

Design pattern

BUILDER

20/11/2017

LAFONT Mathieu
SAUVIAT Louis

ETCHEVERRY Mathieu
TYTGAT Benjamin

PLAN

Introduction

I] Présentation du pattern Builder

1) Problématique

2) Le pattern Builder : une solution !

II] Exemples

Conclusion

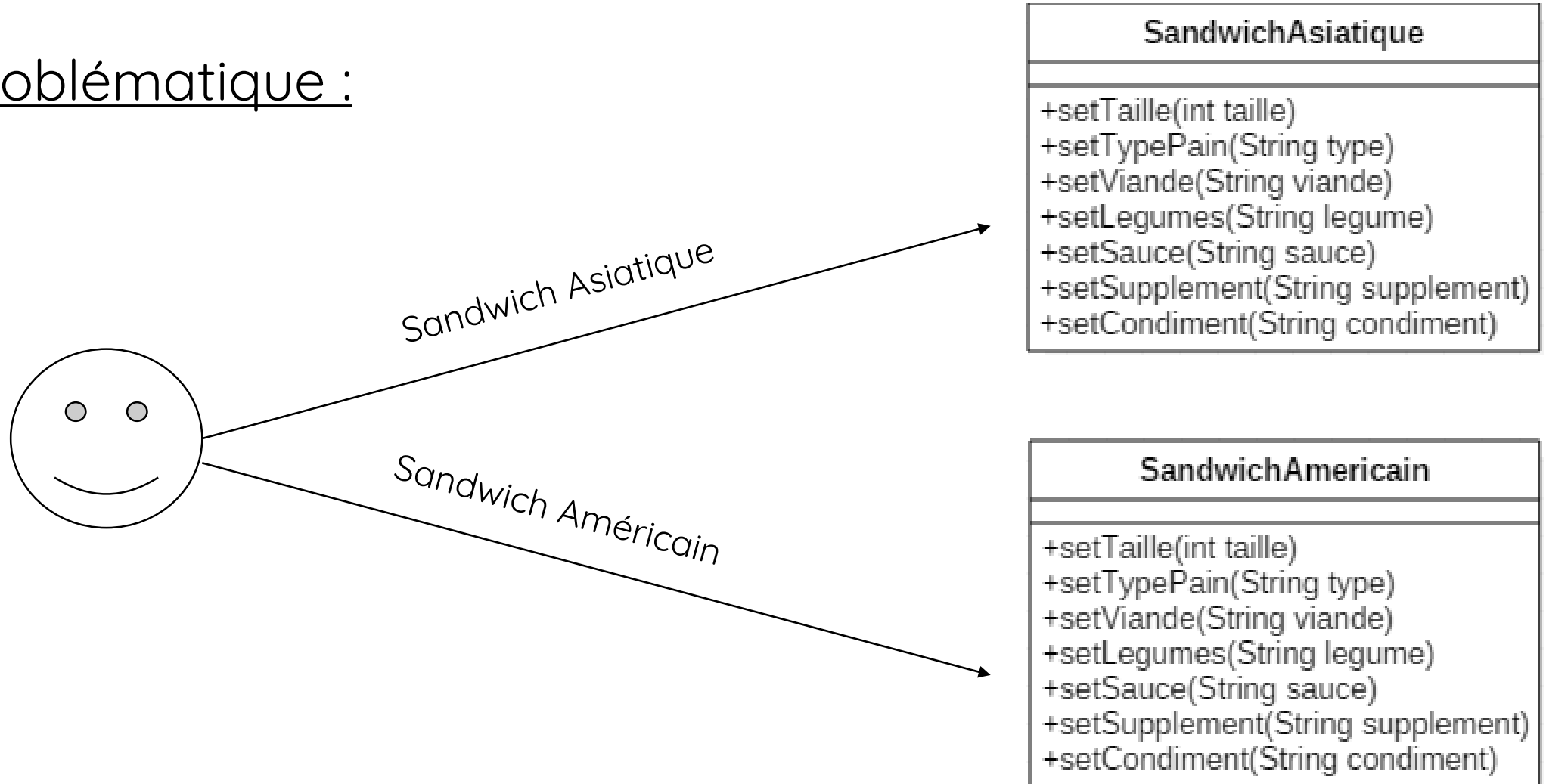
INTRODUCTION

Petits rappels :

- Solution à un problème récurrent
- Définit par nom
- Répond à une problématique
- Solution

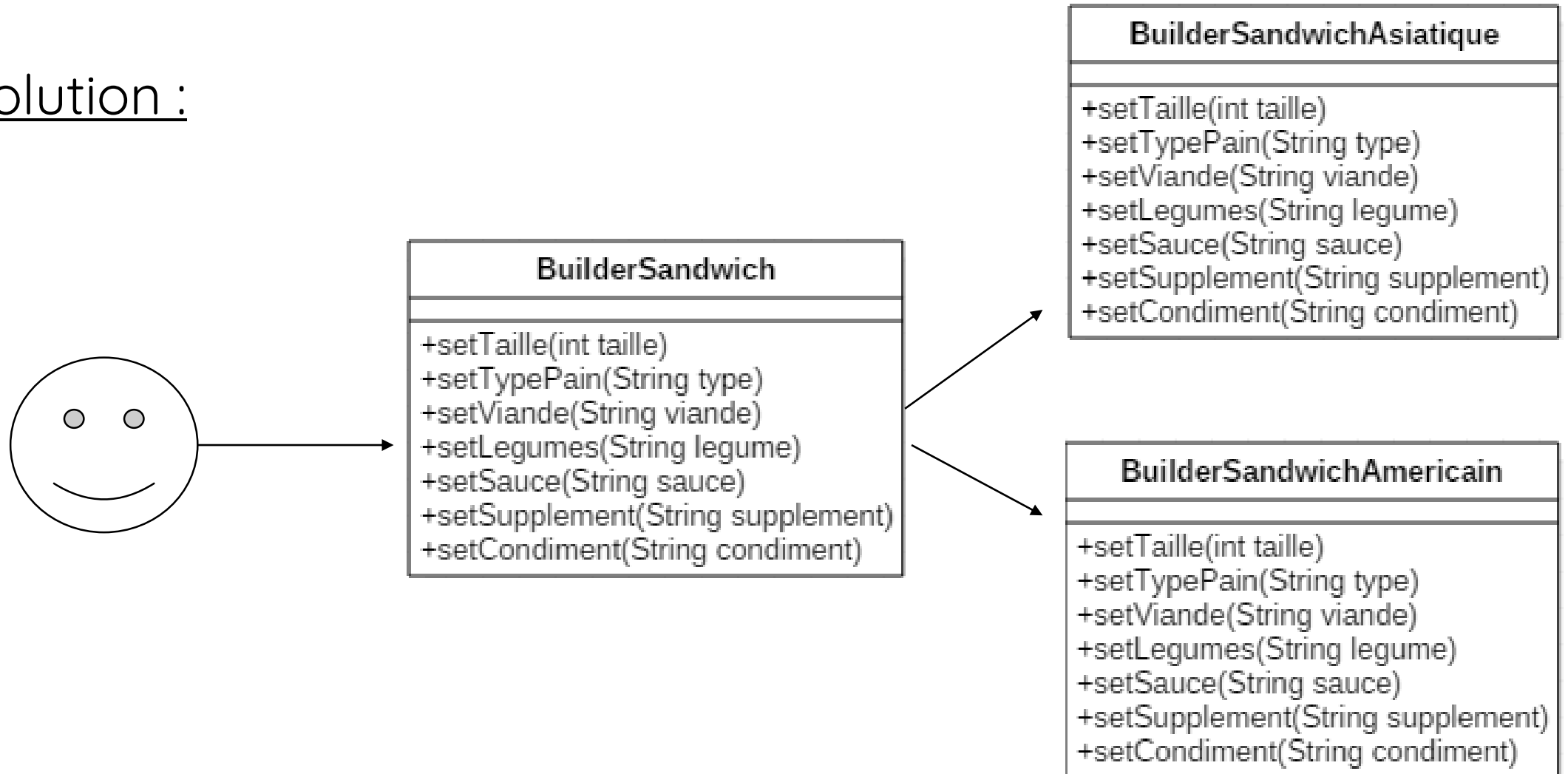
I] PRESENTATION DU PATTERN

1) Problématique :

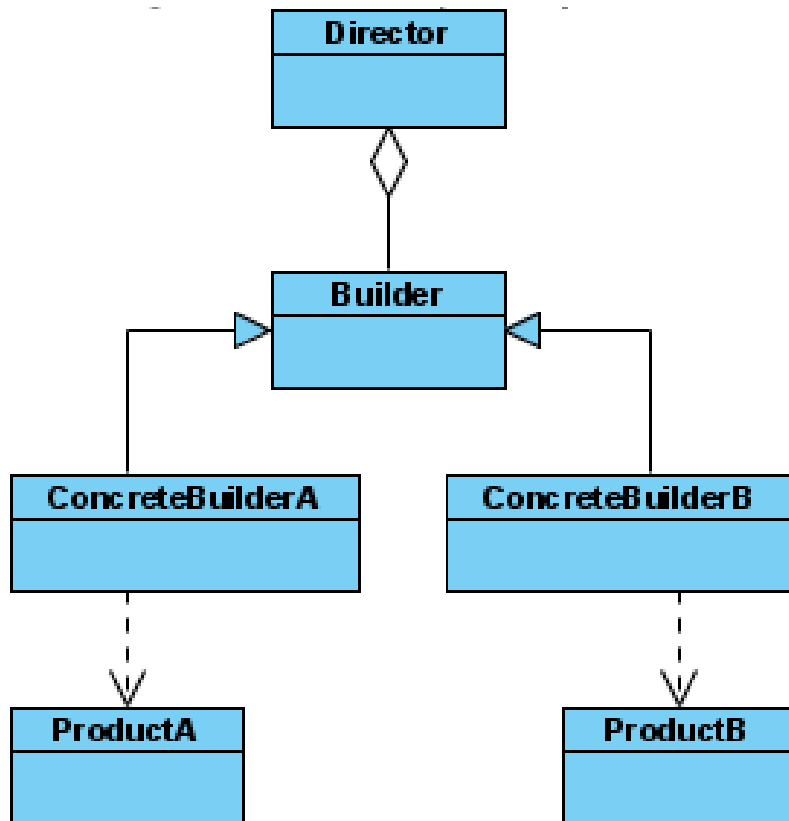


I] PRESENTATION DU PATTERN

2) Solution :



I] PRESENTATION DU PATTERN



Faire un sandwich

Recette d'un sandwich

Recette spécifique

Sandwich fini

II] EXEMPLES

Toujours sur le principe de notre Subway :

- Création du Builder
- Création des ConcreteBuilder
- Création des Produits
- Création du Director

Le produit : un sandwich

```
1 public class Sandwich {
2     private int taille;
3     private String type = "";
4     private String viande = "";
5     private String legume = "";
6     private String sauce = "";
7     private String supplement = "";
8     private String condiment = "";
9
10
11     public void setTaille(int taille) {
12         this.taille = taille;
13     }
14
15     public void setTypePain(String type) {
16         this.type = type;
17     }
18
19     public void setViande(String viande) {
20         this.viande = viande;
21     }
22
23     public void setLegumes(String legume) {
24         this.legume = legume;
25     }
26
```

```
26
27     public void setSauce(String sauce) {
28         this.sauce = sauce;
29     }
30
31     public void setSupplement(String supplement) {
32         this.supplement = supplement;
33     }
34
35     public void setCondiment(String condiment) {
36         this.condiment = condiment;
37     }
38
```


Le Director, classe abstraite

```
1
2 public class Director {
3
4     private BuilderSandwich builderSandwich;
5
6     public void setBuilderSandwich(BuilderSandwich mp) {
7         builderSandwich = mp;
8     }
9
10    public Sandwich getSandwich() {
11        return builderSandwich.getSandwich();
12    }
13
14    public void construireSandwich() {
15        builderSandwich.creerNouveauSandwich();
16        builderSandwich.setTaille();
17        builderSandwich.setTypePain();
18        builderSandwich.setViande();
19        builderSandwich.setLegumes();
20        builderSandwich.setSauce();
21        builderSandwich.setSupplement();
22        builderSandwich.setCondiment();
23    }
24 }
25
```

Le Builder (la recette)

```
1 public abstract class BuilderSandwich {
2     protected Sandwich sandwich;
3
4     public Sandwich getSandwich() {
5         return sandwich;
6     }
7
8     public void creerNouveauSandwich() {
9         sandwich = new Sandwich();
10    }
11
12    public abstract void setTaille();
13
14    public abstract void setTypePain();
15
16    public abstract void setViande();
17
18    public abstract void setLegumes();
19
20    public abstract void setSauce();
21
22    public abstract void setSupplement();
23
24    public abstract void setCondiment();
25 }
26
```

Le ConcreteBuilder asiatique

```
1
2 public class BuilderSandwichAsiatique extends BuilderSandwich {
3
4     public void setTaille() {
5         sandwich.setTaille(30);
6     }
7
8     public void setTypePain() {
9         sandwich.setTypePain("Pain au sésame");
10    }
11
12    public void setViande() {
13        sandwich.setViande("Canard");
14    }
15
16    public void setLegumes() {
17        sandwich.setLegumes("Champignons noirs + carottes");
18    }
19
20    public void setSauce() {
21        sandwich.setSauce("Nuoc mam");
22    }
23
24    public void setSupplement() {
25        sandwich.setSupplement("Jus de citron");
26    }
27
28    public void setCondiment() {
29        sandwich.setCondiment("Cornichons");
30    }
31 }
```

Le ConcreteBuilder américain

```
1
2 public class BuilderSandwichAmericain extends BuilderSandwich {
3
4     public void setTaille() {
5         sandwich.setTaille(15);
6     }
7
8     public void setTypePain() {
9         sandwich.setTypePain("Pain de campagne");
10    }
11
12    public void setViande() {
13        sandwich.setViande("Boeuf");
14    }
15
16    public void setLegumes() {
17        sandwich.setLegumes("Frites + frites");
18    }
19
20    public void setSauce() {
21        sandwich.setSauce("Barbecue + ketchup + mayonnaise");
22    }
23
```

```
23
24    public void setSupplement() {
25        sandwich.setSupplement("Viande");
26    }
27
28    public void setCondiment() {
29        sandwich.setCondiment("Cornichons");
30    }
31 }
```

La commande

```
1 import java.util.ArrayList;
2
3
4 public class Commande {
5
6     public static void main(String[] args) {
7
8         Director director = new Director();
9
10        List<Sandwich> sandwiches = new ArrayList<>();
11
12        BuilderSandwich builderSandwichAmericain = new BuilderSandwichAmericain();
13        BuilderSandwich builderSandwichAsiatique = new BuilderSandwichAsiatique();
14
15        director.setBuilderSandwich(builderSandwichAsiatique);
16        director.construireSandwich();
17        sandwiches.add(director.getSandwich());
18
19        director.setBuilderSandwich(builderSandwichAmericain);
20        director.construireSandwich();
21        sandwiches.add(director.getSandwich());
22
23        sandwiches.forEach(sandwich -> System.out.println(sandwich));
24    }
```

Pain au sésame - 30 - Canard - Cornichons - Champignons noirs + carottes - Nuoc mam - Jus de citron
Pain de campagne - 15 - Boeuf - Cornichons - Frites + frites - Barbecue + ketchup + mayonnaise - Viande