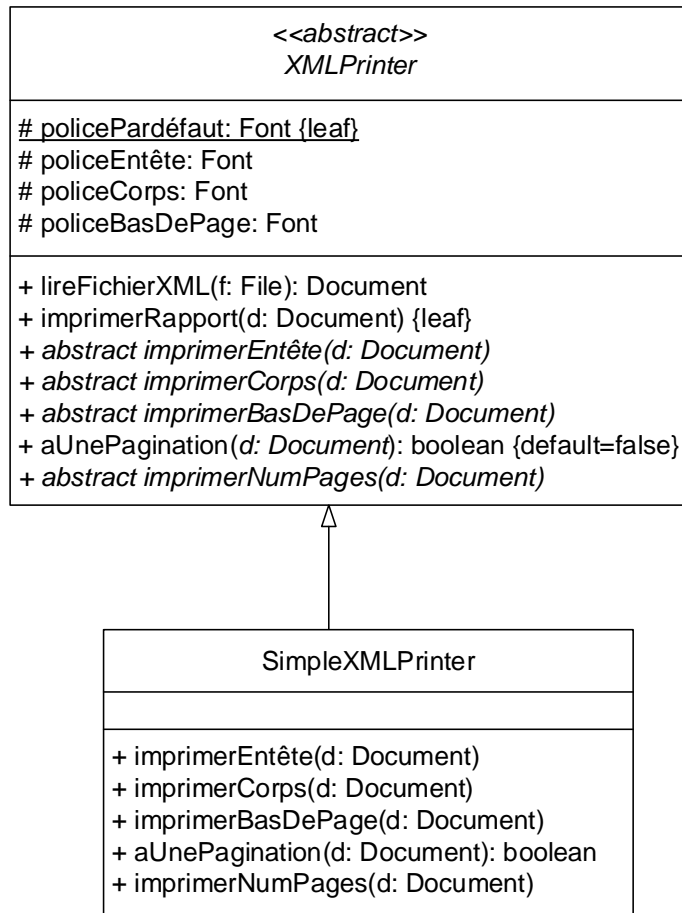


Solutions des exercices sur le patron « Méthode template »

Exercice 1

Le diagramme suivant présente une solution possible avec la classe XMLPrinter faisant partie du framework. La méthode imprimerRapport(d:Document) est la méthode template. Elle définit le squelette imposé par le framework d'impression et elle est finale.

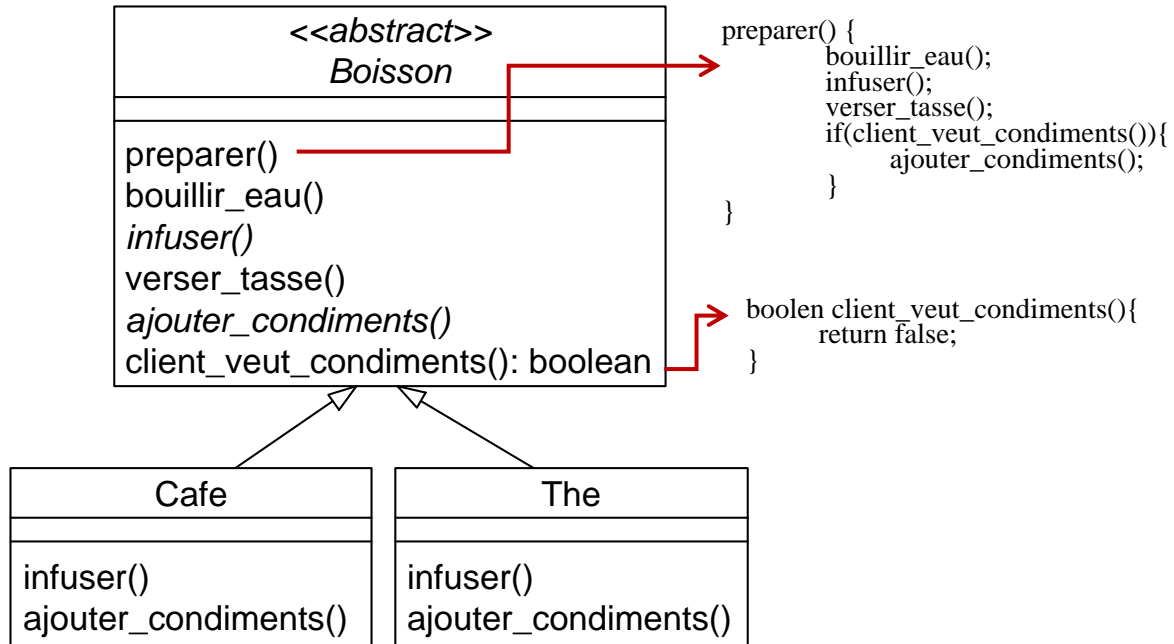


```

final imprimerRapport(d: Document) {
    imprimerEntête(d)
    imprimerCorps(d)
    imprimerBasDePage(d)
    si(aUnePagnation(d)){
        imprimerNumPages(d);
    }
}
  
```

Exercice 2

Pour rendre l'étape ajouter_condiments() optionnelle, on va ajouter une méthode client_veut_condiments() qui retourne un « false » pour spécifier que par défaut on ne rajoute pas de condiments. Comme illustré dans la figure qui suit, on va utiliser cette nouvelle méthode pour contrôler l'appel à la méthode ajouter_condiments(). Une sous-classe de Boisson peut redéfinir cette méthode pour retourner « true » et forcer l'ajout de condiments.



La méthode `client_veut_ajouter_condiments()` est dite une méthode « **hook** ». Contrairement aux méthodes primitives qui sont abstraites, une méthode « **hook** » implémente un comportement par défaut que les sous-classes peuvent étendre. Elle peut être redéfinie par les sous-classes pour implémenter une partie optionnelle d'un algorithme.