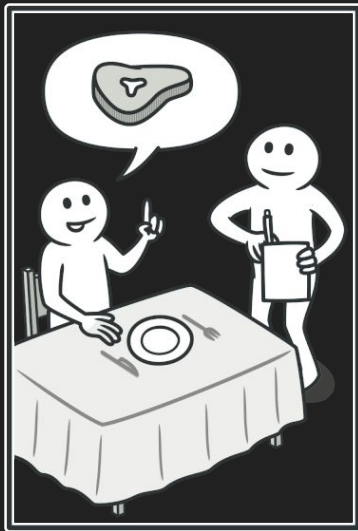


# Command

— это поведенческий паттерн проектирования, который превращает запросы в объекты, позволяя передавать их как аргументы при вызове методов, ставить запросы в очередь, логировать их, а также поддерживать отмену операций.

# Аналогия из жизни

отправитель

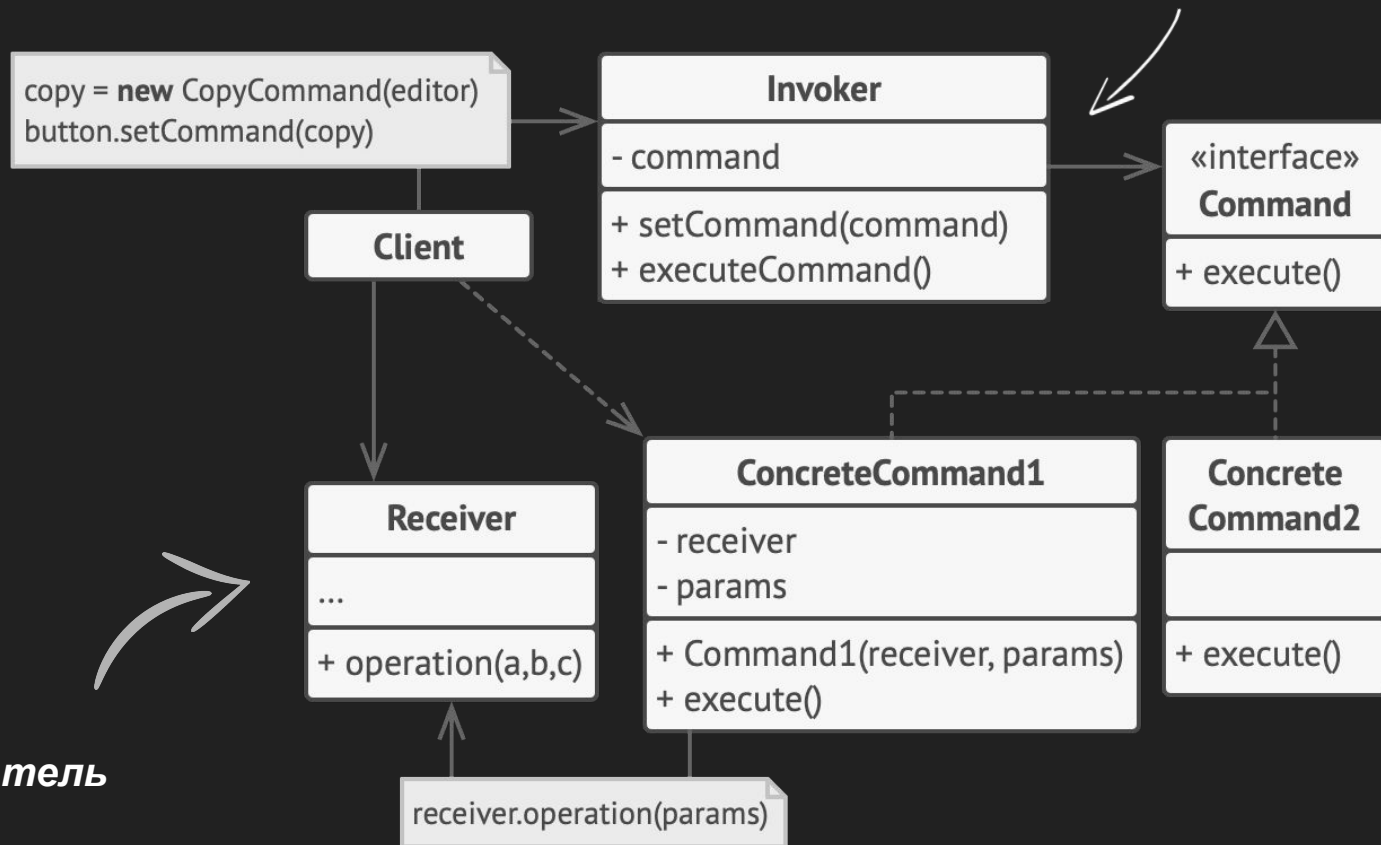


команда

исполнитель

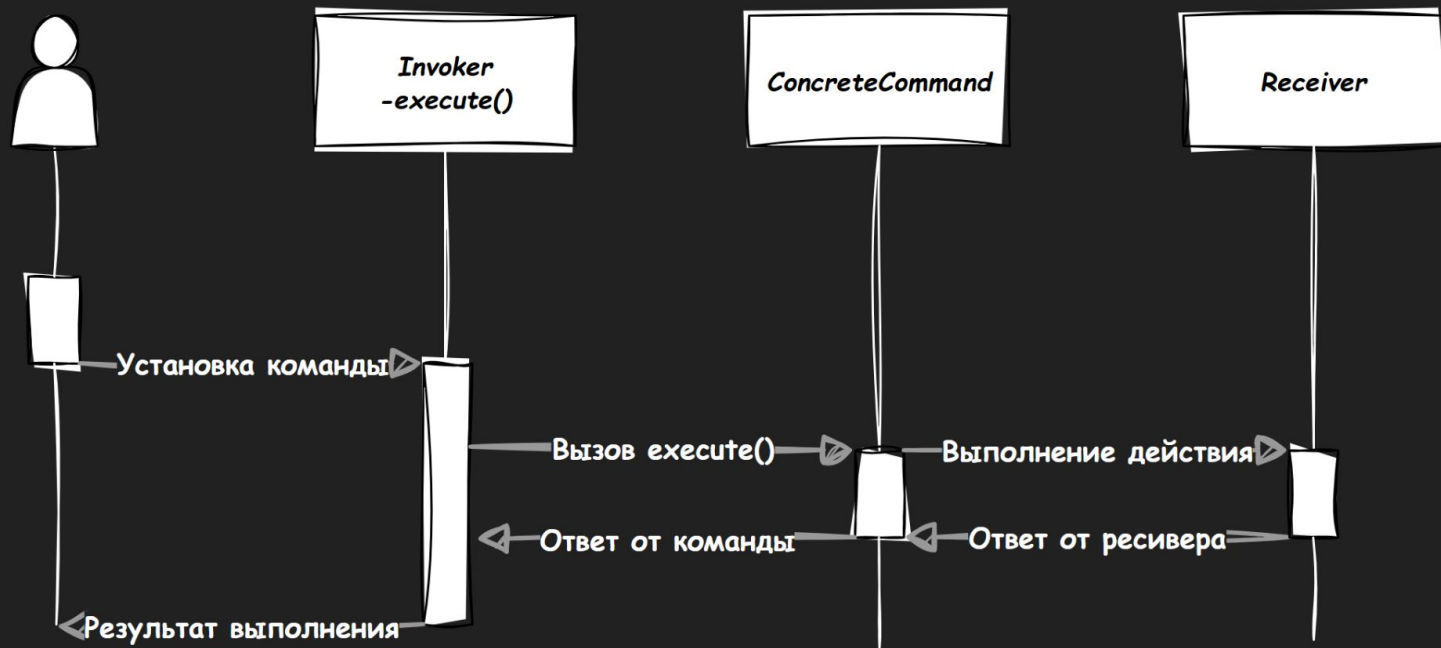
# Структура

*получатель*



*отправитель*

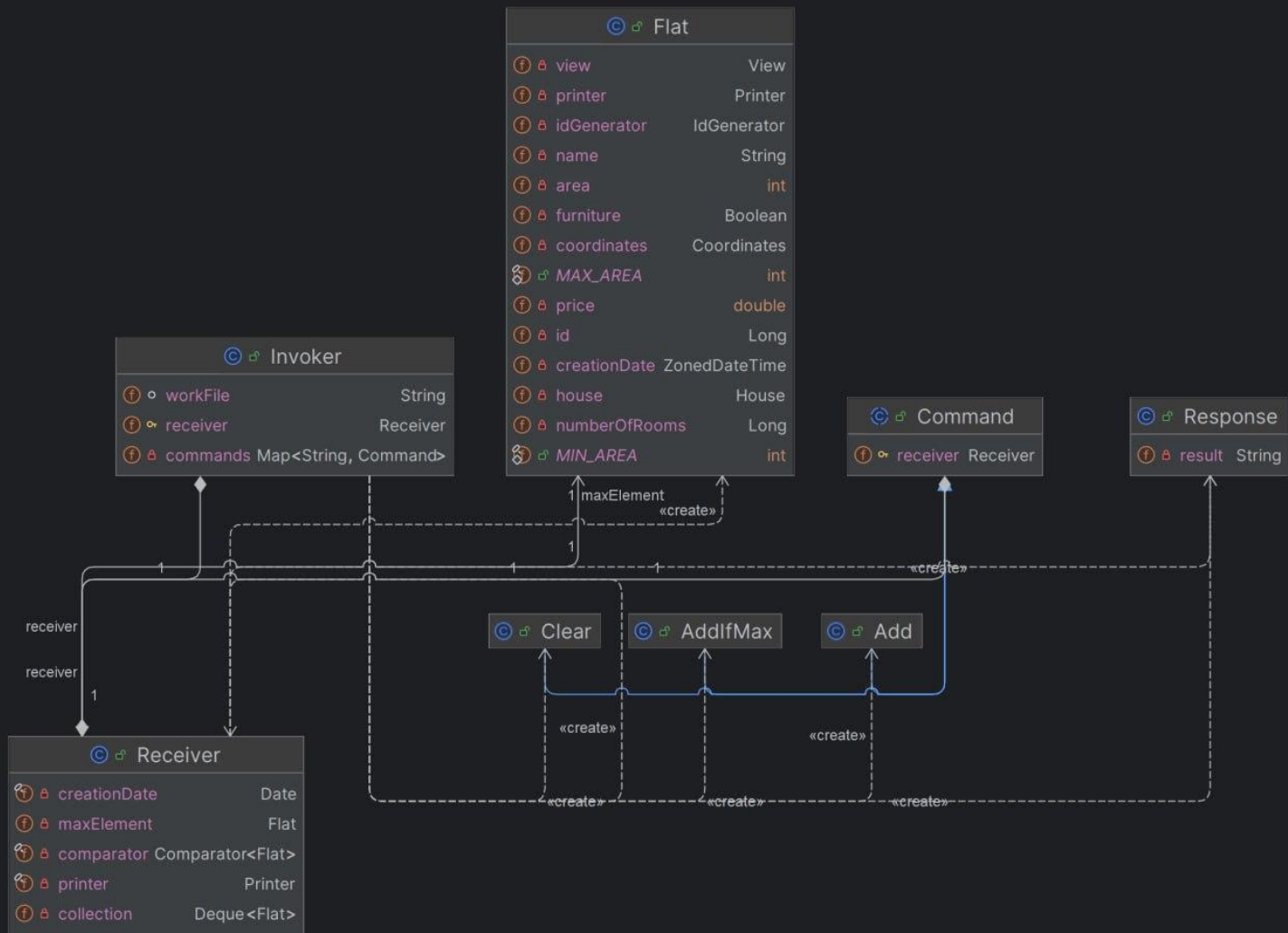
# Диаграмма последовательности UML



# Применимость

Когда:

- Хотим параметризовать объекты выполняемым действием.
- Хотим ставить операции в очередь, выполнять их по расписанию или передавать по сети.
- Хотим реализовать операцию отмены.



# Преимущества и недостатки

- ✓ Убирает прямую зависимость между объектами, вызывающими операции, и объектами, которые их непосредственно выполняют.
- ✓ Позволяет реализовать простую отмену и повтор операций.
- ✓ Позволяет реализовать отложенный запуск операций.
- ✓ Позволяет собирать сложные команды из простых.
- ✓ Реализует *принцип открытости/закрытости*.
- ✗ Усложняет код программы из-за введения множества дополнительных классов.