Bond Class

```
import java.text.DecimalFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class Bond {

    //<!-- variables-->

    private String name;

    private String purchaseDate; //date

    private String maturityDate;

    private double coupon;

    private double frequency;

    private double issuePrice;

    //<!-- variables-->




    //<!---constructor--->


    public Bond(String name, String purchaseDate, String maturityDate, double coupon, double issuePrice, double
frequency) {

        this.name = name;
        this.purchaseDate = purchaseDate;
        this.maturityDate = maturityDate;
        this.coupon = coupon;
        this.issuePrice = issuePrice;
        this.frequency = frequency;

    }

    public Bond() {
    }// default constructor

    //<!---constructor--->


    //<!---getter--->

    public String getName() { return name; }

    public double getFrequency() {
```

```
    return frequency;
}

public double getIssuePrice() {
    return issuePrice;
}

public double getCoupon() {
    return coupon;
} // bond must be stored in decimal (e.g. 50%  -> 0.5)

public double getTerm() {

    /*logic*/

    // maturityDate-purchaseDate

    try {
        Date start;
        Date end;


        SimpleDateFormat dates = new SimpleDateFormat("yyyy/MM/dd");


        start = dates.parse(this.purchaseDate);
        end = dates.parse(this.maturityDate);

        double difference = Math.abs(start.getTime() - end.getTime());
        double differenceByYear = (difference / (24 * 60 * 60 * 1000)) / 365;

        DecimalFormat decf = new DecimalFormat("##.0");

        return Double.parseDouble(decf.format(differenceByYear));

    } catch (ParseException e) {

        return (double) e.getErrorOffset();
    }


}

//<!---getter--->

//<!---setter--->

public void setIssuePrice(double issuePrice) { this.issuePrice = issuePrice; }

public void setCoupon(double coupon) { this.coupon = coupon; }

public void setFrequency(double frequency) { this.frequency = frequency; }

//<!---setter--->
```

}

InvestmentAnalysisSystem Class

```
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Scanner;

public class InvestmentAnalysisSystem {

/*  Deadline: November 30th, 4pm.
*/
    //<!-- variables--->

    private Bond bond;

    private Investor investor;

    private Portfolio portfolio;

    //<!-- variables--->



    public InvestmentAnalysisSystem(Investor investor, Bond bond) {/*default constructor*/

        this.investor = investor;
        this.bond = bond;


    }


    public String getInvestorName() {

        return investor.getName();

    }

    public String getBondName() {

        return bond.getName();
    }

    public int getBondPurchasedNumber() {

        return investor.getNumberOfBonds();
    }

    public double getBondIssuePrice(){

        return bond.getIssuePrice();
    }


    //<!---functions--->
```

```
public double getInvestorBalance(){


    return investor.getBalance();

}

public double payout() {

    double payout = bond.getTerm() * bond.getCoupon() + (100 * bond.getFrequency());
    return payout;
}

public double value(double r) {

    double val = 0;

    for (int i = 1; i <= bond.getTerm(); i++) {

        val += bond.getCoupon() / Math.pow((1 + r), i);
    }
    double actualVal = 100 / Math.pow((1 + r), bond.getTerm());
    val += actualVal;

    return val;

}
public double macaulayDuration(double r) {

    double val1 = 0;
    double val2 = 0;
    for (int i = 1; i <= bond.getTerm(); i++) {
        val1 += i * bond.getCoupon() / Math.pow((1 + r), i);
    }


    val2 = bond.getTerm() * 100 / Math.pow((1 + r), bond.getTerm());


    double val3 = val1 + val2;

    double MacD = val3 / value(r);


    return MacD;



 /*   double val = 0;
    double MacD = 0;

    for (int i = 1; i <= bond.getTerm(); i++) {
            if (i != bond.getTerm()) {

                    val += bond.getCoupon() / Math.pow((1 + r), i);
                    MacD += bond.getCoupon() / Math.pow((1 + r), i) * i;

            } else {

                    val += (bond.getCoupon() + investor.getMoney()) / Math.pow((1 + r), i);
                    MacD += (bond.getCoupon() + investor.getMoney()) / Math.pow((1 + r), i) * i;

            }
    }


    MacD /= val;


    return MacD;
*/

}


public double calculateIrr() {

    double r = 0;




    for( ;bond.getIssuePrice() != value(r); )
    {

        if (value(1 + r) > bond.getIssuePrice()) r += 1;
```

```
            else if (value(0.1 + r) > bond.getIssuePrice()) r += 0.1;
            else if (value(0.01 + r) > bond.getIssuePrice()) r += 0.01;
            else if (value(0.001 + r) > bond.getIssuePrice()) r += 0.001;
            else if (value(0.0001 + r) > bond.getIssuePrice()) r += 0.0001;
            else break;


        }


        return r;

    }


}


//<!---functions--->
```

Portfolio Class

```
import java.util.ArrayList;

public class Investor {

    //<!-- variables-->


    private String name;

    private double balance; //balance

    private Portfolio portfolio;

    private int numberOfBonds;


    //<!-- variables-->


    //<!---constructor--->

    public Investor(String name, int balance, double r) {
        this.name = name;
        this.balance = balance;
        portfolio = new Portfolio(r);
    }


    public Investor() {
    } //default constructor

    //<!---constructor--->


    //<!---getter--->


    public int getNumberOfBonds() {
        return numberOfBonds;
    }

    public String getName() {
        return name;
    }


    public double getBalance() {
        return balance;
    }


    //<!---getter--->
```

```
//<!---setter--->


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


public void setBalance(double balance) {
    this.balance = balance;
}


//<!---setter--->


//<!---functions--->

public void purchasing(Bond... bonds) {

    /*logic*/


    for (Bond bond : bonds) {

        numberOfBonds++;

        if(balance > bond.getIssuePrice()) {

            balance -= bond.getIssuePrice();

            portfolio.add(new InvestmentAnalysisSystem(this, bond));

            portfolio.print(numberOfBonds-1);



        }else{



            System.out.println("+--------------------------------------------------------------------------+ ");
            System.out.println("| "+bond.getName()+" cannot be purchased of your account balance is too low           | ");
            System.out.println("+--------------------------------------------------------------------------+ ");

        }


    }
}
```

```
    //new InvestmentAnalysisSystem(this, bond );

    public void analysePortfolio(){



    }


    //<!---setter--->

}
```

Main Class

```
public class Main {

    public static void main(String[] args) {



        Investor investor1 = new Investor("Jeongho", 400, 0.05 );
        Investor investor2 = new Investor("Andrew", 140, 0.05);
        Investor investor3 = new Investor("Christian", 150, 0.05);
        Investor investor4 = new Investor("Kevin", 160, 0.05);



        Bond bond1 =  new Bond("Bond1", "2017/11/11", "2022/11/11", 5, 100,1);
        Bond bond2 =  new Bond("Bond2", "2017/11/11", "2027/11/11", 4, 95,1);
        Bond bond3 =  new Bond("Bond3", "2017/11/11", "2037/11/11", 3, 92,1);
        Bond bond4 =  new Bond("Bond4", "2017/11/11", "2032/11/11", 2, 120,1);



        investor1.purchasing(bond1, bond2, bond3, bond4);

        investor1.analysePortfolio();


        investor2.purchasing(bond1);

        //investor2.readPortfolio();

        investor3.purchasing(bond1, bond2);

        //investor3.readPortfolio();





    }


}
```

```
import java.util.ArrayList;

public class Portfolio extends ArrayList<InvestmentAnalysisSystem> {


    //investor wants to check the actual value of a bond under different inflation rate

    double r;

    public Portfolio(double r) {

        this.r = r;


    }



    public void print(int i) {


            System.out.println("<!-----------------------" + this.get(i).getInvestorName() + " " + this.get(i).getBondName() + "
Analysis----------------------->\n");


            System.out.println(
                            "name: " + this.get(i).getInvestorName() + "\n" +
                            "current balance: " + this.get(i).getInvestorBalance() + "\n" +
                            "payout: " + this.get(i).payout() + "\n" +
                            "r: " + r + "\n" +
                            "value(" + r + "): " + this.get(i).value(r) + "\n" +
                            "macaulayDuration(" + r + "): " + this.get(i).macaulayDuration(r) + "\n"+
                            "irr: " + this.get(i).calculateIrr() + "\n"
            );

            System.out.println("# of bonds : "+this.get(i).getBondPurchasedNumber());


    }
}
```