

ДОМАШНЕЕ ЗАДАНИЕ ПО PYTHON №4.

Напишите программу, моделирующую экологическую систему океана, в котором обитают хищники и жертвы (добыча). Океан представляется двумерным массивом ячеек. В ячейке может находиться либо хищник, либо добыча, либо препятствие; также ячейка может быть пустой. В каждый квант времени ячейки последовательно обрабатываются. Хищник может съесть соседнюю добычу или просто переместиться на свободную клетку; добыча также может переместиться на свободную клетку. Если в течение определенного времени хищник ничего не съел, то он погибает. Через определенные интервалы времени добыча и хищники размножаются (если рядом есть свободная клетка). При этом потомок занимает эту свободную клетку. Моделирование заканчивается либо когда пройдет определенное число итераций, либо когда погибнут все хищники или вся добыча.

Проверьте на этой модели гипотезу о цикличности популяций хищников и жертв (скорее всего она должна выполняться). Для этого необходимо построить графики зависимости количества хищников и жертв от времени и проверить, становится ли этот график периодическим (но не константным) с какого-то момента. Предусмотрите также возможность просмотра процесса моделирования в консольном режиме для океана небольшого размера.

Параметры моделирования должны храниться в отдельном файле. В параметры включаются:

- Размер океана и его начальное состояние. Допускается сделать отдельную опцию, которая будет говорить, что начальное состояние нужно сгенерировать случайно, но тогда в качестве параметров должны задаваться вероятности различных клеток.
- Интервалы жизни без еды и интервалы размножения для хищников.
- Интервалы размножения для жертв.
- При одинаковых входных параметрах должны получаться одинаковые результаты (чтобы легче было тестировать и отлаживать и чтобы не потерять интересные результаты). Этого можно добиться, если использовать не обычный генератор случайных чисел, а генератор случайных чисел с зерном (seed). Стандартный модуль `random` поддерживает такую возможность. Это зерно должно быть параметром.

Для работы с файлом конфигурации удобно воспользоваться модулем `ConfigParser` (изучите самостоятельно).

При написании решения разрешено пользоваться только стандартной библиотекой. Для анализа результатов можно пользоваться любыми инструментами (рекомендуемый вариант: NumPy, SciPy, Matplotlib, Pandas).

ОФОРМЛЕНИЕ РАБОТЫ

При оформлении работы должны соблюдаться приведенные ниже требования.

- По результатам анализа цикличности нужно написать небольшой отчет. Отчет может состоять из изображений (PNG, JPEG, TIFF), PDF-файлов и TXT-файлов.
- Для основных функций в программе нужно написать юнит-тесты.
- Для программы должен быть сделан удобный интерфейс командной строки. Должны быть предусмотрены опции для пути к файлу конфигурации, количества итераций, режима работы (вывод на экран или сбор статистики).
- Основной файл программы должен называться `main.py` — из него должен быть доступен консольный интерфейс. Этот файл может содержать весь код или импортировать что-то из других файлов. Тесты должны быть выделены в отдельные файлы, причем файл с тестами для кода в `<name>.py` должен называться `<name>_ut.py` или `test_<name>.py`. Код для анализа результатов и создания отчета должен лежать в файле `report.py`. Все остальные файлы отчета должны лежать в папке `report`. Также нужно приложить хотя бы один вариант корректной конфигурации.
- Все файлы с кодом должны быть проверены на наличие проблем с помощью утилиты `pep8.py`. Можете использовать утилиту, установленную у себя на компьютере или воспользоваться приложенным контестом.

ВАРИАЦИИ

Ниже приведены детали реализации, которые можно менять при желании сделать модель более интересной. Для каждой детали указан базовый вариант и пример, как ее можно менять.

- **Гибель хищников от голода.** Базовый вариант — если хищник никого не ел в течение n итераций, он умирает. Можно, например, сделать, чтобы при поедании жертвы у хищника увеличивалась «сытость», а каждую итерацию она снижалась на 1. При достижении «сытости» нуля хищник погибает.
- **Размножение.** Базовый вариант — каждые n итераций каждое существо генерирует новое существо рядом с собой (n может быть разным у хищников и жертв). Можно вводить зависимости этих периодов от сытости, наличия существ другого типа рядом и т.п.
- **Порядок обработки.** Базовый вариант — клетки обрабатываются сверху-вниз-слева-направо. Можно сделать, например, чтобы хищники всегда обрабатывались раньше. Но следите за тем, чтобы каждый хищник/жертва обрабатывались ровно один раз за итерацию.
- **Выбор действий.** Базовый вариант — каждое существо каждый ход делает случайное действие из возможных (при это каждые n итераций безусловно размножается). Можете сделать, чтобы существа вели себя более интеллектуально, например, хищники могут двигаться в сторону жертвы, если она достаточно близко и всегда есть ее, если она рядом. Но учтите, что в задаче подразумевается моделирование океана большого размера (для проверки гипотезы цикличности), поэтому слишком ресурсоемкую логику поведения делать не нужно.