

# Чек-лист проекта «Изгиб Питона»

---

## Технологии для освоения

- Классы и наследование.
- Вызовы методов базового класса.
- Установка и импорт сторонней компоненты.
- Импорт и применение штатной компоненты Python.

## Обязательные критерии

Все автотесты успешны.

Проект можно запустить и поиграть. Должны быть соблюдены такие условия: - Змейка проходит сквозь границы окна. Например, достигая верха, появляется снизу. - Змейка ест яблоки и увеличивается. - Когда змейка совершает «самоукус» — то есть голова змейки сталкивается с хвостом — хвост отбрасывается, а остаётся только голова.

Стартовое или новое положение яблока нужно выбирать с учётом положения змейки. Не должно быть никакого шанса, что яблоко окажется на теле змейки.

В методе `draw()` в базовом классе не должно быть никаких рисующих действий. Ведь смысл этого метода «нарисовать всю фигуру», а базовый класс её не знает.

Все рисующие на поле действия — заливка поля, раскраска ячейки или рамки — могут находиться либо в функции `main()`, либо в методах `draw()`.

## Рекомендации

Игра заканчивается при нажатии `ESC`.

В заголовке окна отображается рекордный размер змейки.

В коде нет повторных применений литералов. Например, если в расчётах участвуют габариты окна, они применяются через именованные константы.

В коде нет повторных вычислений констант. Например, если потребуется центральная позиция поля, она вычисляется однократно.

В коде нет повторений одинаковых действий. Например, при создании змеи и после «самоукуса» нужно задать значения нескольких полей — эти действия должны быть описаны в единственном месте.

Раскраска одной ячейки для яблока или для змейки — это одинаковые действия с параметрами «позиция» и «цвет». Для этого действия лучше создать новый метод базового класса.

У змейки на одном шаге меняется только голова и иногда последняя ячейка хвоста. Поэтому не нужно каждый раз рисовать все сегменты.

Самоукус не может произойти у короткой змеи. Это можно учесть в коде.

Расчёт нового направления для змеи выполнять не через if-elif-elif-elif, а через извлечение из константного словаря {(клавиша, старое-направление): новое-направление}

Если применять для извлечения метод `новое = словарь.get((клавиша, старое), старое)`, такой словарь будет содержать восемь элементов.

Расчёт нового положения головы для змеи выполнять одним действием (без if-else) через компоненты старого положения и компоненты направления, применяя «остаток от деления».

Расчёт свободной ячейки для яблока можно выполнять без цикла, если заранее рассчитать множество из всех ячеек поля в константу `ALL_CELLS = set(ячейка_x_y for x in range(ширина) for y in range(высота))` и применить разность множеств `случайная_ячейка = choice(tuple(ALL_CELLS - set(занятые)))`.

Рекордные размеры змейки можно сохранять между запусками.

Можно усложнить правил игры: - Менять скорость змейки при нажатии каких-то клавиш, например, q и w. В этом случае текущую скорость и применяемые клавиши обязательно показывать, например, в заголовке окна. - На поле появляется не одно, а несколько яблок. - На поле появляются не только яблоки, но и случайные камни, «отравы» и так далее.

Можете познакомиться с удачными усложнениями в статье на Хабре.

[[https://habr.com/ru/companies/yandex\\_praktikum/articles/811707/](https://habr.com/ru/companies/yandex_praktikum/articles/811707/)]