

Patron Visiteur

<https://www.youtube.com/watch?v=pL4mOUDI54o&list=PLF206E906175C7E07&index=26>

<http://www.newthinktank.com/2012/11/visitor-design-pattern-tutorial/>

```
@startuml
interface Visitor {

    // Created to automatically use the right
    // code based on the Object sent
    // Method Overloading
    + double visit(Liquor liquorItem);
    + double visit(Tobacco tobaccoItem);
    + double visit(Necessity necessityItem);
}
class TaxVisitor implements Visitor
class TaxHolidayVisitor implements Visitor

interface Visitable {

    // Allows the Visitor to pass the object so
    // the right operations occur on the right
    // type of object.

    // accept() is passed the same visitor object
    // but then the method visit() is called using
    // the visitor object. The right version of visit()
    // is called because of method overloading

    + double accept(Visitor visitor);

}

class Liquor implements Visitable
class Necessity implements Visitable
class Tobacco implements Visitable

VisitorTest --> TaxVisitor: taxCal
VisitorTest --> TaxHolidayVisitor: taxHolidayCalc

VisitorTest --> Necessity: milk
VisitorTest --> Liquor: vodka
VisitorTest --> Tobacco: cigars

@enduml
```