```java
package problem3;

import java.util.Scanner;

import problem3.employeeinfo.*;

public class Main {
    Employee[] emps = null;

    public static void main(String[] args) {
        new Main();
    }

    Main() {
        emps = new Employee[3];
        emps[0] = new Employee("Jim Daley", 2000, 9, 4);
        emps[1] = new Employee("Bob Reuben", 1998, 1, 5);
        emps[2] = new Employee("Susan Randolph", 1997, 2, 13);
        emps[0].createNewChecking(10500);
        emps[0].createNewSavings(1000);
        emps[0].createNewRetirement(9300);
        emps[1].createNewChecking(34000);
        emps[1].createNewSavings(27000);
        emps[2].createNewChecking(10038);
        emps[2].createNewSavings(12600);
        emps[2].createNewRetirement(9000);
        // for phase I – console output
        Scanner sc = new Scanner(System.in);
        System.out.print("See a report of all account balances? (y/n) ");
        String answer = sc.next();
        if (answer.equalsIgnoreCase("y"))
        {
            String info = getFormattedAccountInfo();
            System.out.println(info);
        } else {
            // do nothing..the application ends here
        }
    }

    private String getFormattedAccountInfo() {
         StringBuilder str = new StringBuilder("");
         for (Employee employee : emps)
         {
             str.append("\n");
             str.append(employee.getFormattedAcctInfo());

        }
        return str.toString();
    }
    }
```

```
package problem3.employeeinfo;

class Account {




        private final static double DEFAULT_BALANCE = 0.0;
        private double balance;
        private AccountType  acctType;

        private Employee employee;

        Account(Employee emp, AccountType acctType, double balance ) {
                employee = emp;
                this.acctType = acctType;
                this.balance = balance;

        }

        Account(Employee emp, AccountType acctType) {
                this(emp, acctType, DEFAULT_BALANCE);
        }

        public String toString()
        {
                String template  = "";
                if (this.acctType!=null)
                {
                        template  = String.format("Account type: %s",
this.acctType);
                }

                if (this.acctType!=null)
                {
                        template = template + String.format("\nCurrent bal: %s\n",
this.getBalance());
                }


                return template;
        }

        public void makeDeposit(double deposit) {
                // implement
                this.balance=this.balance+deposit;
        }

        public boolean makeWithdrawal(double amount) {
                if (this.balance <= amount)
                {
                        this.balance=this.balance-amount;
                        return true;

                }
        return false;
    }

    public static double getDefaultBalance() {
        return DEFAULT_BALANCE;
    }

    public double getBalance() {
        return balance;
    }

    public AccountType getAcctType() {
        return acctType;
    }



}

package problem3.employeeinfo;

public enum AccountType {
     CHECKING ,
     SAVINGS ,
     RETIREMENT ;

}

package problem3.employeeinfo;
import static java.util.Optional.ofNullable;

import java.time.LocalDate;
import java.util.Date;
import java.util.GregorianCalendar;

public class Employee {

    private Account savingsAcct;
    private Account checkingAcct;
    private Account retirementAcct;
    private String name;
    private LocalDate hireDate;



    public String getName() {
        return name;
    }


    public LocalDate getHireDate() {
        return hireDate;
    }
```

```
       public Employee(String name, int yearOfHire, int monthOfHire, int 
dayOfHire){
              this.name = name;




              hireDate = LocalDate.of(yearOfHire, monthOfHire, dayOfHire);

       }


       public void createNewChecking(double startAmount) {
              // implement
              if (this.checkingAcct == null ) {
                     this.checkingAcct=new Account(this, AccountType.CHECKING, 
startAmount);

              }



       }

       public void createNewSavings(double startAmount) {

                            if (this.savingsAcct == null ) {
                                   this.savingsAcct=new Account(this, 
AccountType.SAVINGS, startAmount);

                            }

       }

       public void createNewRetirement(double startAmount) {
              if (this.retirementAcct == null )
              {
                     this.retirementAcct=new Account(this, 
AccountType.RETIREMENT, startAmount);

              }

       }

       public String getFormattedAcctInfo()
       {
              String template  = "ACCOUNT INFO FOR %s\n\n%s%s%s";
              return String.format(template,this.getName(),
(this.checkingAcct==null?"":this.checkingAcct),(this.savingsAcct==null?"":this.s
avingsAcct) ,(this.retirementAcct==null?"":this.retirementAcct) );
       }
       public void deposit(AccountType acctType, double amt){
              switch (acctType)
              {
```

```
        case CHECKING:
            this.checkingAcct.makeDeposit(amt);

            break;

        case RETIREMENT:
            this.retirementAcct.makeDeposit(amt);

            break;
        case SAVINGS:
            this.savingsAcct.makeDeposit(amt);
            break;
        default:
            System.out.println("the type that your enter is not define 
on our database");
            break;
        }
    }
    public boolean withdraw(AccountType acctType, double amt){
        switch (acctType)
        {
        case CHECKING:
            return this.checkingAcct.makeWithdrawal(amt);



        case RETIREMENT:
            return this.retirementAcct.makeWithdrawal(amt);


        case SAVINGS:
            return this.savingsAcct.makeWithdrawal(amt);
        default:
            System.out.println("the type that your enter is not define 
on our database");
            break;
        }
        return false;
    }

}
```