

BUILDER

Builder wzorec projektowy, który umożliwia tworzenie różnych typów i reprezentacji obiektu na podstawie tego samego procesu budowania i pozwala na tworzenie skomplikowanych obiektów krok po kroku.

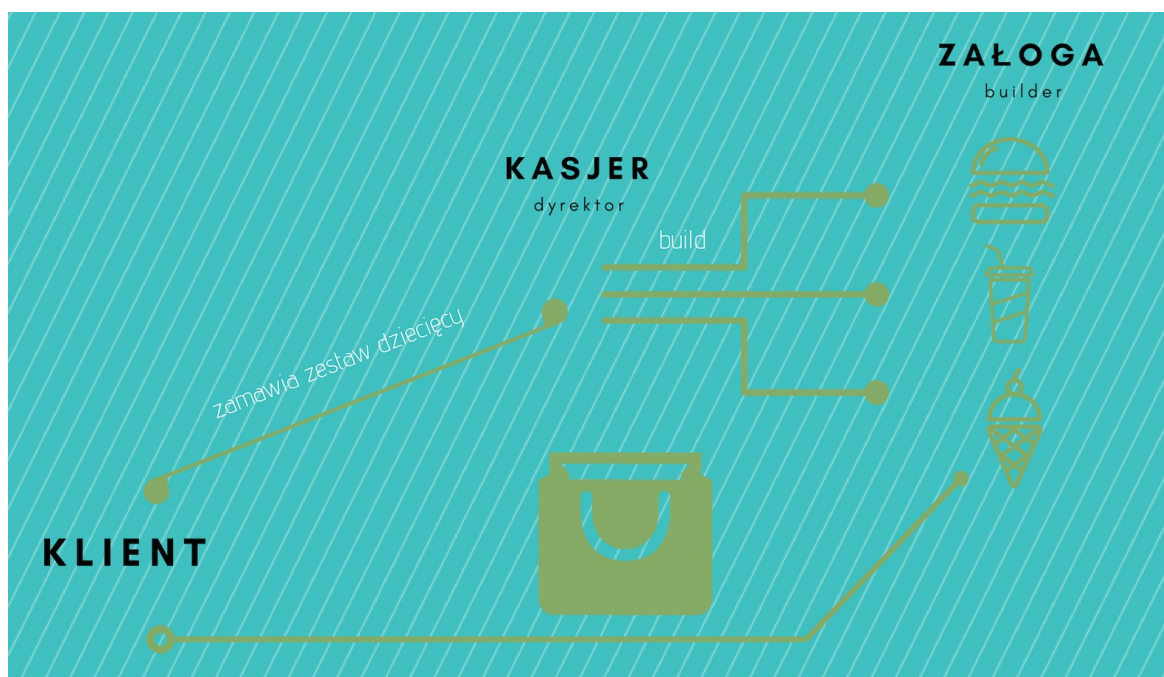
Kiedy korzystać?

- Zbyt duża liczba parametrów w konstruktorze.
`HappyMeal happymeal = new HappyMeal(hamburger, fries, toy, drink, extra);`
- Duża złożoność produktu – możemy realizować krok po kroku

Wzorec builder jest w gruncie rzeczy obiektem opakującym wszystkie możliwe parametry, które mógłbyś chcieć podać w konstruktorze. Pozwala na używanie setterów to budowania krok po kroku listy parametrów.

Wzorec wykorzystuje następujące obiekty:

- 1) Dyrektor – obiekt odpowiedzialny za tworzenie obiektu docelowego za pomocą buildera
- 2) Builder – odpowiada za tworzenie elementów obiektu docelowego
- 3) Produkt – obiekt docelowy



Dyrektor wywołuje **builder** interpretując odpowiednio dane wejściowe. **Builder** za każdym razem gdy jest wywoływany tworzy część produktu finalnego - bardziej złożonego obiektu. Kiedy produkt finalny zostanie ukończony **klent** otrzyma wynik z **buildera**.

```
// Builder
class HappyMealBuilder {
    String hamburger, fries, toy, drink, extra;
    HappyMealBuilder setHamburger(hamburger) { this.hamburger = hamburger;
return this; }
    HappyMealBuilder setFries(fries) { this.fries = fries;    return this; }
    HappyMealBuilder setToy(toy) { this.toy = toy; return this; }
    HappyMealBuilder setDrink(drink) { this.drink = drink; return this; }
    HappyMealBuilder setExtra(extra) { this.extra = extra; return this; }
    HappyMeal build() {
        return new HappyMeal(this); // Pass in the builder
    }
}

// Usage
HappyMeal happymeal = new HappyMealBuilder()
    .setHamburger("cheeseburger")
    .setFries("duze")
    .setToy("potwór")
    .setDrink("woda niegazowana")
    .setExtra("jabłko")
    .build();
```