

```

public class App {
    public static void main(String[] args) throws Exception {
        SavingsAccount savingsAccount = new SavingsAccount(1111, 5000, 0);
        savingsAccount.withdraw(10000);
        System.out.println(savingsAccount);

        System.out.println();

        CheckingAccount checkingAccount = new CheckingAccount(1111, 5000, 0, 5000);
        checkingAccount.withdraw(10001);
        System.out.println(checkingAccount);
    }
}

```

```

import java.util.Date;

```

```

class Account {
    private int id;
    private double balance;
    private double lastBalance;
    private double annualInterestRate;
    private Date dateCreated;
    private double totalWithdraw;
    private double totalDeposit;

    public Account() {
        this.id = 0;
        this.balance = 0;
        this.annualInterestRate = 0;
        this.dateCreated = new java.sql.Date(System.currentTimeMillis());
        this.totalWithdraw = 0;
        this.totalDeposit = 0;
        this.lastBalance = this.balance;
    }

    public Account(int id, double balance, double annualInterestRate) {
        this.id = id;
        this.balance = balance;
        this.annualInterestRate = annualInterestRate;
        this.dateCreated = new java.sql.Date(System.currentTimeMillis());
        this.totalWithdraw = 0;
        this.totalDeposit = 0;
        this.lastBalance = this.balance;
    }
}

```

```
}
```

```
// Setter
```

```
public void setId(int id) {  
    this.id = id;  
}
```

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

```
public void setAnnualInterestRate(double annualInterestRate) {  
    this.annualInterestRate = annualInterestRate;  
}
```

```
public void setDateCreated(Date dateCreated) {  
    this.dateCreated = dateCreated;  
}
```

```
public void setLastBalance(double lastBalance) {  
    this.lastBalance = lastBalance;  
}
```

```
// Getter
```

```
public Double getMonthlyInterestRate() {  
    return this.annualInterestRate / 12;  
}
```

```
public Double getMonthlyInterest() {  
    return ((this.annualInterestRate / 12) * this.lastBalance) / 100;  
}
```

```
public Double getLastBalance() {  
    return this.lastBalance + ((this.annualInterestRate / 100) * this.lastBalance);  
}
```

```
public Date getDateCreated() {  
    return this.dateCreated;  
}
```

```
public int getID() {
```

```

        return this.id;
    }

    public double getBalance() {
        return this.balance;
    }

    public double getTotalWithdraw() {
        return this.totalWithdraw;
    }

    public double getTotalDeposit() {
        return this.totalDeposit;
    }

    // Method

    public void withdraw(double withdraw) {
        this.totalWithdraw += withdraw;
        if (withdraw < 0) {
            System.out.println("INPUT ERROR");
        } else {
            this.lastBalance -= withdraw;
        }
    }

    public void deposit(double deposit) {
        this.totalDeposit += deposit;
        if (deposit < 0) {
            System.out.println("INPUT ERROR");
        } else {
            this.lastBalance += deposit;
        }
    }
}

class CheckingAccount extends Account {
    private double overdraftLimit;

    public CheckingAccount(int id, double balance, double annualInterestRate, double
    overdraftLimit) {
        super(id, balance, annualInterestRate);
    }
}

```

```

        this.overdraftLimit = overdraftLimit;
    }

    // getter
    public double getOverdraftLimit() {
        return overdraftLimit;
    }

    public String toString() {
        String transaction = new String("");
        if (getTotalWithdraw() > 0) {
            transaction += ("Withdraw: " + String.format("%.2f", getTotalWithdraw()) + "\n");
        }
        if (getTotalDeposit() > 0) {
            transaction += ("Deposit: " + String.format("%.2f", getTotalDeposit()) + "\n");
        }
        String overdraftError = new String("");
        if (getLastBalance() < overdraftLimit * (-1)) {
            setLastBalance(getBalance());
            overdraftError += "WITHDRAW ERROR!\n";
        }
        return "Checking Account\n" +
            "Overdraft Limit: " + String.format("%.2f", overdraftLimit) + "\n" +
            "Balance is " + String.format("%.2f", getBalance()) + "\n" +
            transaction +
            overdraftError +
            "Balance is " + String.format("%.2f", getLastBalance()) + "\n" +
            "This account was created at " + String.format("%tc", getDateCreated());
    }
}

```

```

class SavingsAccount extends Account {

    public SavingsAccount(int id, double balance, double annualInterestRate) {
        super(id, balance, annualInterestRate);
    }

    public String toString() {
        String transaction = new String("");
        if (getTotalWithdraw() > 0) {
            transaction += ("Withdraw: " + String.format("%.2f", getTotalWithdraw()) + "\n");
        }
        if (getTotalDeposit() > 0) {

```

```

        transaction += ("Deposit: " + String.format("%.2f", getTotalDeposit()) + "\n");
    }
    String aboutBalance = new String("");
    if (getLastBalance() < 0) {
        setLastBalance(getBalance());
        aboutBalance += "WITHDRAW ERROR!\n";
        transaction = new String("");
    } else {
        aboutBalance += "Balance is " + String.format("%.2f", getLastBalance()) + "\n";
    }
    return "Saving Account\n" +
        "Balance is " + String.format("%.2f", getBalance()) + "\n" +
        transaction +
        aboutBalance +
        "This account was created at " + String.format("%tc", getDateCreated());
    }
}

```