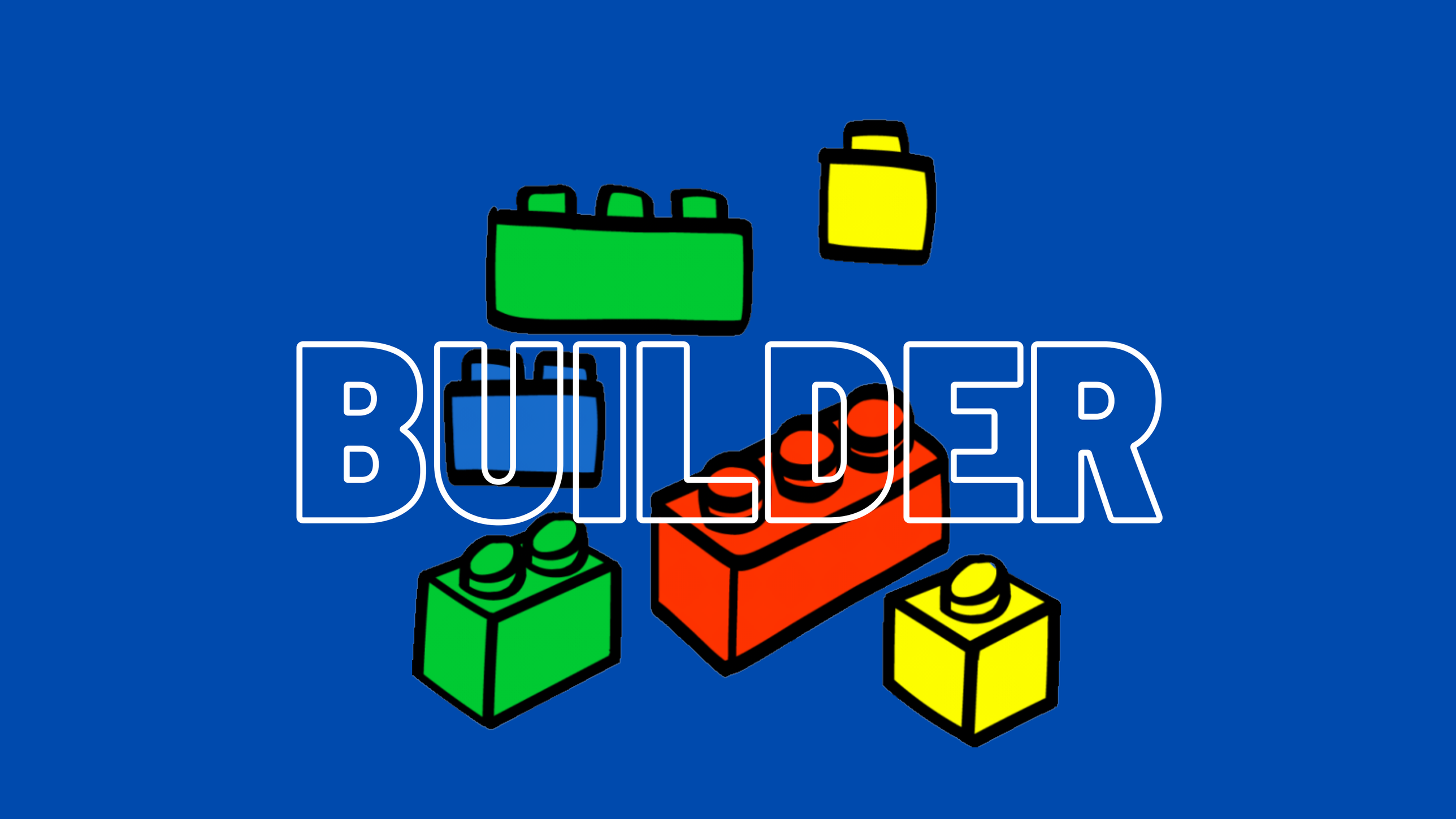

BUILDER

STIAI CA?

1.Builder este un pattern de proiectare în Java care ajută la crearea de obiecte complexe prin specificarea proprietăților dorite.

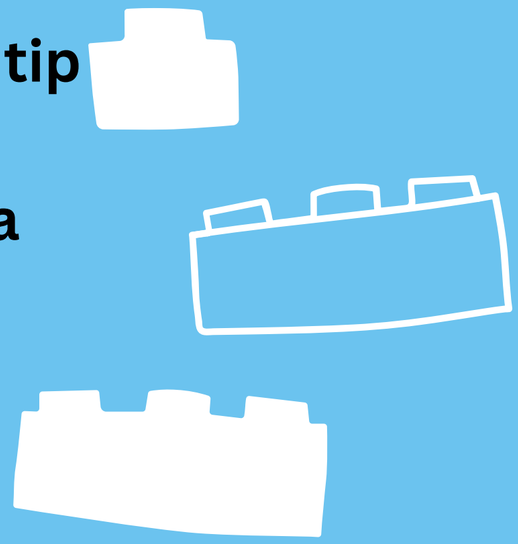
2.Acesta separă procesul de construcție a obiectului de reprezentarea acestuia, permițând crearea diferitelor reprezentări ale aceluiași obiect fără a afecta procesul de construcție.

3.Acesta utilizează o interfață builder separată care permite specificarea opțională a diferitelor proprietăți ale obiectului și creează obiectul final printr-un apel la o metodă de construire.

4.Obiectele create prin intermediul unui builder sunt de obicei finale, ceea ce înseamnă că nu pot fi modificate după crearea lor.

Cuvintele cheie care ne pot ajuta sa recunoastem pattern-ul Builder in Java sunt:

- Clasa Builder care este o clasa separata care se ocupa de construirea obiectului
- Metode de tip set pentru fiecare proprietate a obiectului ce poate fi setata
- Metoda build care finalizeaza construirea obiectului si il returneaza
- Posibilitatea de a lantui mai multe apeluri de metode de tip set in timpul construirii obiectului
- Utilizarea design pattern-ului Builder ajuta la construirea obiectelor complexe si la evitarea constructorilor cu un numar mare de parametri.



SINGLETON – CLASA BUILDER POATE FI SINGLETON, ASTFEL ÎNCÂT, CONSTRUIREA DE OBIECTE SĂ FIE CENTRALIZATĂ

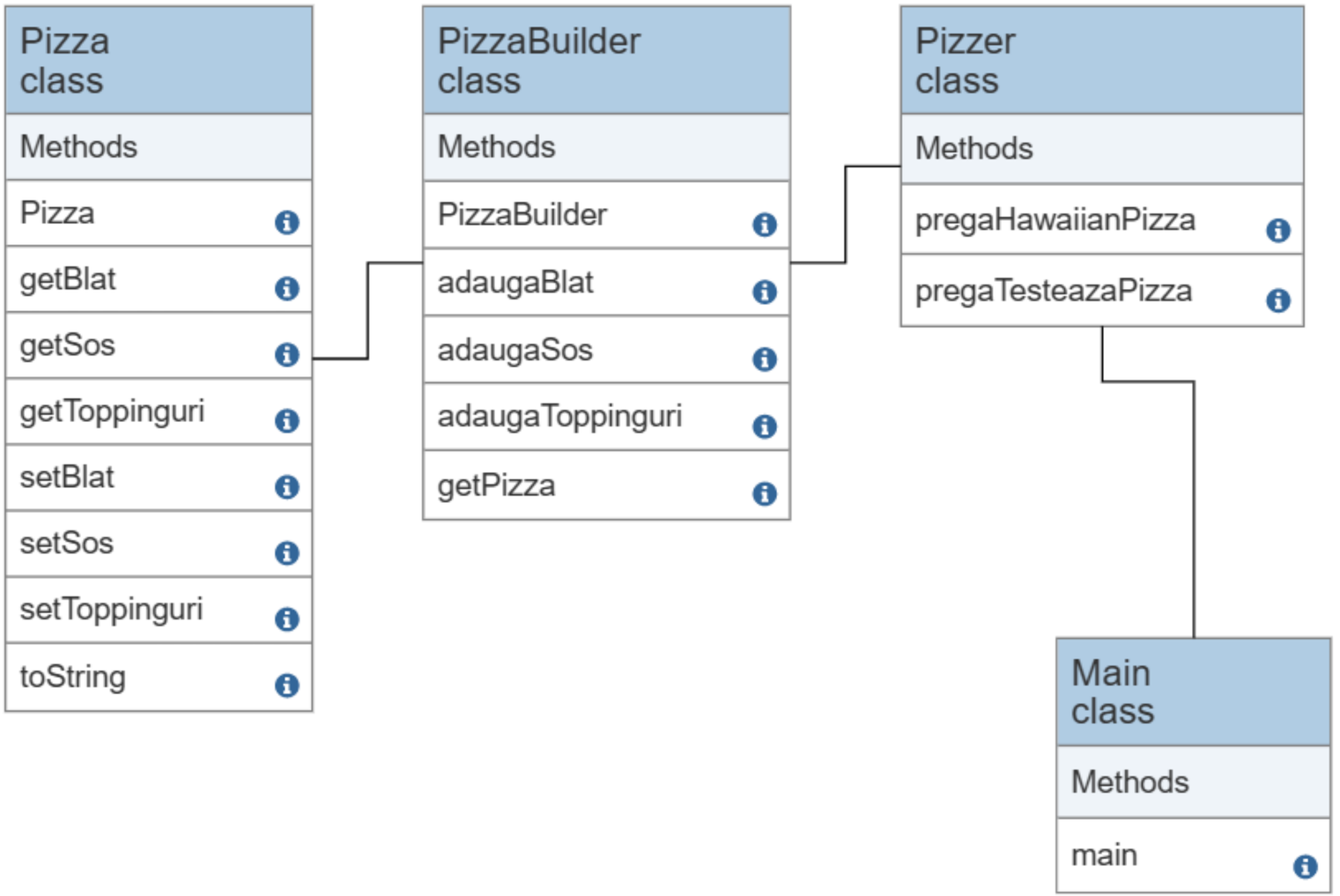
PIZZA -BUILDER

1.Clasa **Pizza** - Aceasta clasa este responsabila pentru reprezentarea unei pizze. Aceasta contine trei variabile membru, una pentru blat, una pentru sos si una pentru topping-uri. Clasa contine de asemenea un constructor si metode **get** si **set** pentru fiecare variabila membru pentru a putea accesa si modifica valorile acestora.

2.Clasa **PizzaBuilder** - Aceasta clasa este responsabila pentru construirea unui obiect de tip Pizza. Aceasta contine o referinta catre obiectul **Pizza** in constructia caruia se vor adauga diferitele proprietati ale acesteia, prin intermediul unor metode de tip **set**. Metodele de tip **set** returneaza un obiect de tip **PizzaBuilder**, astfel incat acestea pot fi lantuite in timpul construirii obiectului Pizza. Clasa contine de asemenea o metoda **get** pentru a returna obiectul Pizza construit.

3.Clasa **Pizzer** - Aceasta clasa contine metode pentru a construi diferite tipuri de pizza folosind obiectul **PizzaBuilder**. Metoda **pregaTesteazaPizza** construieste o pizza traditionala cu blat, sos si topping-uri specificate, iar metoda **pregaHawaiianPizza** construieste o pizza hawaiiana cu blat, sos si topping-uri specificate.

Principiul builder este utilizat in aceste clase pentru a construi obiecte Pizza. In loc sa se foloseasca un constructor cu parametri pentru a crea un obiect Pizza, clasa PizzaBuilder este folosita pentru a construi obiectul, permitand astfel adaugarea sau omisiunea unor proprietati de-a lungul construirii, fara a fi nevoie sa se creeze mai multi constructori. Acest lucru faciliteaza crearea obiectelor Pizza, in special atunci cand exista mai multe combinatii posibile de proprietati pentru aceasta.


Pizza class
Methods
Pizza
getBlat
getSos
getToppinguri
setBlat
setSos
setToppinguri
toString
PizzaBuilder class
Methods
PizzaBuilder
adaugaBlat
adaugaSos
adaugaToppinguri
getPizza
Pizzer class
Methods
pregaHawaiianPizza
pregaTesteazaPizza
Main class
Methods
main

```
package clase;

public class PizzaBuilder {
    private Pizza pizza;

    public PizzaBuilder() {
        pizza = new Pizza("", "", "");
    }

    public PizzaBuilder adaugaBlat(String blat) {
        pizza.setBlat(blat);
        return this;
    }

    public PizzaBuilder adaugaSos(String sos) {
        pizza.setSos(sos);
        return this;
    }

    public PizzaBuilder adaugaToppinguri(String toppinguri) {
        pizza.setToppinguri(toppinguri);
        return this;
    }

    public Pizza getPizza() {
        return pizza;
    }
}
```

În secvența de cod prezentată, se aplică principiul Builder pentru a construi obiecte de tip Pizza. Clasa PizzaBuilder este utilizată pentru a permite construirea obiectelor de tip Pizza pas cu pas, în funcție de preferințele utilizatorului.

Prin intermediul metodelor "adaugaBlat", "adaugaSos" și "adaugaToppinguri", utilizatorul poate specifica tipul blatului, tipul sosului și topping-urile dorite pentru pizza. Fiecare metodă setează valoarea corespunzătoare în obiectul Pizza folosit de PizzaBuilder.

Metoda "getPizza" este apoi utilizată pentru a obține obiectul Pizza construit, după ce toate opțiunile au fost setate.

Astfel, prin utilizarea principiului Builder în acest caz, se oferă flexibilitate și ușurință de utilizare utilizatorilor, care pot specifica opțiunile dorite în mod incremental, evitând astfel construirea unui constructor supraincarcat cu toate opțiunile posibile.

Ex. pentru acasă:

Să se implementeze un builder pentru crearea unui obiect de tip Burger, având câmpurile "tip de chiflă", "carne", "sos", "legume" și "condimente". Builderul trebuie să ofere metode pentru a seta fiecare câmp în parte, respectiv "setChifla(String)", "setCarne(String)", "setSos(String)", "setLegume(String)" și "setCondimente(String)". De asemenea, builderul trebuie să ofere o metodă "build()" pentru a crea obiectul Burger cu valorile setate în câmpuri. Trebuie să se asigure că obiectul Burger creat are toate câmpurile setate corect.