Chapter 6, #7 – Determine the interest on a given balance at the end of a year.

```
package Homework2.bank;
import java.math.BigDecimal;
public class Account {
    protected long number;
    protected String name;
    protected BigDecimal balance;
    protected BigDecimal interestRate;
    // Constructor:
    public Account(long number, String name, BigDecimal balance, BigDecimal interestRate) {
        this.number = number;
        this.name = name;
        this.balance = balance;
        this.interestRate = interestRate;
    }
    // Two overloaded constructors:
    public Account(int number, BigDecimal balance, BigDecimal interestRate){
        this(number, "DirectDeposit", balance, interestRate);
    }
    public Account(int number, BigDecimal balance){
        this(number, "Checking", balance, BigDecimal.valueOf(2));
    }
    // Accessor methods:
    public long getNumber() { return number; }
    public String getName(){ return name; }
    public BigDecimal getBalance() { return balance; }
    public BigDecimal getInterestRate() { return interestRate; }
}
```

--- ---

```
package Homework2.bank;
import java.math.BigDecimal;
import java.math.RoundingMode;
import java.util.Calendar;
public class SavingsAccount extends Account {
    private int day;
    private int month;
    private int year;
    // Constructor:
    public SavingsAccount(long number, String name, BigDecimal balance, BigDecimal interestRate, int day, int
month, int year) {
        super(number, name, balance, interestRate);
        this.day = day;
        this.month = month;
        this.year = year;
    }
    // Two overloaded constructors:
    public SavingsAccount(String name, BigDecimal balance, BigDecimal interestRate, int day, int month, int year)
{
        this(2394874, name, balance, interestRate, day, month, year);
    }
    public SavingsAccount(int number, BigDecimal balance, BigDecimal interestRate, int day, int month, int year)
{
        this(number, "Savings", balance, interestRate, day, month, year);
    }
    // Accessor method:
    public int getYear() { return year; }
    // Three methods:
    public void deposit(BigDecimal amount) {
        balance = balance.add(amount);
    }
    public boolean withdraw(BigDecimal amount) {
        if (amount.compareTo(balance) > 0) { // amount > balance
            System.out.println("Insaficiant ballance");
            return false;
        } else {
            balance = balance.subtract(amount);
            return true;
        }
    }
    public BigDecimal computeInterest() {
        BigDecimal monthlyInterestNumerator = balance.multiply(interestRate).divide(BigDecimal.valueOf(100));
        BigDecimal monthlyInterest = monthlyInterestNumerator.divide(BigDecimal.valueOf(12), 2,
RoundingMode.HALF_UP);
        Calendar cal = Calendar.getInstance();
        cal.set(year, month - 1, day);
```

```
        Calendar endMonth = Calendar.getInstance();
        endMonth.set(year, month - 1, 1);
        endMonth.add(Calendar.MONTH, 1);
        endMonth.add(Calendar.DAY_OF_MONTH, -1);
        int daysOfTheMonth = endMonth.get(Calendar.DAY_OF_MONTH);
        int daysLeftTillTheEndOfTheMonth = daysOfTheMonth - day;
        BigDecimal monthsTillTheEndOfTheYear = BigDecimal.valueOf(12 -
month).multiply(monthlyInterest).setScale(2, RoundingMode.HALF_UP);;
        BigDecimal totalDaily = monthlyInterest.divide(BigDecimal.valueOf(daysOfTheMonth), 2,
RoundingMode.HALF_UP);
        BigDecimal daysTillTheEndOfTheMonth =
totalDaily.multiply(BigDecimal.valueOf(daysLeftTillTheEndOfTheMonth)).setScale(2, RoundingMode.HALF_UP);;
        // Determine the interest on a given balance at the end of a year:
        BigDecimal interestOnGivenBalance_EndOfYear = monthsTillTheEndOfTheYear.add(daysTillTheEndOfTheMonth);
        return interestOnGivenBalance_EndOfYear.setScale(2, RoundingMode.HALF_UP);
    }
}
```

--- ---

```
package Homework2.bank;
import java.math.BigDecimal;
public class Calculation {
    public static void main(String[] args) {
        SavingsAccount savings1 = new SavingsAccount(13742894, BigDecimal.valueOf(0), BigDecimal.valueOf(18), 5,
1, 2018);
        SavingsAccount savings2 = new SavingsAccount("Savings", BigDecimal.valueOf(200), BigDecimal.valueOf(2),
15, 11, 2020);
        savings1.deposit(BigDecimal.valueOf(100));
        savings2.withdraw(BigDecimal.valueOf(100));
        System.out.println("Interest on a given balance at the end of the " + savings1.getYear() + " year: " +
savings1.computeInterest());
        System.out.println("Interest on a given balance at the end of the " + savings2.getYear() + " year: " +
savings2.computeInterest());
    }
}
```