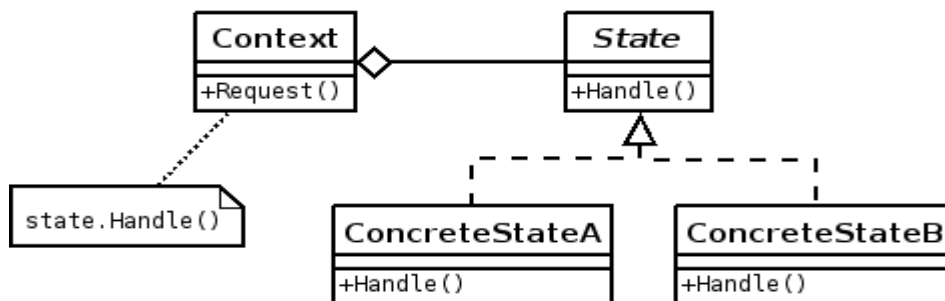


Software design patterns: 2010 – 2011

(exercise 7 – State, Strategy, Template method, Visitor)

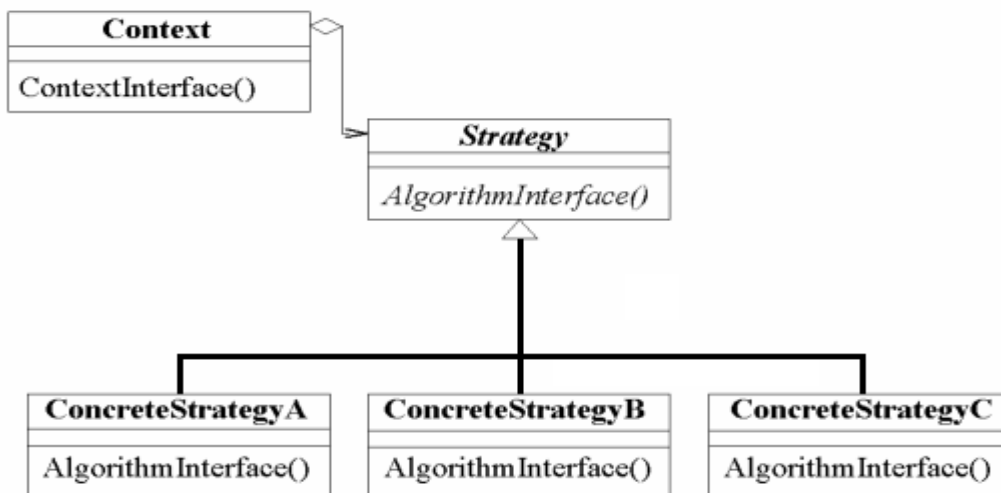
1. State design pattern:

Променя поведението на обект, в зависимост от промените във вътрешното си състояние.



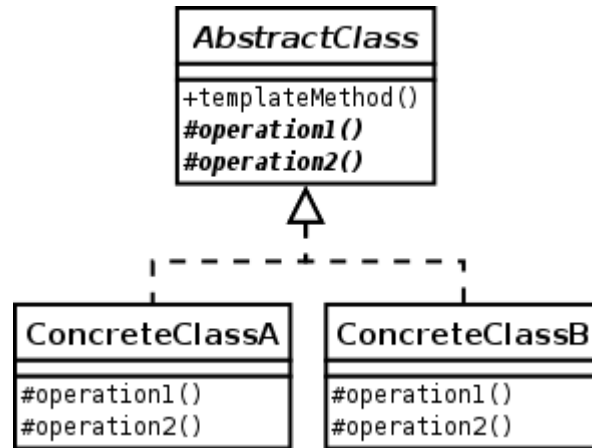
2. Strategy design pattern :

Дефинира семейство алгоритми, капсулира всеки от тях и ги прави взаимнозаменяеми.



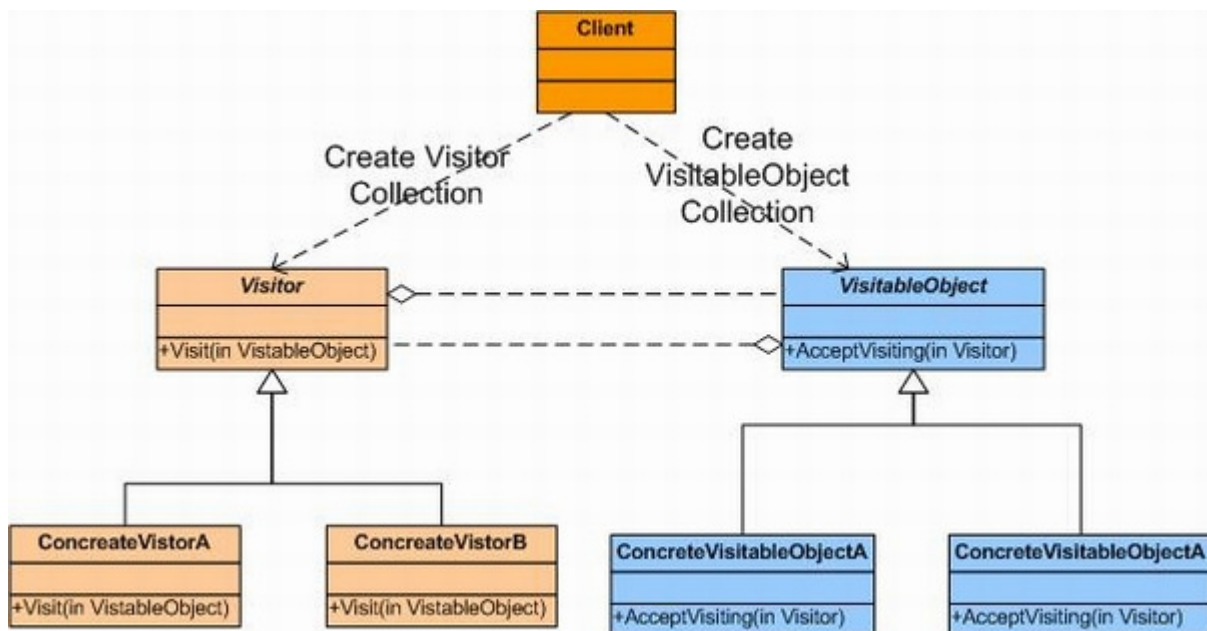
3. Template method design pattern:

Дефинира скелета на алгоритъм в операции, като остава на подкласовете да ги имплементират (или поне някои от тях)



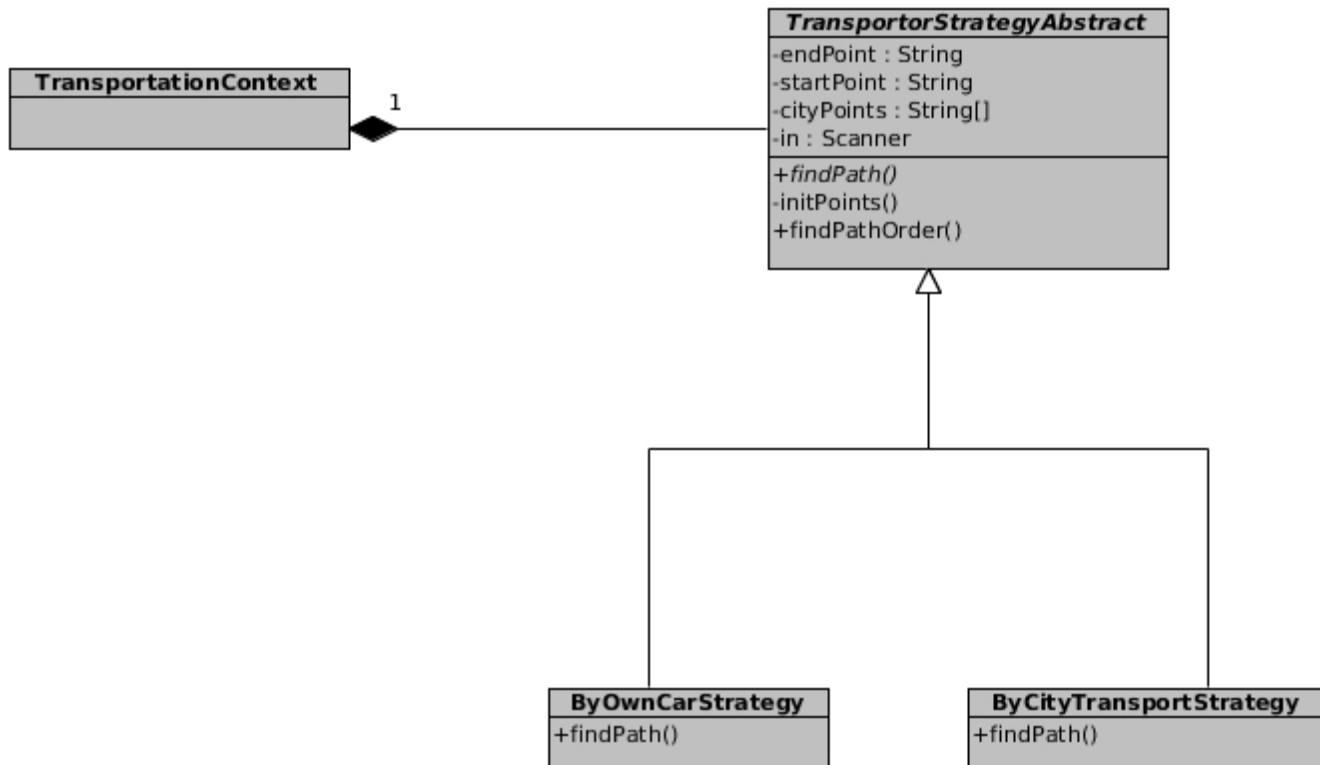
3. Visitor design pattern:

Представя операции, които се извършват върху елементите на обектна структура. Посетител дава възможност да се дефинира нова операция, без да се променят класовете на елементите, върху които работим.



Задача 1:

Нека погледнем в сорс кода на **transportOrderSystem** папката. Там имаме система за маршрутен указател, чиято цел е от зададени две точки в град София и избор на превозно средство (със собствена кола или с градски транспорт) да предложи алгоритъм на клиента да най-добър маршрут между зададените две точки с избория от него вид транспорт. Тази система не е довършена и имаме за цел да я доимплементираме. Нека си помогнем с предоставената UML диаграма:



Това което имаме е **TransportorStrategyAbstract** и **TransportationContext** класовете. Като в **TransportationContext** ще трябва да допълваме бизнес логика.

- 1.1. Създайте двата подкласа: **ByOwnCarStrategy** и **ByCityTransportStrategy**, които ще имат за цел да предлагат стратегия на потребителя за маршрут от двете избрани крайни точки в София. **ByCityTransportStrategy** ще предложи маршрут, ограничавайки се с това какво предлага софийския градски транспорт, докато другата стратегия няма да има подобни ограничения.
- 1.2. Виждате ли и **Template** метода шаблона, който използвате когато имплементирате под – класовете на стратегията ?
- 1.3. Нека сега и контекста използва съответната стратегия. В класа **TransportationContext** добавете един Enumeration примерно, с който ще определяте от това какво вид транспорт е избрал потребителя и разбира се коя от двете стратегии да вземете (това може да се имплементира като `factory method` шаблона). Добавете и метод който първо ще пита потребителя какъв вид транспорт иска и второ, в зависимост от отговора му (работим с конзола) инициализира конкретната стратегия,

която иска от потребителя двете крайни точки , с които ще борави алгоритъма за намиране на удобен маршрут по между им.

1.4.Тествайте така имплементирана система в **TransportTest** класа.

1.5.Нека сега имплементираме същата система като направим рефакторинг, който да не използва стратегия шаблона, а **State design** шаблона. Това става като 'преместим' Enumeration , който добавихте в TransportationContext класа от него в **TransportorStrategyAbstract** . Т.с. Този клас също така ще и пита потребителя с какъв вид транспорт ще се придвижва между двете точки.

Имайки предвид двете решения какви са недостатъците и предимствата на двете съответно, ако има такива , от гледна точка на кохезия ?

Задача 2:

В папка visitor има Employee клас с прости данни за работника: име, дни отпуска и заплата , с помощта на **Visitor** шаблона, искаме да добавим операция върху този клас, който с последователно подаване на служители да калкулираме общите им свободни дни. Нека си помагаме със следната UML диаграма. Тествайте приложението.

