (0, 0, 0) от 0 до 10: 1
Введите значение для элемента (0, 0, 1) от 0 до 10: 2
Введите значение для элемента (0, 1, 0) от 0 до 10: 3
Введите значение для элемента (0, 1, 1) от 0 до 10: 4
Введите значение для элемента (1, 0, 0) от 0 до 10: 5
Введите значение для элемента (1, 0, 1) от 0 до 10: 6
Введите значение для элемента (1, 1, 0) от 0 до 10: 7
Введите значение для элемента (1, 1, 1) от 0 до 10: 8
4. Откроется окно с 3D-диаграммой массива.

Часть 2.

Part2.py

Этот скрипт позволяет вам создать и визуализировать трёхмерный массив данных, применяя инструкции из файла. Инструкции определяют, какие кубики изменять и когда. Ниже приведены шаги для использования программы.

Шаг 1: Подготовка файла с инструкциями

1. Создайте текстовый файл (например, `instructions.txt`).
2. В каждой строке файла добавьте инструкцию в формате:
`from_index to_index tick color`
где:
- `from\_index` — координаты исходного кубика (например, `0,0,0`),
- `to\_index` — координаты целевого кубика (например, `1,0,0`),
- `tick` — время выполнения инструкции (целое число),
- `color` — цвет целевого кубика (целое число от 0 до 10).

Пример содержимого файла «instructions.txt»:

```
0,0,0 1,0,0 1 5
1,0,0 2,0,0 2 3
0,0,0 0,1,0 1 2
0,1,0 0,2,0 2 7
```

Шаг 2: Запуск программы

1. Запустите программу, выполнив скрипт на вашей системе.
python part2.py

Шаг 3: Ввод имени файла с инструкциями

После запуска программы вы увидите следующий вопрос:

«Введите имя файла с инструкциями:»

Введите имя файла, содержащего инструкции, например:

«instructions.txt»

Шаг 4: Ввод начальных координат кубика

Программа запросит начальные координаты кубика:

«Введите начальные координаты кубика (например, 0,0,0):»

Введите, например 0,0,0

Шаг 5: Ввод цвета начального кубика

Программа запросит цвет начального кубика:

«Введите цвет начального кубика (от 0 до 10):»

Введите цвет начального кубика, например: 1

Шаг 6: Анимация массива

После ввода начальных данных программа автоматически прочитает инструкции из файла и начнёт анимацию изменения массива. Каждый кубик будет изменяться в соответствии с инструкциями на каждом шаге времени ('tick').

Дополнительные замечания

- Цветовая шкала: Значения от 0 до 10 представлены различными цветами на диаграмме.
- Отсутствие осей: На диаграмме оси отсутствуют для лучшей визуализации структуры массива.
- Паузы между тактами: Между каждым шагом времени ('tick') происходит пауза в 1 секунду. Вы можете изменить длительность паузы, изменив значение `time.sleep(1)` в коде.

Пример работы программы

1. Создайте файл `instructions.txt` с следующим содержимым:

0,0,0 1,0,0 1 5

1,0,0 2,0,0 2 3

0,0,0 0,1,0 1 2

0,1,0 0,2,0 2 7

2. Запустите программу и введите имя файла:

Введите имя файла с инструкциями: `instructions.txt`

3. Введите начальные координаты кубика:

Введите начальные координаты кубика (например, 0,0,0): 0,0,0

4. Введите цвет начального кубика:

Введите цвет начального кубика (от 0 до 10): 1

5. Откроется окно с анимированной 3D-диаграммой массива, показывающей изменения в соответствии с инструкциями.

Часть 3,4

newPart4.py

Шаг 1: Запуск программы

1. Запустите программу, выполнив скрипт на вашей системе.
python newPart4.py

Шаг 2: Ввод размеров массива

После запуска программы вы увидите несколько запросов для ввода размеров массива и параметров распространения волны:

1. Размер n: Введите целое число от 1 до 10, представляющее размер по оси X.
2. Размер m: Введите целое число от 1 до 10, представляющее размер по оси Y.
3. Размер k: Введите целое число от 1 до 10, представляющее размер по оси Z.
4. Размер d: Введите целое число, представляющее максимальное расстояние распространения волны.
5. Значение t: Введите целое число, представляющее параметр t, используемый в записи ДНК.

Шаг 3: Ввод начальной точки роста

Программа запросит начальную точку роста:

«Введите начальную точку роста (x y z):»

Введите координаты начальной точки в формате `x y z`, например: 1 1 1

Программа проверит, что введенные координаты находятся внутри массива. Если координаты некорректны, программа сообщит об ошибке и запросит новые координаты.

Шаг 4: Заполнение массива

Программа запросит значения для каждого элемента массива последовательно:

«Введите значения для каждого элемента массива (всего 27 элементов):»

Введите значение для элемента (0, 0, 0):»

Введите целые числа от 0 до 10 для каждого элемента массива. Программа сохраняет все промежуточные состояния массива для последующей анимации.

Шаг 5: Распространение волны

После заполнения массива программа начнет распространение волны от начальной точки. Программа сохраняет все промежуточные состояния массива во время распространения волны для последующей анимации.

Шаг 6: Ввод интервала времени для анимации

Программа запросит интервал времени для отрисовки объекта:

«Введите интервал времени для отрисовки объекта:»

Введите целое число, представляющее интервал времени в миллисекундах между кадрами анимации.

Пример ввода:

«Введите интервал времени для отрисовки объекта: 500»

(данные вводятся в значении миллисекунд, 1000 = 1 секунда)

Шаг 7: Анимация массива

После ввода интервала времени программа начнет анимацию изменения массива. Анимация покажет процесс заполнения массива и последующее распространение волны от начальной точки.

Дополнительные замечания

- Цветовая шкала: Значения от 0 до 10 представлены различными цветами на диаграмме.
- Отображение осей: На диаграмме отображаются оси X, Y и Z для лучшей ориентации.

Пример работы программы

1. Запустите программу и введите размеры массива и параметры:

Введите размер n: 3

Введите размер m: 3

Введите размер k: 3

Введите размер d: 2

Введите значение t: 5

2. Введите начальную точку роста:

Введите начальную точку роста (x y z): 0 0 0

3. Введите значения для каждого элемента массива:

Введите значения для каждого элемента массива (всего 27 элементов):

Введите значение для элемента (0, 0, 0): 1

Введите значение для элемента (0, 0, 1): 2

4. Введите интервал времени для анимации:

Введите интервал времени для отрисовки объекта: 500

5. Откроется окно с анимированной 3D-диаграммой массива, показывающей процесс заполнения массива и распространение волны.

Часть 5,6

Документация к программе

Описание программы

Программа представляет собой интерактивное приложение, которое позволяет пользователю создавать трехмерный массив (тело), заполнять его значениями через графический интерфейс, распространять волну от заданной начальной точки и визуализировать процесс заполнения и распространения волны с использованием

анимации. Программа также записывает информацию о процессе распространения волны в формате "ДНК" для дальнейшего анализа.

Структура программы

Программа состоит из нескольких функций, каждая из которых выполняет определенную задачу. Ниже приведено описание каждой функции и их параметров.

1. `create\_body(n, m, k)`

****Описание**:**

Создает два трехмерных массива: один для хранения значений тела (`body`) и другой для хранения состояний (`states`).

****Параметры**:**

- `n` (int): Размер массива по оси X.
- `m` (int): Размер массива по оси Y.
- `k` (int): Размер массива по оси Z.

****Возвращаемое значение**:**

Кортеж из двух трехмерных массивов: `(body, states)`.

2. `fill\_body\_with\_entries(body, entry\_widgets, n, m, k, dna\_text)`

****Описание**:**

Заполняет массив `body` значениями, введенными пользователем через виджеты `Entry`. Значения записываются в текстовое поле `dna\_text`.

****Параметры**:**

- `body` (numpy.ndarray): Массив, который будет заполняться.
- `entry\_widgets` (list): Трехмерный список виджетов `Entry`, используемых для ввода значений.
- `n` (int): Размер массива по оси X.
- `m` (int): Размер массива по оси Y.
- `k` (int): Размер массива по оси Z.
- `dna\_text` (tk.Text): Текстовое поле для вывода информации.

****Возвращаемое значение**:**

Список состояний массива во время заполнения (`fill\_states`).

3. `propagate\_wave(body, start\_point, t, d\_max, dna\_text)`

****Описание**:**

Распространяет волну от заданной начальной точки в массиве `body`. Записывает информацию о процессе распространения волны в формате "ДНК".

****Параметры**:**

- `body` (numpy.ndarray): Массив, в котором распространяется волна.
- `start\_point` (tuple): Начальная точка распространения волны в формате `(x, y, z)`.
- `t` (int): Значение `t`, используемое для записи в ДНК.
- `d\_max` (int): Максимальное расстояние распространения волны.
- `dna\_text` (tk.Text): Текстовое поле для вывода информации.

****Возвращаемое значение**:**

Кортеж из трех элементов:

1. Обновленный массив `body`.
2. Список записей ДНК.
3. Список состояний массива во время распространения волны (`wave\_states`).

4. `visualize\_body(fill\_states, wave\_states, start\_point, dna\_text)`

****Описание**:**

Визуализирует процесс заполнения массива и распространения волны с использованием анимации.

****Параметры**:**

- `fill\_states` (list): Список состояний массива во время заполнения.
- `wave\_states` (list): Список состояний массива во время распространения волны.
- `start\_point` (tuple): Начальная точка распространения волны в формате `(x, y, z)`.
- `dna\_text` (tk.Text): Текстовое поле для вывода информации.

****Возвращаемое значение**:**

Объект анимации (`ani`).

5. `main()`

****Описание**:**

Основная функция программы, которая инициализирует графический интерфейс, запрашивает у пользователя входные данные, вызывает функции для заполнения массива, распространения волны и визуализации.

Графический интерфейс

Программа использует библиотеку `tkinter` для создания графического интерфейса. Интерфейс включает следующие элементы:

1. ****Поля ввода размерностей массива**:**

Пользователь вводит размеры массива (`n`, `m`, `k`), начальную точку роста (`x`, `y`, `z`), максимальное расстояние распространения волны (`d`) и значение `t`.

2. ****Матрица виджетов Entry**:**

Трехмерная матрица виджетов `Entry`, где пользователь вводит значения для заполнения массива.

3. ****Кнопка "Заполнить и распространить волну"**:**

При нажатии на кнопку программа заполняет массив значениями, распространяет волну и визуализирует процесс.

4. ****Текстовое поле для вывода ДНК**:**

Выводит информацию о процессе распространения волны в формате "ДНК".

Визуализация

Программа использует библиотеку `matplotlib` для визуализации массива и процесса распространения волны. Анимация показывает:

1. Процесс заполнения массива.
2. Процесс распространения волны.

Цвета элементов массива зависят от их значений и определяются с помощью цветовой карты `viridis`.

Пример использования

1. Запустите программу.
2. Введите размеры массива (`n`, `m`, `k`), начальную точку роста (`x`, `y`, `z`), максимальное расстояние распространения волны (`d`) и значение `t`.
3. Заполните значения массива через виджеты `Entry`.
4. Нажмите кнопку "Заполнить и распространить волну".
5. Наблюдайте за анимацией заполнения массива и распространения волны.
6. Просмотрите информацию о процессе распространения волны в текстовом поле "Массив ДНК".

Пример ввода ВК-ДНК программ роста объектов моделирования:

1. Ручной ввод назначения клетки (цвет) в трехмерном ДНК-массиве объекта
- Хотите ввести массив самостоятельно? (да/нет): да

Введите глубину (z) от 1 до 10: 3
 Введите количество строк (y) от 1 до 10: 3
 Введите количество столбцов (x) от 1 до 10: 3
 Введите значение для элемента (0, 0, 0) от 0 до 10: 0
 Введите значение для элемента (0, 0, 1) от 0 до 10: 0
 Введите значение для элемента (0, 0, 2) от 0 до 10: 0
 Введите значение для элемента (0, 1, 0) от 0 до 10: 0
 Введите значение для элемента (0, 1, 1) от 0 до 10: 1
 Введите значение для элемента (0, 1, 2) от 0 до 10: 0

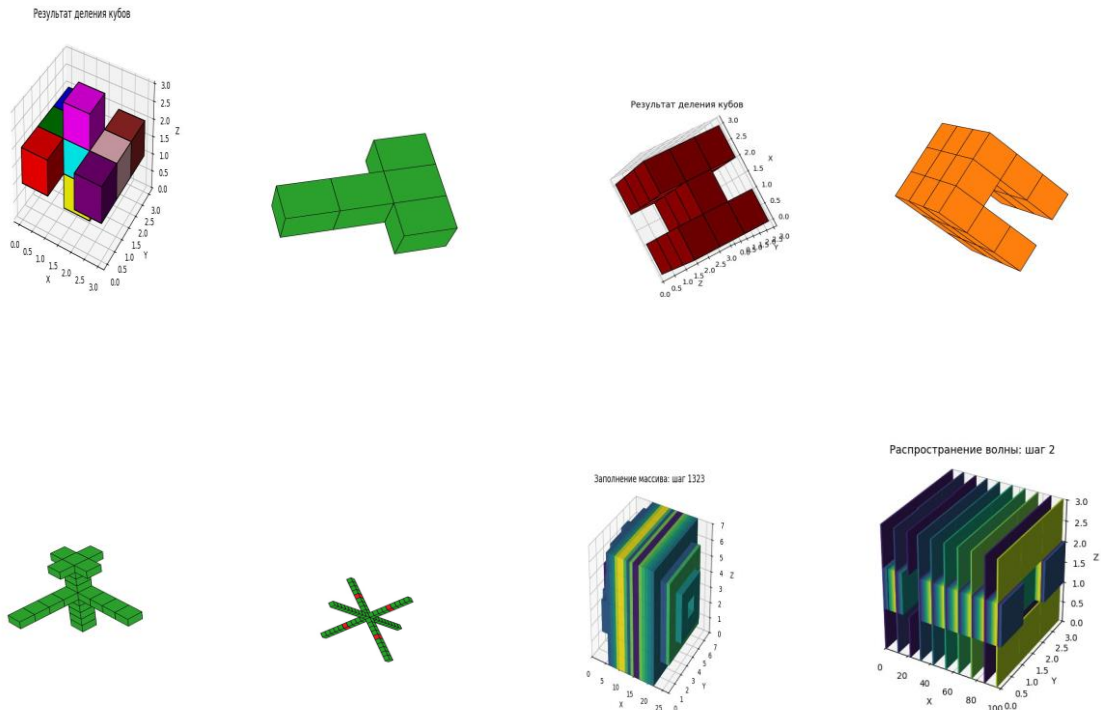
2.ВК-ФЗ (формула жизни): № по порядку, номер клетки которая делится, номер клетки после деления, направление деления, время деления, свойства клетки (цвет)

3.В виде: x y z –клетки которая делиться, x y z –новая клетка после деления, время деления, свойства (цвет)

В этом варианте направление деления задается координатами клеток:

0,5,5 0,5,5 0 7
 0,5,5 1,5,5 1 2
 1,5,5 2,5,5 2 2
 2,5,5 3,5,5 3 9
 3,5,5 4,5,5 4 2
 4,5,5 5,5,5 5 2
 5,5,5 6,5,5 6 3
 6,5,5 7,5,5 7 2
 7,5,5 8,5,5 8 8
 8,5,5 9,5,5 9 1

Скриншоты образцов моделирования:



Источники информации:

1. <https://vks-ai.ru/>
2. <http://vkcdnk.ru/>
3. https://vks-vk.ucoz.ru/index/nkhu_novyj_khristianskij_universitet/0-227

