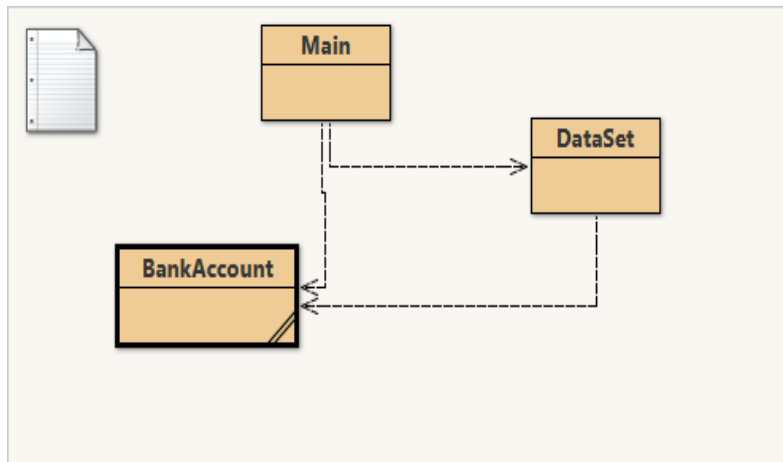


Abstract classes- Interfaces

no_abstraction



```
public class DataSet {
```

```
    private double sum;  
    private BankAccount max;  
    private BankAccount min;  
    private int counter;
```

```
    public DataSet() {  
        sum = 0;  
        counter = 0;  
        max = null;  
        min = null;  
    }
```

```
    public void add(BankAccount item) {  
        sum = sum + item.getBalance();  
  
        if(counter == 0) {  
            max = item;  
            min = item;  
        }  
        else if(item.getBalance() > max.getBalance())  
            max = item;  
        else if(item.getBalance() < min.getBalance())  
            min = item;  
  
        counter++;  
    }
```

No Abstraction

```
public class BankAccount {
```

```
    private double balance;  
  
    public BankAccount(double amount) {  
        balance = amount;  
    }  
  
    public double getBalance() {  
        return balance;  
    }  
}
```

```
    public double getAverage() {  
        if(counter == 0)  
            return 0;  
        else  
            return sum/counter;  
    }  
  
    public BankAccount getMaximum() {  
        return max;  
    }  
  
    public BankAccount getMinimum() {  
        return min;  
    }  
}
```

```
public class Main {
```

No Abstraction

```
    public static void main(String[] args) {  
        DataSet ds = new DataSet();
```

```
        BankAccount BA1 = new BankAccount(1500);  
        BankAccount BA2 = new BankAccount(2000);  
        BankAccount BA3 = new BankAccount(1000);
```

```
        ds.add(BA1);  
        ds.add(BA2);  
        ds.add(BA3);
```

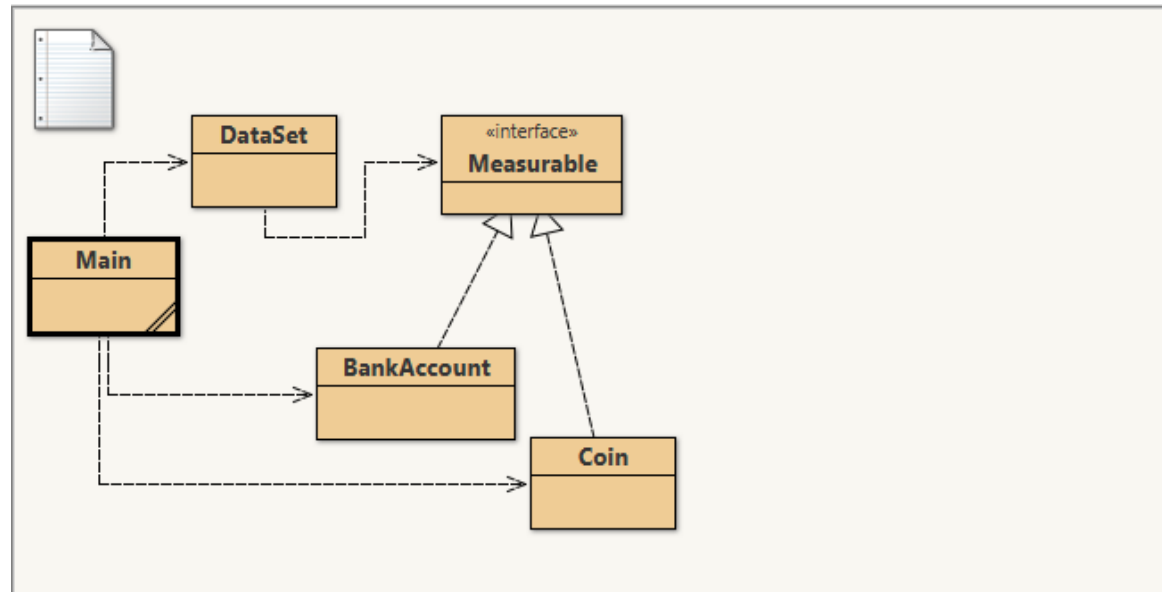
```
        System.out.println("Average is: " + ds.getAverage());  
        System.out.println("Maximum is: " + ds.getMaximum().getBalance());  
        System.out.println("Minimum is: " + ds.getMinimum().getBalance());
```

```
    }
```

```
}
```

Average is: 1500.0
Maximum is: 2000.0
Minimum is: 1000.0

....interface ...προσθέτω και coins



Abstraction

```
public interface Measurable {  
  
    double getMeasure();  
  
}
```

```
public class BankAccount implements Measurable {
```

```
    private double balance;
```

```
    public BankAccount(double amount) {  
        balance = amount;  
    }
```

```
    public double getBalance() {  
        return balance;  
    }
```

```
    public double getMeasure() {  
        return getBalance();  
    }
```

```
}
```

```
public class Coin implements Measurable {
```

```
    private double value;
```

```
    public Coin(double amount) {  
        value = amount;  
    }
```

```
    public double getValue() {  
        return value;  
    }
```

```
    public double getMeasure() {  
        return value;  
    }
```

```
}
```

Abstraction

```
public class DataSet {
```

```
    private double sum;  
    private Measurable max;  
    private Measurable min;  
    private int counter;
```

```
    public DataSet() {  
        sum = 0;  
        counter = 0;  
        max = null;  
        min = null;}  
}
```

```
    public void add(Measurable item) {  
        sum = sum + item.getMeasure();  
  
        if(counter == 0) {  
            max = item;  
            min = item;  
        }  
        else if(item.getMeasure() > max.getMeasure())  
            max = item;  
        else if(item.getMeasure() < min.getMeasure())  
            min = item;  
  
        counter++;  
    }  
}
```

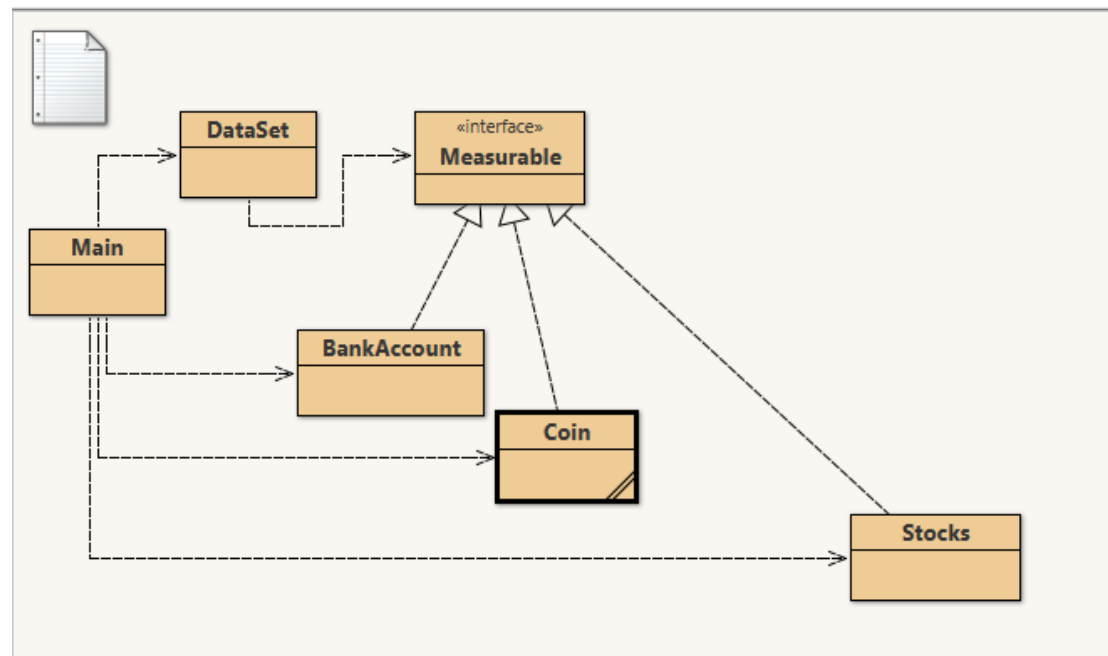
```
    public double getAverage() {  
        if(counter == 0)  
            return 0;  
        else  
            return sum/counter;  
    }  
  
    public Measurable getMaximum() {  
        return max;  
    }  
  
    public Measurable getMinimum() {  
        return min;  
    }  
}
```

Abstraction

```
public class Main {  
  
    public static void main(String[] args) {  
        DataSet ds = new DataSet();  
  
        BankAccount BA1 = new BankAccount(1500);  
        BankAccount BA2 = new BankAccount(2000);  
        BankAccount BA3 = new BankAccount(1000);  
        ds.add(BA1);  
        ds.add(BA2);  
        ds.add(BA3);  
  
        Coin C1 = new Coin(50);  
        Coin C2 = new Coin(25);  
        Coin C3 = new Coin(15.27);  
  
        ds.add(C1);  
        ds.add(C2);  
        ds.add(C3);  
  
        System.out.println("Average is: " + ds.getAverage());  
        System.out.println("Maximum is: " + ds.getMaximum().getMeasure());  
        System.out.println("Minimum is: " + ds.getMinimum().getMeasure());  
    }  
}
```

Average is: 765.0450000000001
Maximum is: 2000.0
Minimum is: 15.27

.... Προσθέτω και stocks



Abstraction...συνέχεια ...προσθέτω μια κλάση ακόμα

```
public class Stocks implements Measurable {  
    final int value= 100;  
    private double balance;  
  
    public Stocks (double amount) {  
        balance = amount;  
    }  
  
    public double getBalance() {  
        return (balance* value);  
    }  
  
    public double getMeasure() {  
        return getBalance();  
    }  
}
```

```
public class Main {  
  
    public static void main(String[] args) {  
        DataSet ds = new DataSet();  
  
        BankAccount BA1 = new BankAccount(1500);  
        BankAccount BA2 = new BankAccount(2000);  
        BankAccount BA3 = new BankAccount(1000);  
        ds.add(BA1);  
  
        ds.add(BA2);  
        ds.add(BA3);  
  
        Coin C1 = new Coin(50);  
        Coin C2 = new Coin(25);  
        Coin C3 = new Coin(15.27);  
        ds.add(C1);  
        ds.add(C2);  
        ds.add(C3);  
  
        Stocks S1= new Stocks (10);  
        Stocks S2 = new Stocks (5);  
  
        System.out.println( C1 instanceof Measurable);  
        System.out.println("Average is: " + ds.getAverage());  
        System.out.println("Maximum is: " + ds.getMaximum().getMeasure());  
        System.out.println("Minimum is: " + ds.getMinimum().getMeasure());  
    }  
}
```

```
true  
Average is: 765.04500000000001  
Maximum is: 2000.0  
Minimum is: 15.27
```