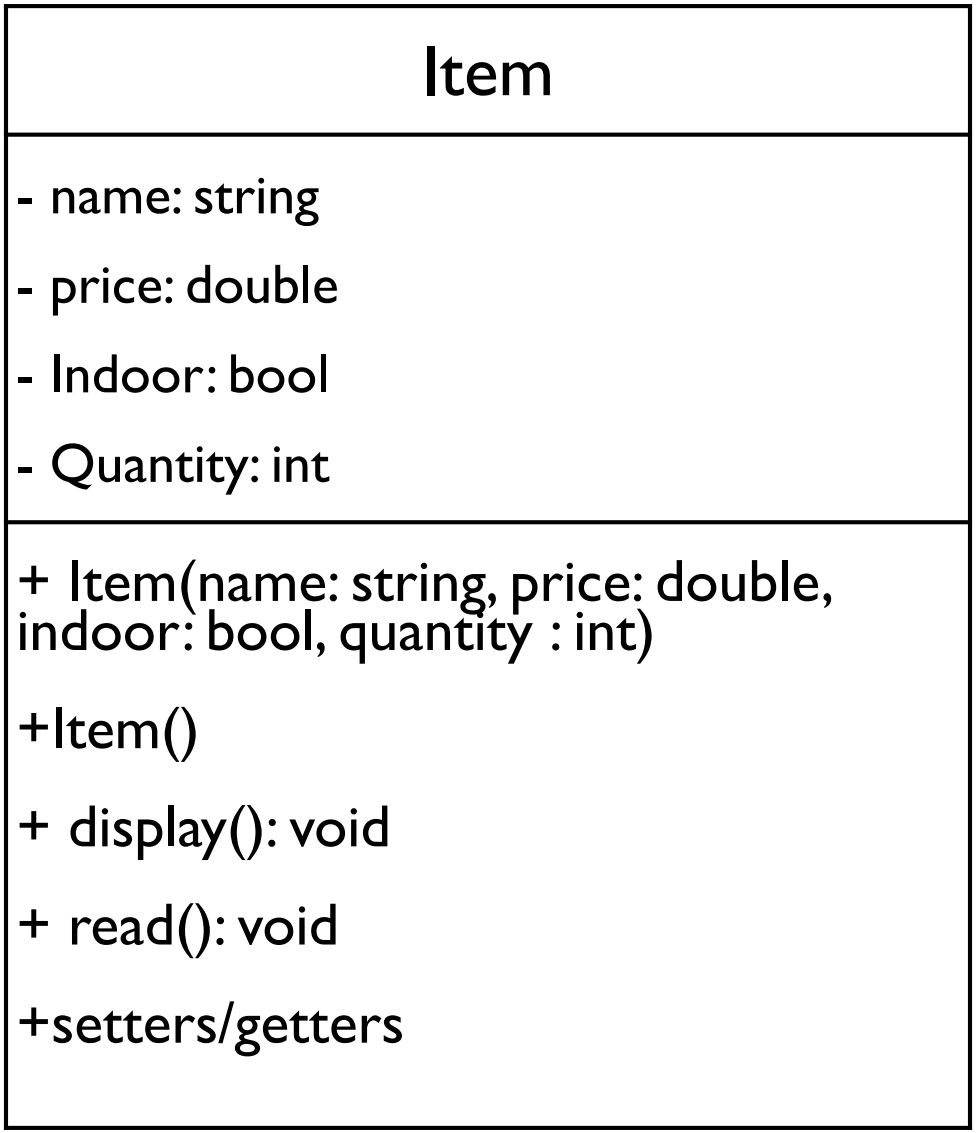
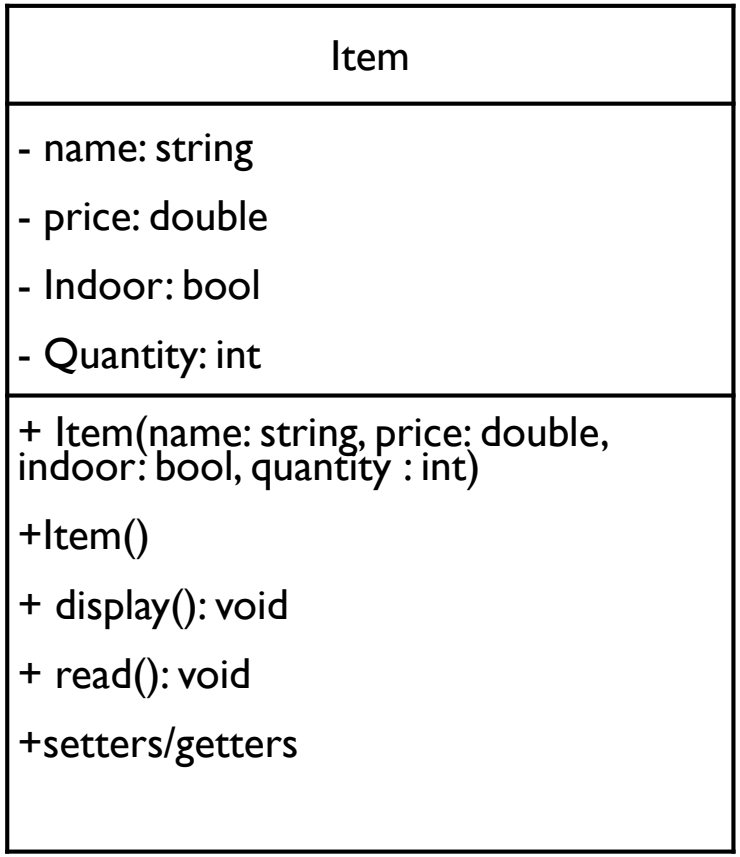
CSC 113
Tutorial 2

Array of objects


Item
- name: string
- price: double
- Indoor: bool
- Quantity: int
+ Item(name: string, price: double, indoor: bool, quantity : int)
+Item()
+ display(): void
+ read(): void
+setters/getters

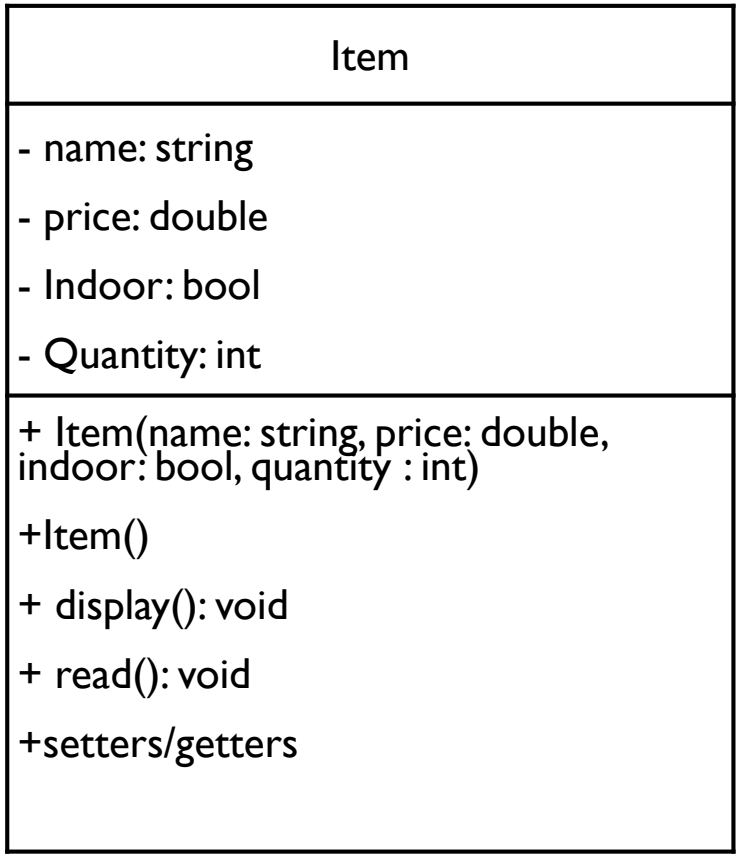
-name: like table, TV, etc.
-price: the item's price.
-Indoor: indicates if the item is used inside the home or outdoor.
-Quantity: number of items in stock.

```
public class Item {

    private String name;
    private double price;
    private boolean indoor;
    private int quantity;



}
```



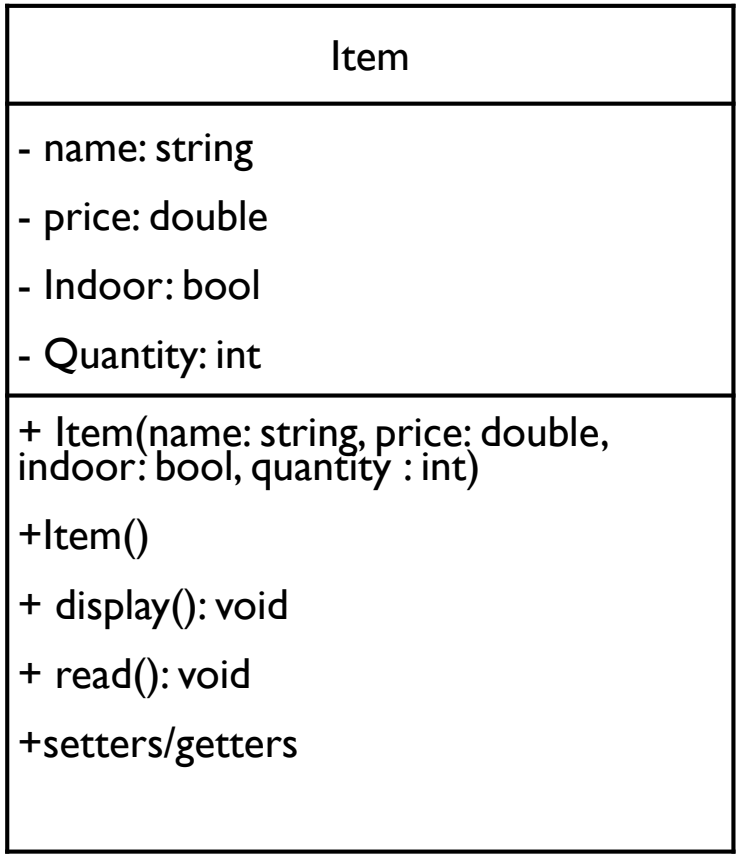
Constructor 1

```
public Item(String name,
     double price,
      boolean indoor,
      int quantity)
      {
            this.name = name;
            this.price = price;
            this.indoor = indoor;
            this.quantity = quantity;
}
```



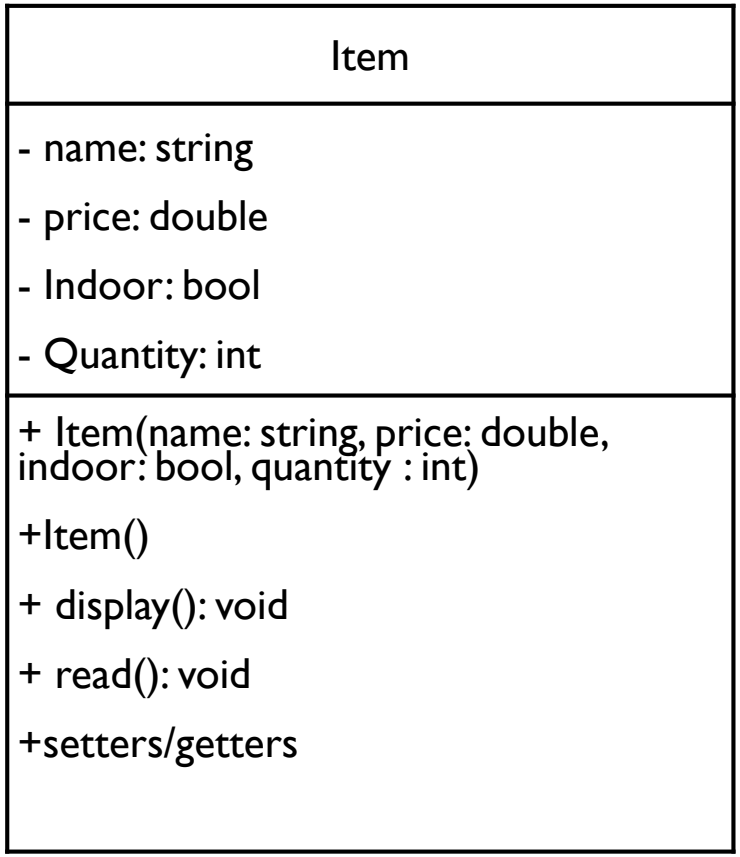
+display: this method prints all the attributes of the Item object.

```
public void display()
   {
       System.out.println(
        "name: "+name
              +" price: "+price
              +" indoor: "+indoor
              +" quantity: "+quantity);
   }
```



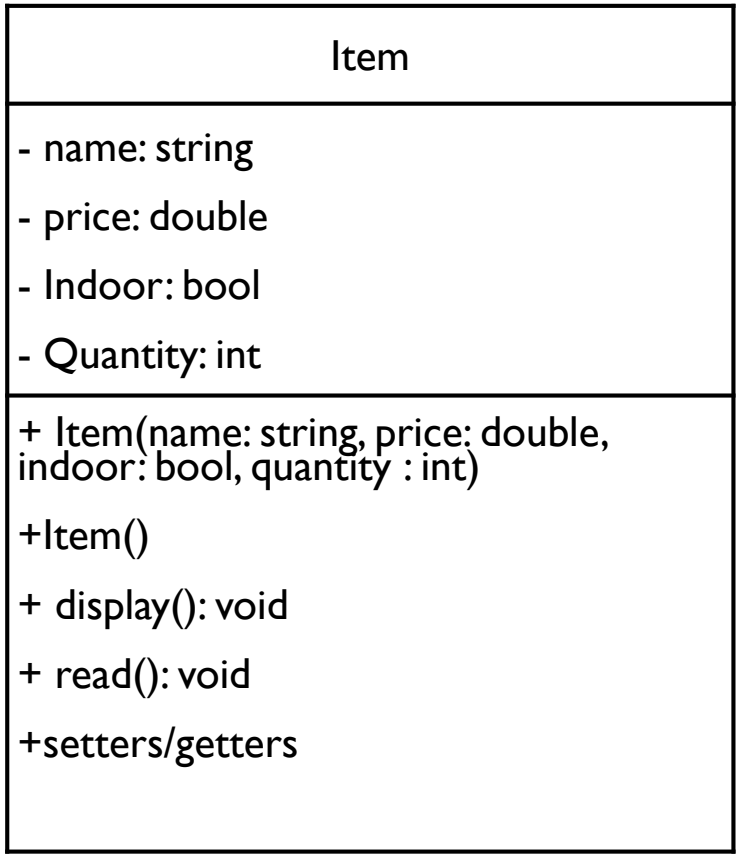
Constructor 2

```
public Item()
    {
        name = "";
        price = -1;
        indoor = false;
        quantity = -1;
    }
```



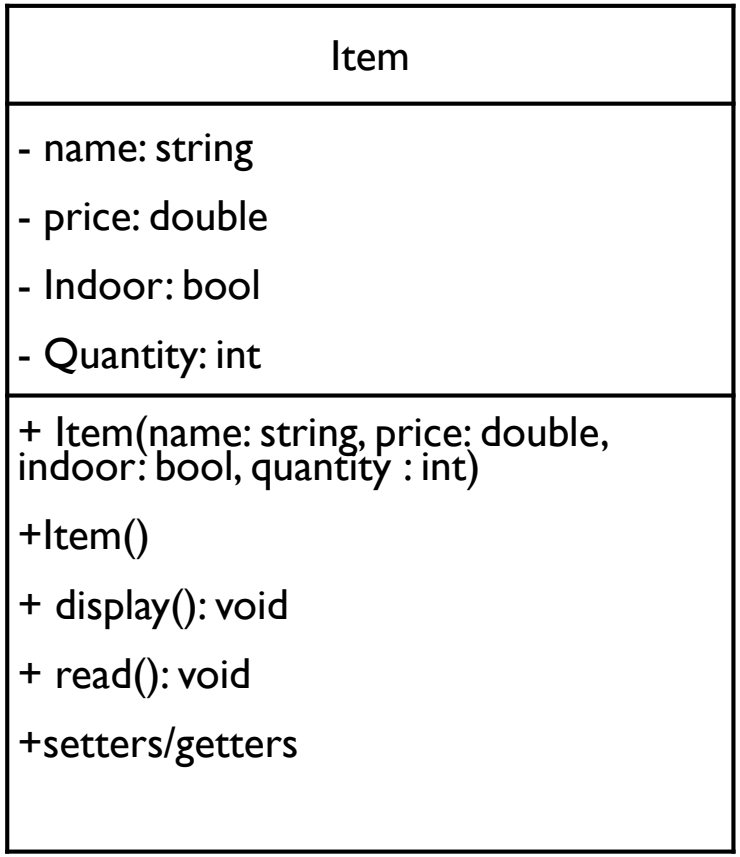
+read: this method reads the attributes of the Item object from the keyboard.

```
public void read()
    {
        Scanner s = new Scanner(System.in);
        print("Please, Eneter name: ");
        name = s.next();
        print("Eneter price: ");
        price = s.nextDouble();
        print("indoor?(true/false) ");
        indoor = s.nextBoolean();
        print("Eneter quantity: ");
        quantity = s.nextInt();

    }
```



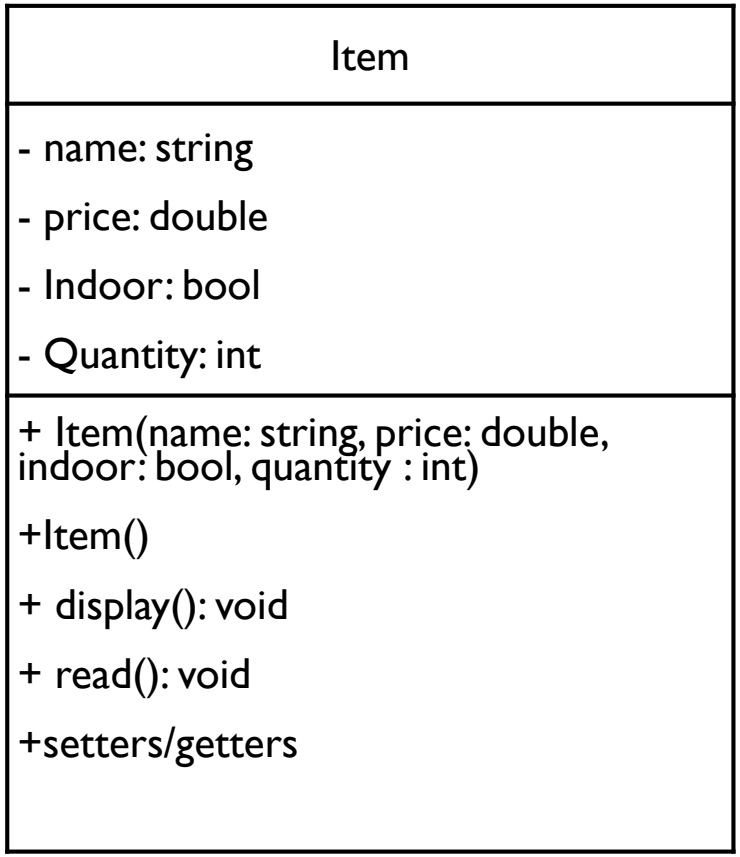
+setters/getters

```
public void setName(String name)
  {
      this.name = name;
  }

  public void setPrice(double price)
  {
      this.price = price;
  }

  public void setIndoor(boolean indoor)
  {
      this.indoor = indoor;
  }
```



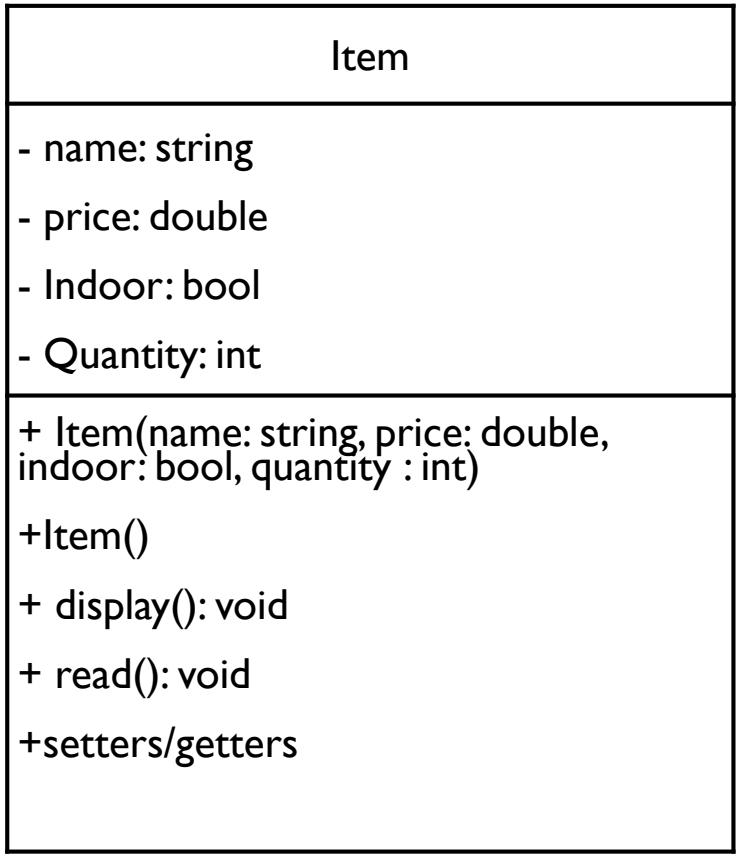
+setters/getters

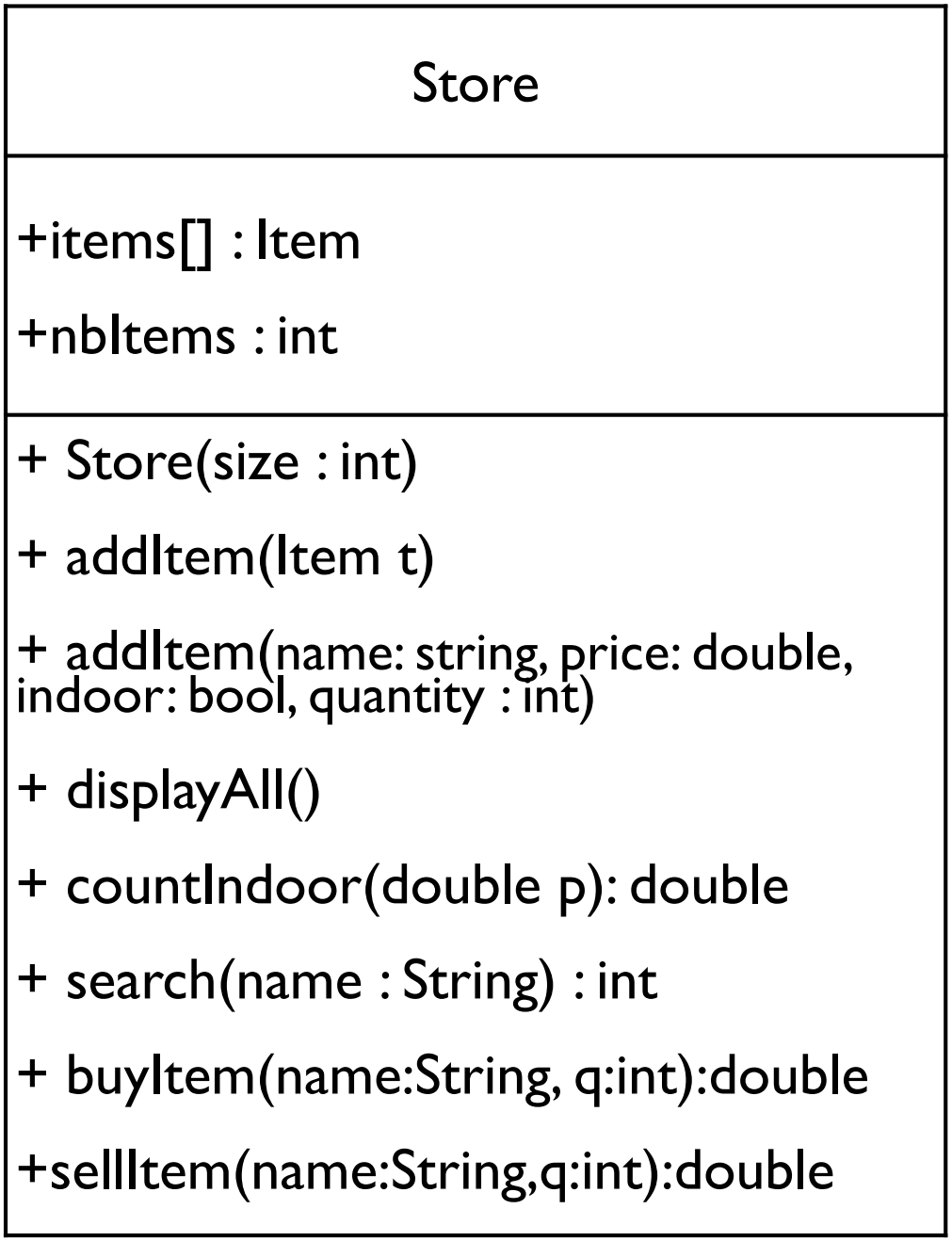
```
public void setQuantity(int quantity)
  {
      this.quantity = quantity;
  }

  public double getPrice()
  {
      return price;
  }

  public String getName()
  {
      return name;
  }
```



+setters/getters

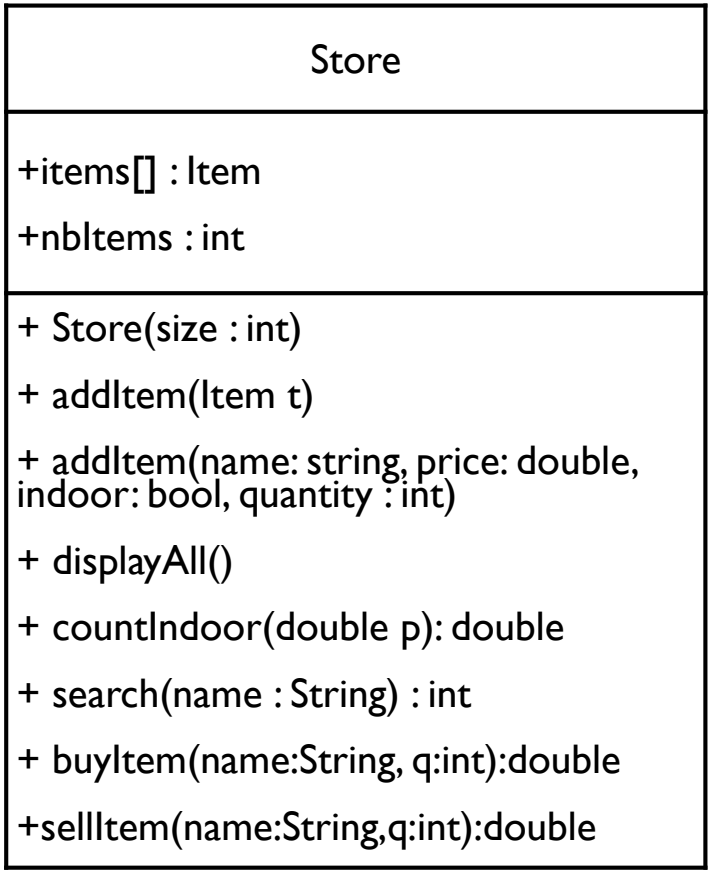
```
public boolean getIndoor()
{
    return indoor;
}

public int getQuantity()
{
    return quantity;
}
```




Store
+items[] : Item
+nbItems : int
+ Store(size : int)
+ addItem(Item t)
+ addItem(name: string, price: double, indoor: bool, quantity : int)
+ displayAll()
+ countIndoor(double p): double
+ search(name : String) : int
+ buyItem(name:String, q:int):double
+sellItem(name:String,q:int):double

+items: contains Item objects.
+nbItems: equals the number of elements in items.

```
public class Store {
    public Item[] items;
    public int nbItems;

}
```



Constructor

```
public Store(int size)
   {
      items = new Item[size];
      nbItems = 0;
   }
```

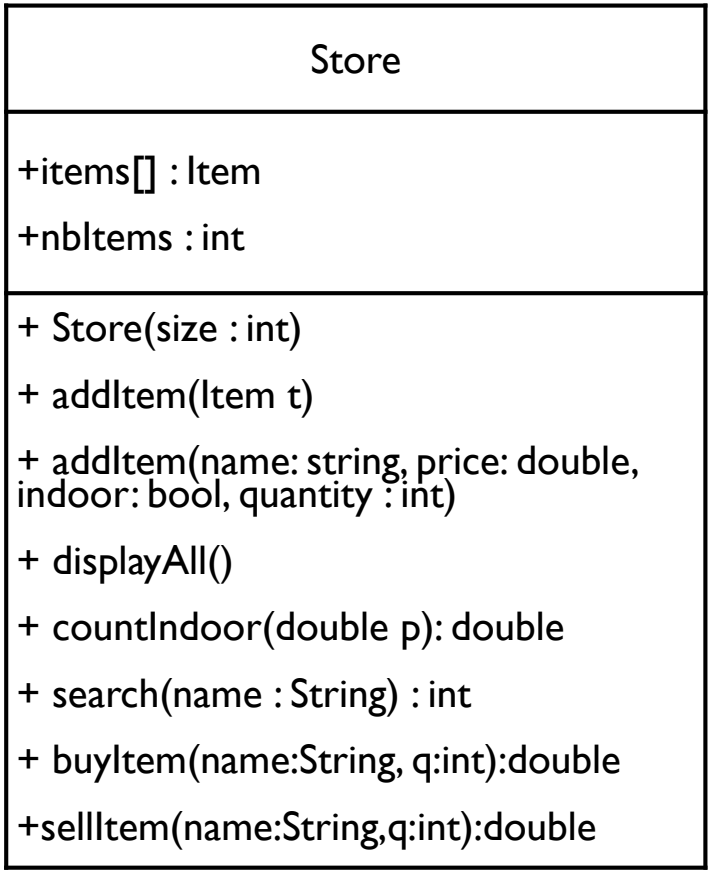
Store
+items[] : Item +nbItems : int
+ Store(size : int) + addItem(Item t) + addItem(name: string, price: double, indoor: bool, quantity : int) + displayAll() + countIndoor(double p): double + search(name : String) : int + buyItem(name:String, q:int):double +sellItem(name:String,q:int):double

+addItem(Item t): this method adds the item t to the array.

```
public boolean addItem(Item t)
   {
      if(nbItems < items.length)
      {
         items[nbItems++]=t;
         return true;
      }
      return false;
   }
```

Store
+items[] : Item +nbItems : int
+ Store(size : int) + addItem(Item t) + addItem(name: string, price: double, indoor: bool, quantity : int) + displayAll() + countIndoor(double p): double + search(name : String) : int + buyItem(name:String, q:int):double +sellItem(name:String,q:int):double

+addItem(name: string, price: double, indoor: bool, quantity : int)
: this method adds a new item with the given attributes to the array.




```
public boolean addItem(String name,
      double price,
      boolean indoor,
      int quantity)
   {
      if(nbItems < items.length)
      {
         items[nbItems++] = new Item(name, price, indoor, quantity);
         return true;
      }
      return false;
}
```

+displayAll: display all elements of the array. In this case all items.

```
public void displayAll()
    {
        int i;
      System.out.println("---------------");
        for(i=0; i<nbItems; i++)
            items[i].display();
    }
```

Store
+items[] : Item +nbItems : int
+ Store(size : int) + addItem(Item t) + addItem(name: string, price: double, indoor: bool, quantity : int) + displayAll() + countIndoor(double p): double + search(name : String) : int + buyItem(name:String, q:int):double +sellItem(name:String,q:int):double

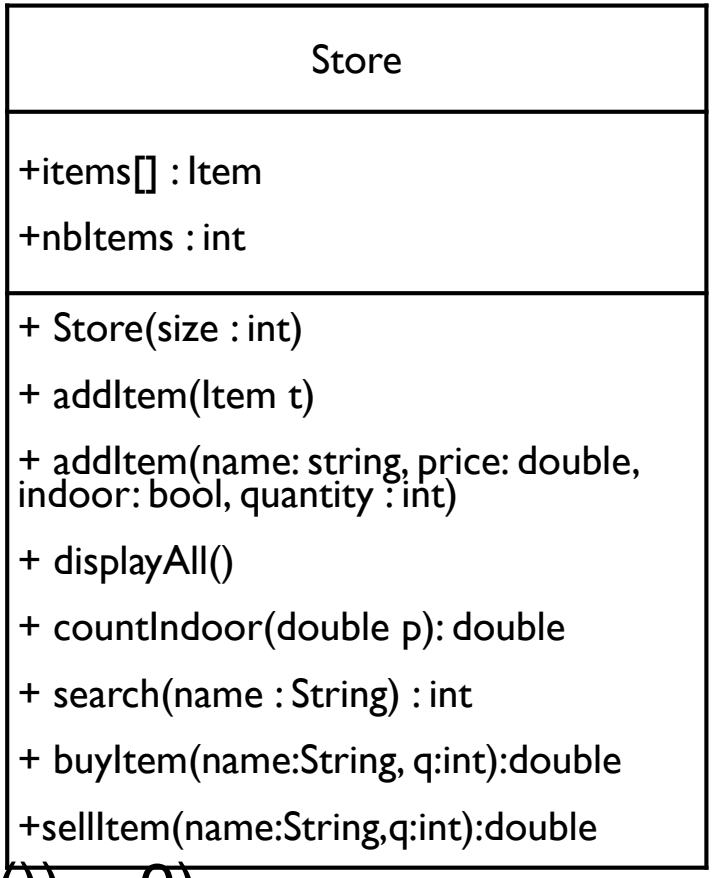
+countIndoor(double p): integer: this method provides the number of items that are indoor and its price less than p.

```
public int countIndoor(double p)
   {
      int i,indoorItems=0;
      for(i=0; i<nbItems; i++)
         if(items[i].getPrice() < p
               && items[i].getIndoor())
            indoorItems++;

      return indoorItems;
   }
```

Store
+items[] : Item +nbItems : int
+ Store(size : int) + addItem(Item t) + addItem(name: string, price: double, indoor: bool, quantity : int) + displayAll() + countIndoor(double p): double + search(name : String) : int + buyItem(name:String, q:int):double +sellItem(name:String,q:int):double

+Search(name : String) : int
will return the index of the item with the same name. If not found it will return -1.

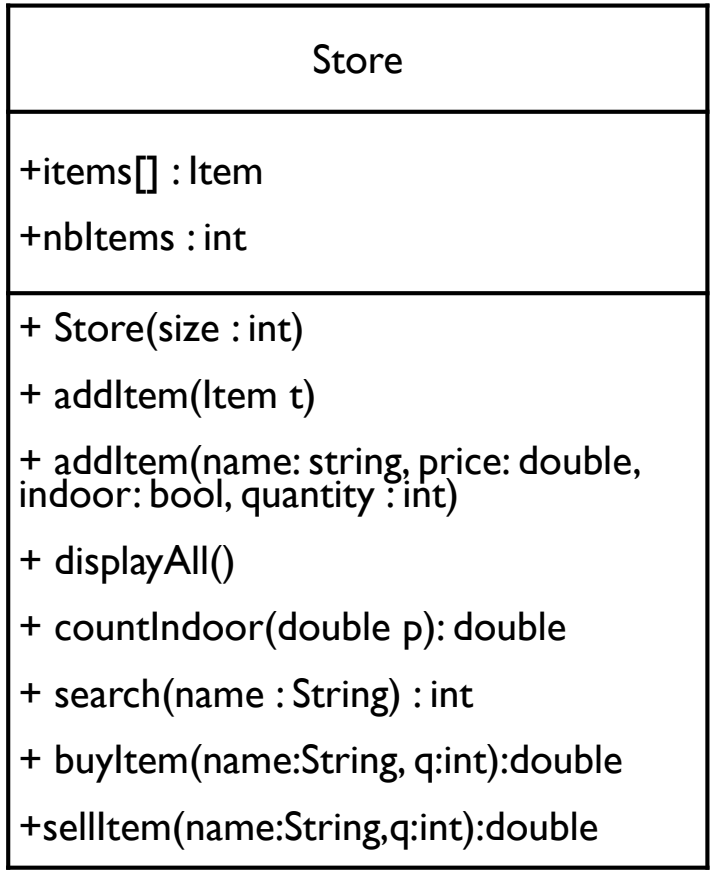
```
public int search(String name)
    {
        int i;

        for(i=0; i<nbItems; i++)
            if(name.compareTo(items[i].getName())==0)
                return i;

        return -1;
    }
```



CompareTo

- returns the difference between the first different characters in both strings.
- If they are equal. it returns the difference in length.
- if they are equal in chars and length it returns 0.

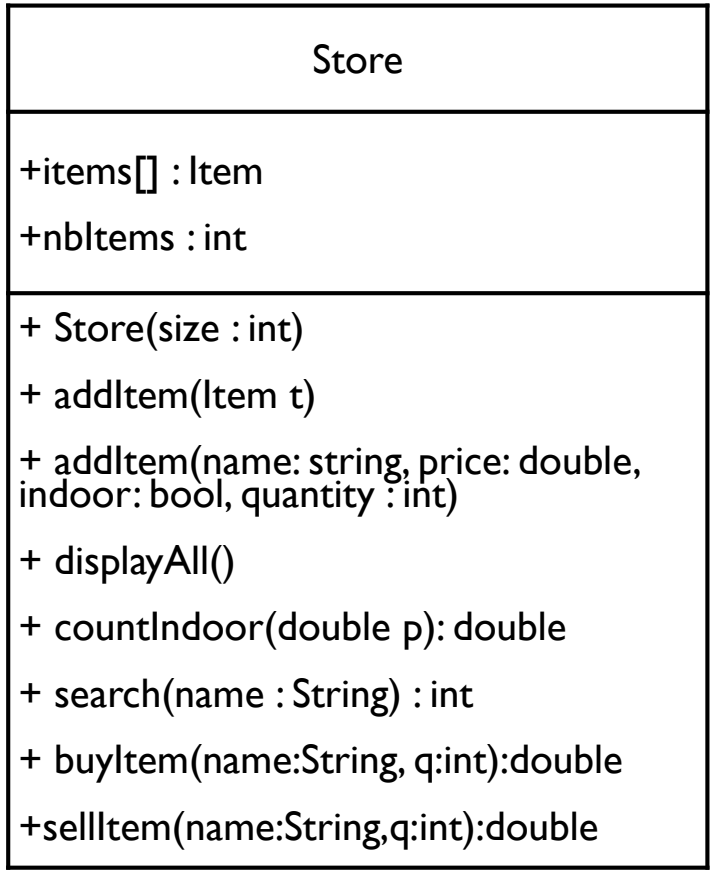
+buyItem(name: String, q:int): we use it if we want to buy a quantity "q" of item "name". It will return the total price that we must pay to the provider.




```
public double buyItem(String name, int q)
    {
        double price;
        int i = search(name);
        if(i == -1)
            return -1;

        price = items[i].getPrice()*q;
        items[i].setQuantity(items[i].getQuantity()+q);
        return price;
    }
```

+sellItem(name: String, q: int): we use it if we want to sell a quantity "q" of item "name". It will return the total price that we must get from the customer.

```
public double sellItem(String name, int q)
    {
        double price;
        int i = search(name);
        if(i == -1)
            return -1;

        if(q > items[i].getQuantity())
            q=items[i].getQuantity();

        price = items[i].getPrice()*q;
        items[i].setQuantity(items[i].getQuantity()-q);
        return price;
    }
```



Main

```
public static void main(String[] args) {
	// TODO Auto-generated method stub
	Store myStore = new Store(25);

	myStore.addItem("table1", 200.0, true, 6);
	myStore.displayAll();

	Item t1 = new Item("chair", 50.0, true, 20);
	myStore.addItem(t1);
	myStore.displayAll();

	Item t2 = new Item();
	t2.read();
	myStore.addItem(t2);
	myStore.displayAll();

	double price = 100;
	System.out.println("\nNumber of items indoor with price less than ("+price+") is
"+myStore.countIndoor(price));
```

Output

```
------------------------
name: table1 price: 200.0 indoor: true quantity: 6
------------------------
name: table1 price: 200.0 indoor: true quantity: 6
name: chair price: 50.0 indoor: true quantity: 20
Please, Eneter name: aaa
Eneter price: 90
indoor?(true/false) false
Eneter quantity: 90
------------------------
name: table1 price: 200.0 indoor: true quantity: 6
name: chair price: 50.0 indoor: true quantity: 20
name: aaa price: 90.0 indoor: false quantity: 90

Number of items indoor with price less than (100.0) is 1
```

Main

```
String name = "table1";
int q = 5;
price = myStore.buyItem(name, q);
if(price != -1)
    System.out.println("\nyou have bought "+q+" of item "+name+" for "+price);
else
    System.out.println("Error: item does not exist");
myStore.displayAll();

name = "table";
q = 5;
price = myStore.buyItem(name, q);
if(price != -1)
    System.out.println("\nyou have bought "+q+" of item "+name+" for "+price);
else
    System.out.println("Error: item does not exist");

myStore.displayAll();
```

Output

you have bought 5 of item table1 for 1000.0
---
name: table1 price: 200.0 indoor: true quantity: 11
name: chair price: 50.0 indoor: true quantity: 20
name: aaa price: 90.0 indoor: false quantity: 90
Error: item does not exist
---
name: table1 price: 200.0 indoor: true quantity: 11
name: chair price: 50.0 indoor: true quantity: 20
name: aaa price: 90.0 indoor: false quantity: 90

Main

```
 name = "chair";
    q = 5;
    price = myStore.sellItem(name, q);
    if(price != -1)
        System.out.println("\nyou have sold "+q+" of item "+name+" for "+price);
    else
        System.out.println("Error: item does not exist");
    myStore.displayAll();

    name = "chai";
    q = 5;
    price = myStore.sellItem(name, q);
    if(price != -1)
        System.out.println("\nyou have sold "+q+" of item "+name+" for "+price);
    else
        System.out.println("Error: item does not exist");
}
```

Output

you have sold 5 of item chair for 250.0
---
name: table1 price: 200.0 indoor: true quantity: 11
name: chair price: 50.0 indoor: true quantity: 15
name: aaa price: 90.0 indoor: false quantity: 90
Error: item does not exist