

In class exercise (I)

Problem specification:

An e-commerce company based in Maryland, U.S. called Dogwood, needs an online sale system. You are asked to write the part of the system to deal with sale order.

Stage One

- Step I. Create the class library for sale order

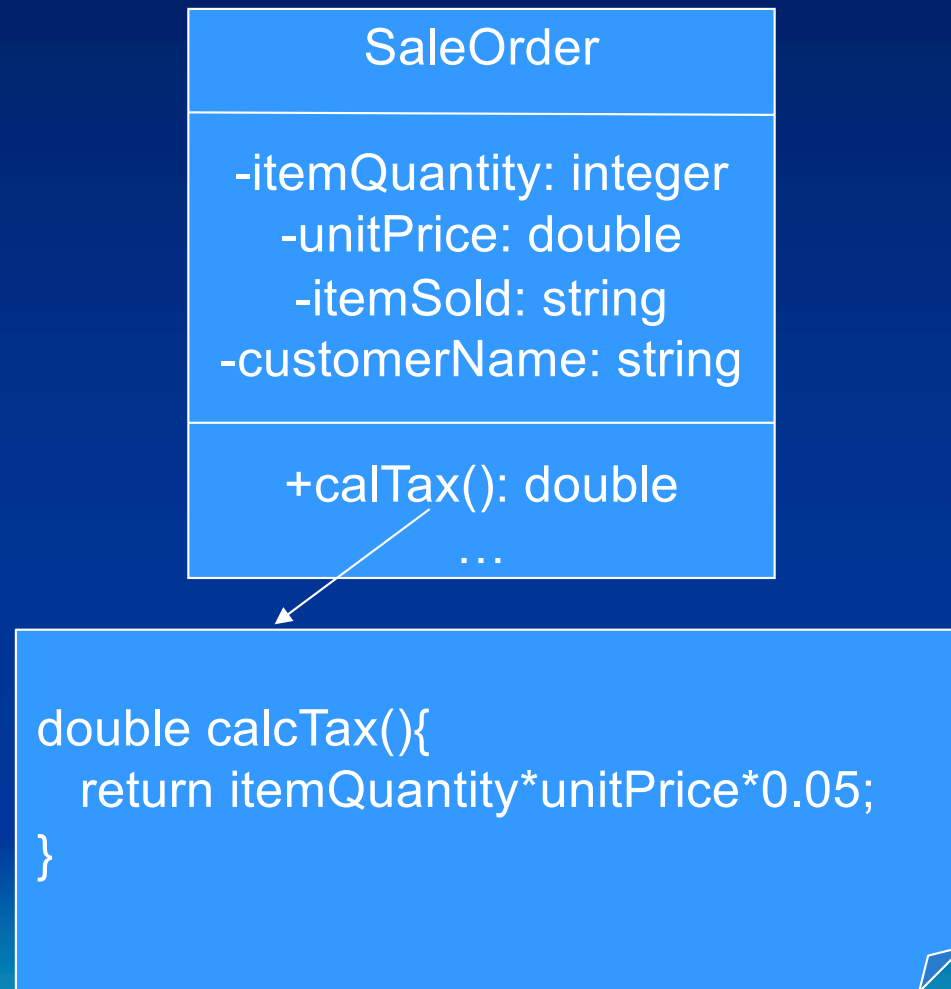
Implement the sale order class according to the following specification. Please note that this class is not responsible for getting input from users. You must assume that the information needed is passed in through its constructor.

| Class | Attributes | Operations |
|-----------|--|---|
| SaleOrder | customName, itemSold, itemQuantity, unitPrice | Calculate price before tax Calculate tax amount Calculate total price Display a receipt with custom name, item sold, quantity, unit price, tax amount and total price. |

- Step II. Write an application using the sale order library

Write an application to test the above class you create. Basic things you need to do are to create an object of SaleOrder and to see if the receipt is displayed properly.

First SaleOrder Design



Result of Stage One:

- Your library works so well for Dogwood
- Another two e-commerce companies based in Maryland, Elm and Holly, bought your library and used it in their on-line sale application as well.



Stage Two:

- Requirement changes:

Dogwood's business is booming and they decide to extend their business to Delaware.

You are asked to rewrite the sale order library to deal with the taxation rule with both Maryland and Delaware.

- Step I. Changing requirement leading to changing sale order library
- Step II. Write an application using the new sale order library

Write an application to test the above class you create. Basic things you need to do are to create an object of SaleOrder and to see if the receipt is displayed properly for both Maryland and Delaware customers.

Result of Stage Two:

- Your library works so well for Dogwood's business in Delaware and Maryland.
- You also sold your new library to an Delaware and Maryland e-commerce firm called Chestnut.
- Due to your change to sale order library, Elm and Holly are affected.



Stage Three:

- Requirement changes:

Dogwood's business is booming again and they decide extend their business to the rest of U.S. *You are asked to rewrite the sale order library to deal with the taxation rule with all the states in U.S..*

- Step 0. Before we do anything, we need to sit back and think first.



Requirement
changes

What got
us into this
mess?

design



How can
we make
our
design
flexible?

Identify the
changing
part

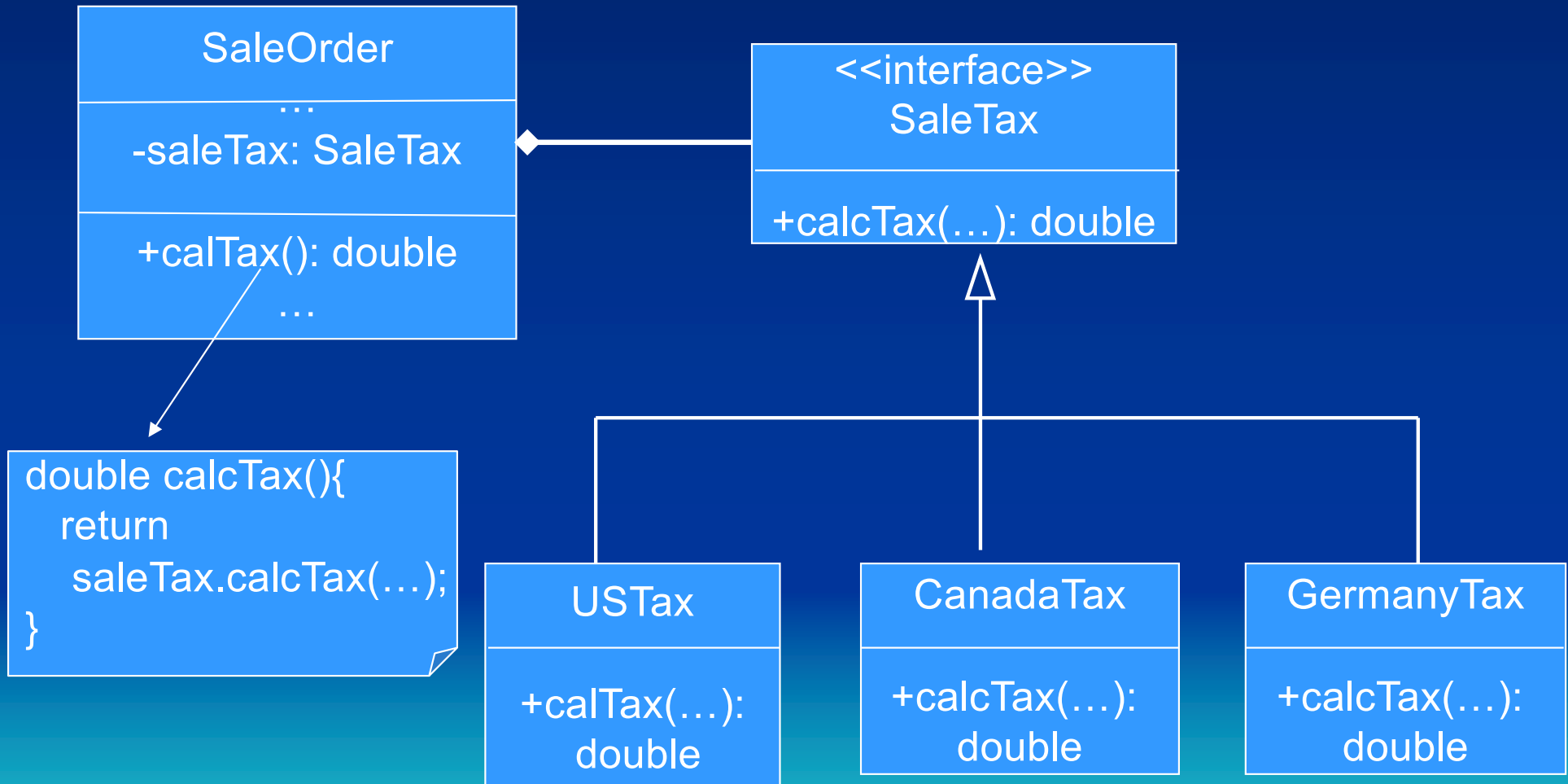
Separate
changing
part

Implement
changes
without
affecting the
existing

What is
changing in
sale order?

tax and
how tax is
calculated

New SaleOrder Design



- Step 1. Implement the new sale order class according to the new design



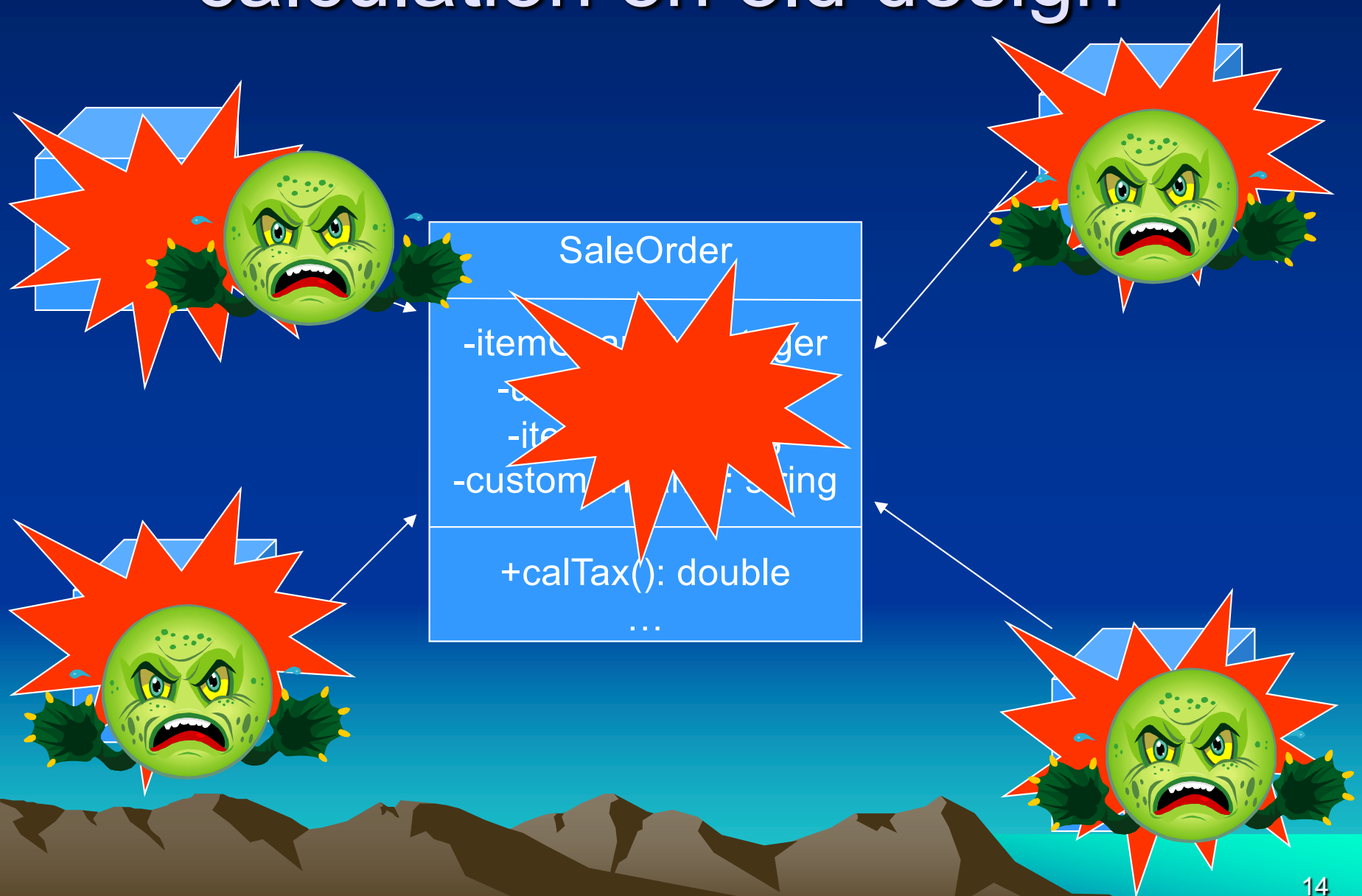
- Step II. Write an application using the new sale order library

Write an application to test the above class you create. Your application should be something like:

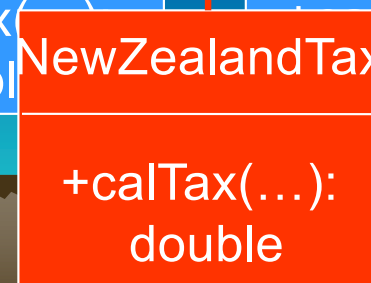
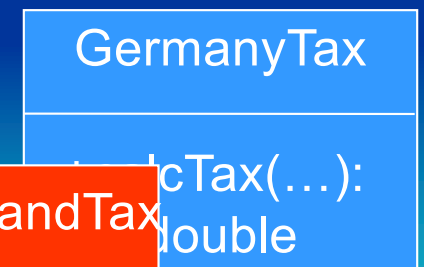
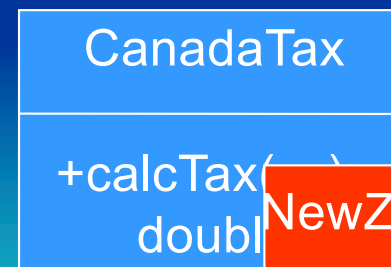
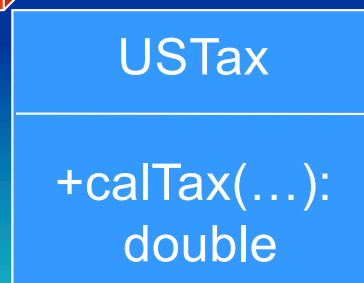
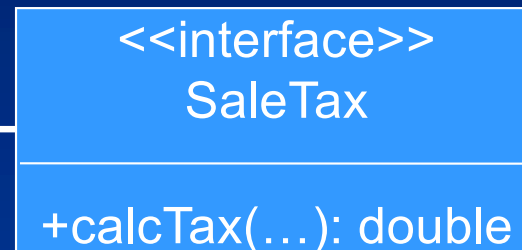
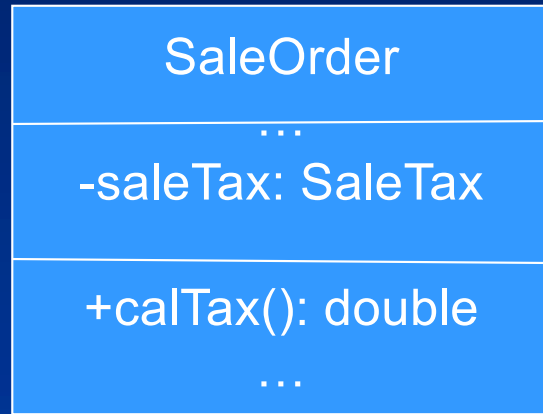
```
SaleOrder saleOrder1 =  
    new saleOrder("Joe", "appletSeed", 10, 1.0, new USTax());
```

```
SaleOrder saleOrder2 =  
    new saleOrder("Mary", "roseTree", 2, 25.0, new CanadaTax());
```

Change requirement on tax calculation on old design



Change requirement on tax calculation on new design

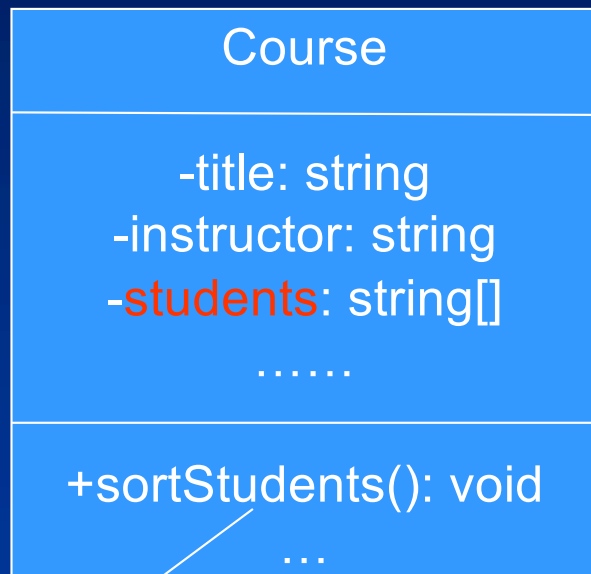


In class exercise (II)

Problem One Specification:

1. *You and your friend started a company to deliver software solution to customer problem.*
2. *Your first business is from Salisbury University Registrar's office. They are using a class library to keep course information. Their current class library allow to sort the students list using a selection sort.*
3. *Now they ask you to improve the design to allow different sorting strategy being used.*

Current Course Class Design



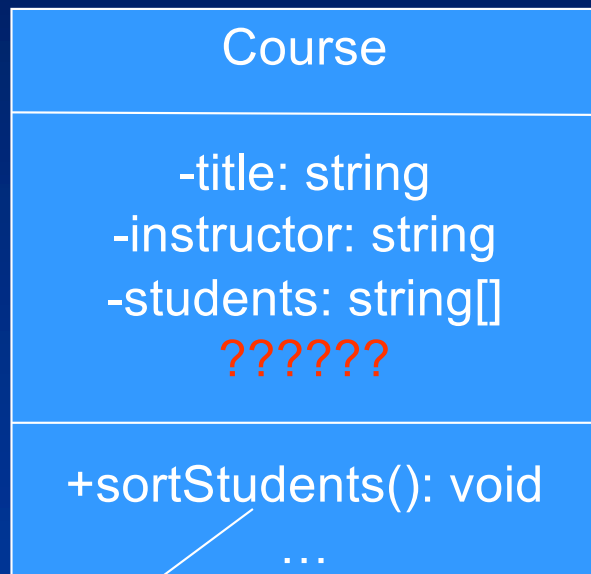
```
void sortStudents(){
    // selection sort of students array

    ... ..
}
```

Application

```
Course calcII = new Course(...);
...
calcII.sortStudents();
...
```

New Course Class Design



```
void sortStudents(){  
    // sort of students array  
  
    ??????  
  
}
```

Application



```
Course calcI = new Course(...,  
    new BubbleSort());  
  
or  
calcI.setSortStrategy(new BubbleSort());  
  
calcI.sortStudents();  
...
```

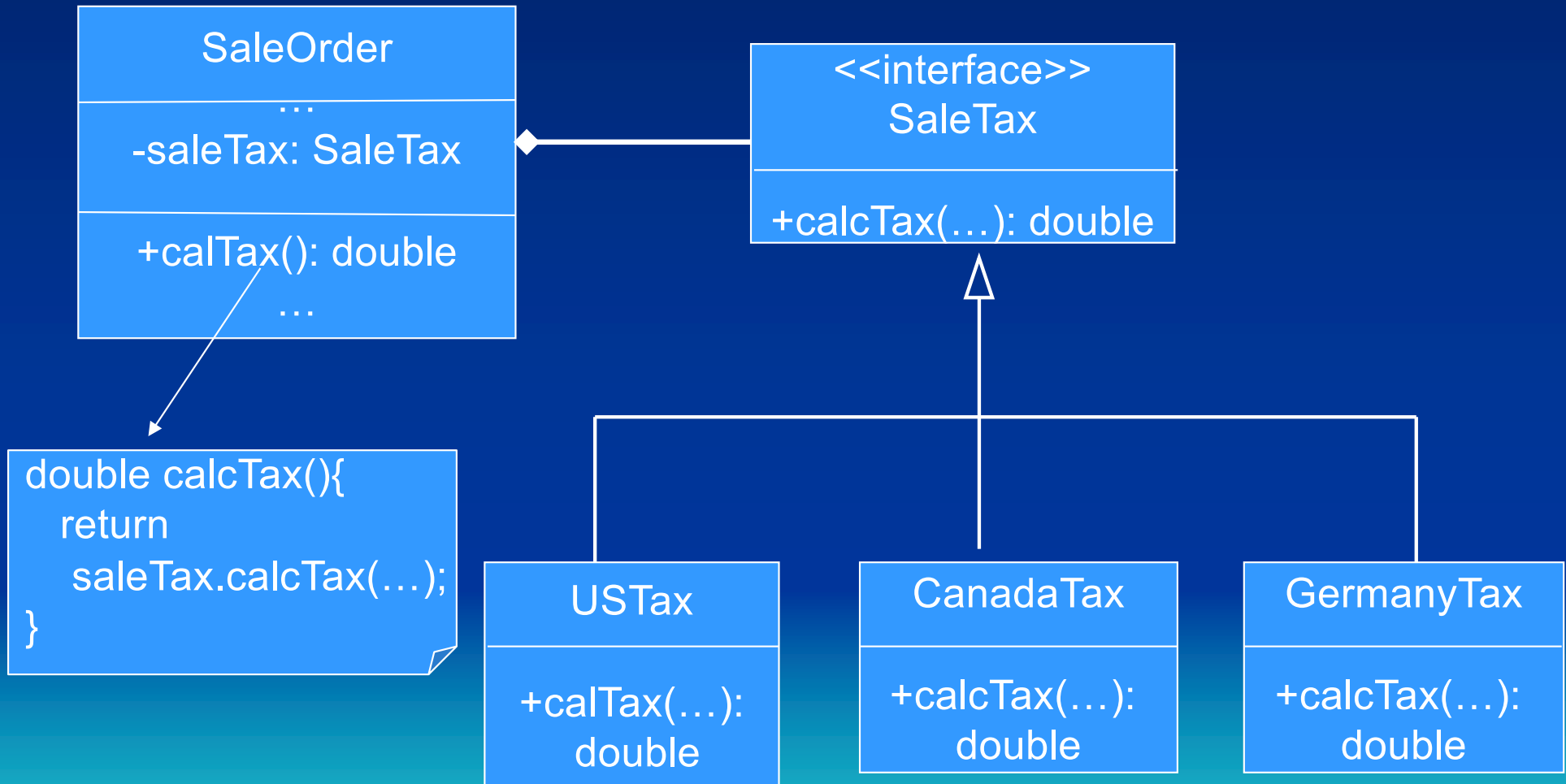
Problem Two Specification:

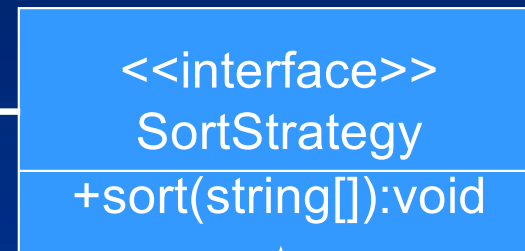
1. *Your second business is from Perdue Stadium Ticket Office. They asked you to write a class library to calculate ticket sale price.*
2. *Here are the rule for ticket sale at Perdue Stadium:*
 - *Ticket sale price consist of two parts: ticket price and sale tax.*
 - *Ticket price also varies between children, and adult.*

Your view

- Short sighted with quick and dirt approach
- Long term with solid and clear approach

SaleOrder Design





```
void sortStudents(){  
    // sort of students array  
    sortStrategy.sort(students);  
}
```

