

## Виртуальная фабрика роботов-художников

Вы создаете программное обеспечение для фабрики, которая производит роботов-художников. Роботы умеют "рисовать" (генерировать строковые паттерны) и их можно комбинировать для создания сложных произведений.

### Часть 1:

Создайте базовый класс PainterBot:

- Атрибуты: name (имя), style (стиль рисования, например, "Пиксельный"), efficiency (эффективность, целое число от 1 до 10).
- Метод paint(length): возвращает строку, представляющую рисунок длиной length. **Базовая реализация:** возвращает строку из length символов #.
- Создайте два класса-потомка, которые переопределяют метод paint:
- LinePainterBot (рисует линиями): его paint(length) возвращает строку из length символов =.
- WavePainterBot (рисует волнами): его paint(length) возвращает строку, где символы ~ и - чередуются (например, для length=5: ~-~-~).

### Часть 2:

Напишите функцию gallery\_exhibition(painter\_list, length).

- Эта функция принимает список роботов-художников (painter\_list) и длину рисунка length.
- Функция должна вызвать метод paint(length) у **каждого** робота в списке и напечатать результат вместе с его именем.

### Часть 3:

Добавьте роботам возможность "объединяться" для совместной работы.

- Перегрузите оператор + (\_\_add\_\_). Когда два робота a и b складываются (a + b), должен создаваться и возвращаться **новый** робот (ComboPainterBot).

- Новый робот получает комбинированное имя: `f"Комбо [{a.name} + {b.name}]"`.
- Его стиль: "Комбинированный".
- Его эффективность: меньшая из двух эффективностей (слабый робот замедляет связку).
- Его метод `paint(length)`: возвращает строку, где рисунки от двух исходных роботов соединены через `|`. Например, если `a.paint(3)` вернул `###`, а `b.paint(3)` вернул `===`, то их комбо вернет `### | ===`.