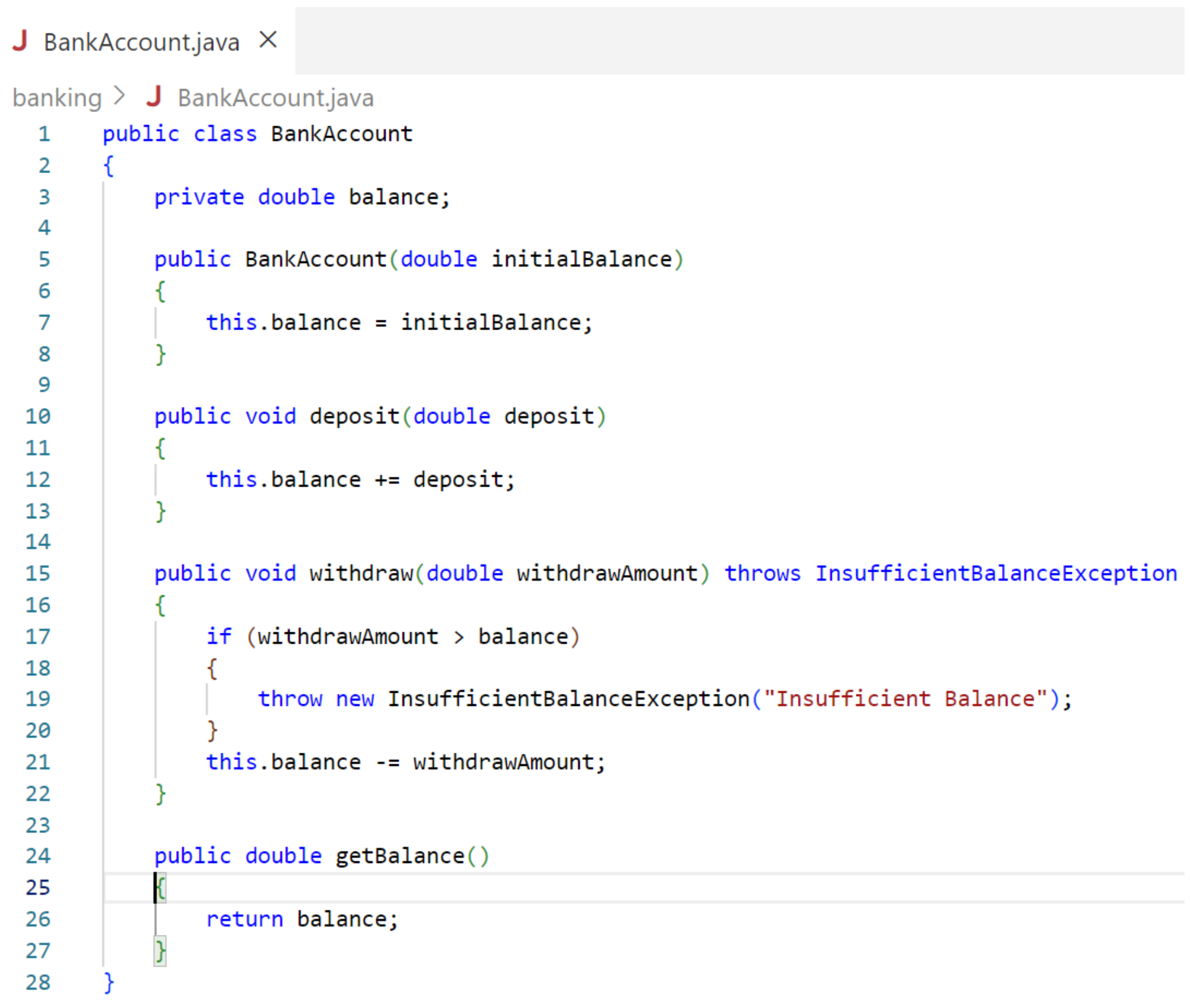
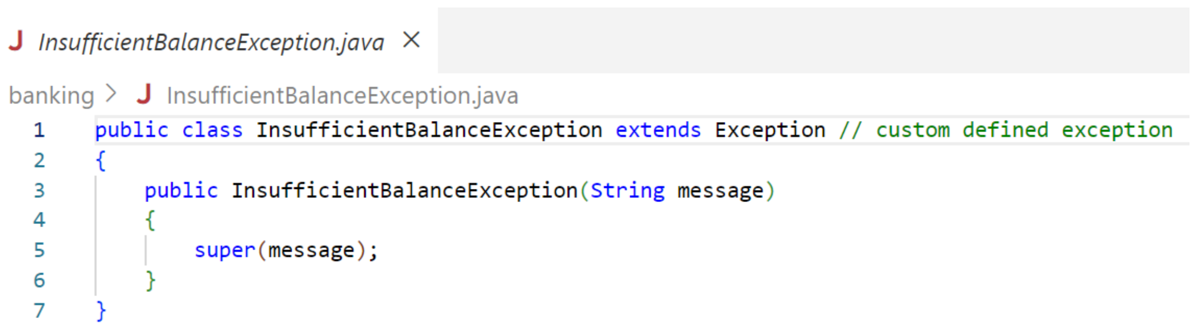
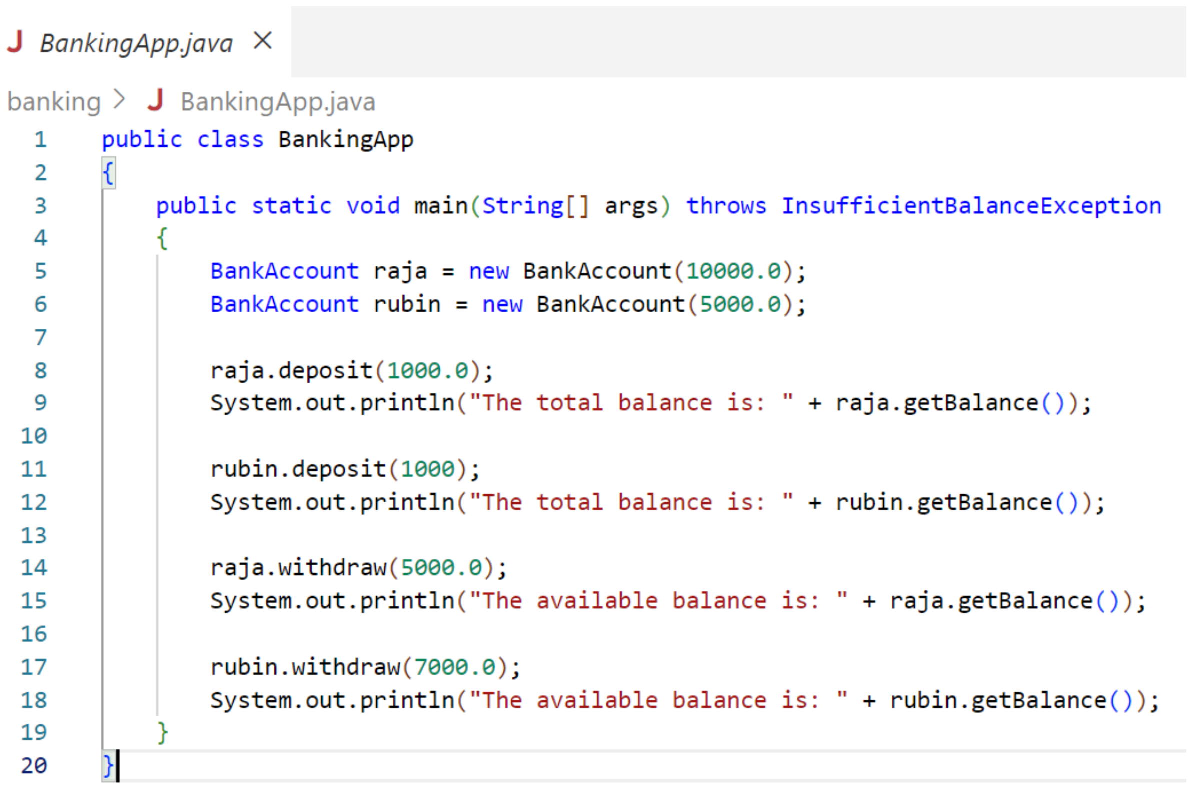


```java
public class BankAccount
{
    private double balance;

    public BankAccount(double initialBalance)
    {
        this.balance = initialBalance;
    }

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

    public void withdraw(double withdrawAmount) throws InsufficientBalanceException
    {
        if (withdrawAmount > balance)
        {
            throw new InsufficientBalanceException("Insufficient Balance");
        }
        this.balance -= withdrawAmount;
    }

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

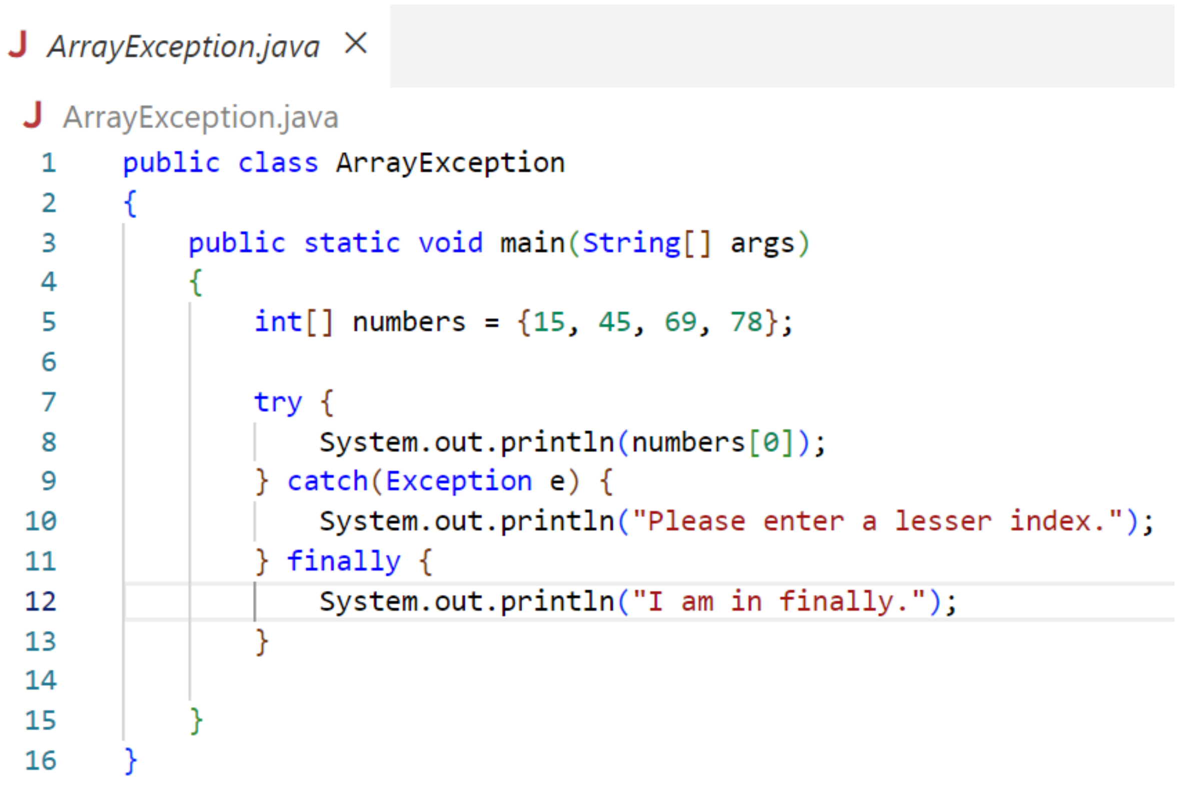

InsufficientBalanceException.java
banking > InsufficientBalanceException.java

```java
public class InsufficientBalanceException extends Exception // custom defined exception
{
    public InsufficientBalanceException(String message)
    {
        super(message);
    }
}
```


BankingApp.java

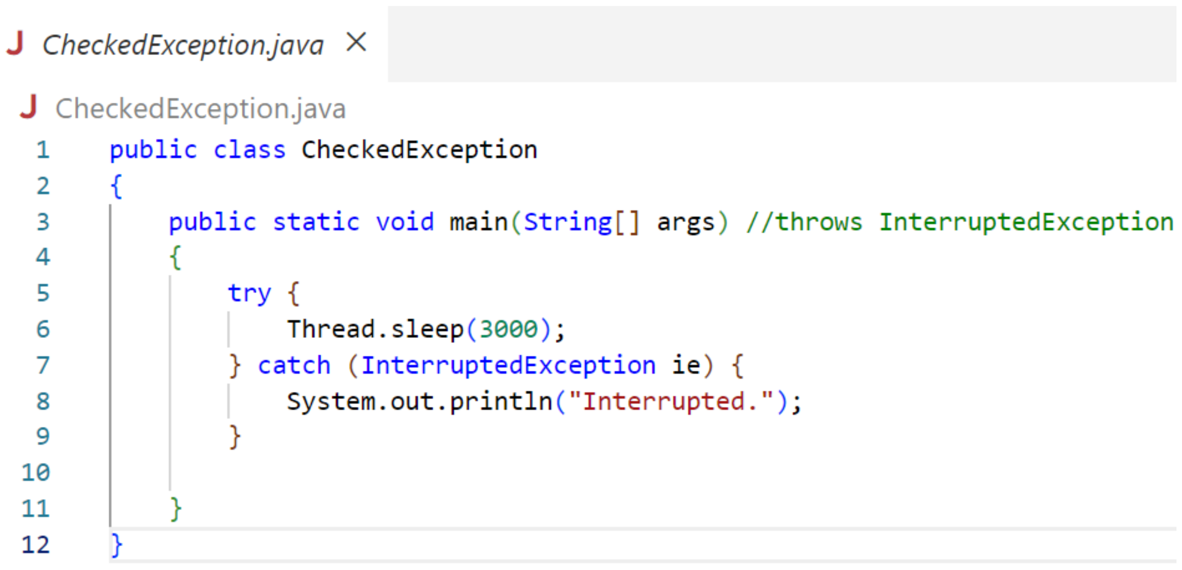
banking > BankingApp.java

```java
public class BankingApp
{
    public static void main(String[] args) throws InsufficientBalanceException
    {
        BankAccount raja = new BankAccount(10000.0);
        BankAccount rubin = new BankAccount(5000.0);

        raja.deposit(1000.0);
        System.out.println("The total balance is: " + raja.getBalance());

        rubin.deposit(1000);
        System.out.println("The total balance is: " + rubin.getBalance());

        raja.withdraw(5000.0);
        System.out.println("The available balance is: " + raja.getBalance());

        rubin.withdraw(7000.0);
        System.out.println("The available balance is: " + rubin.getBalance());
    }
}
```

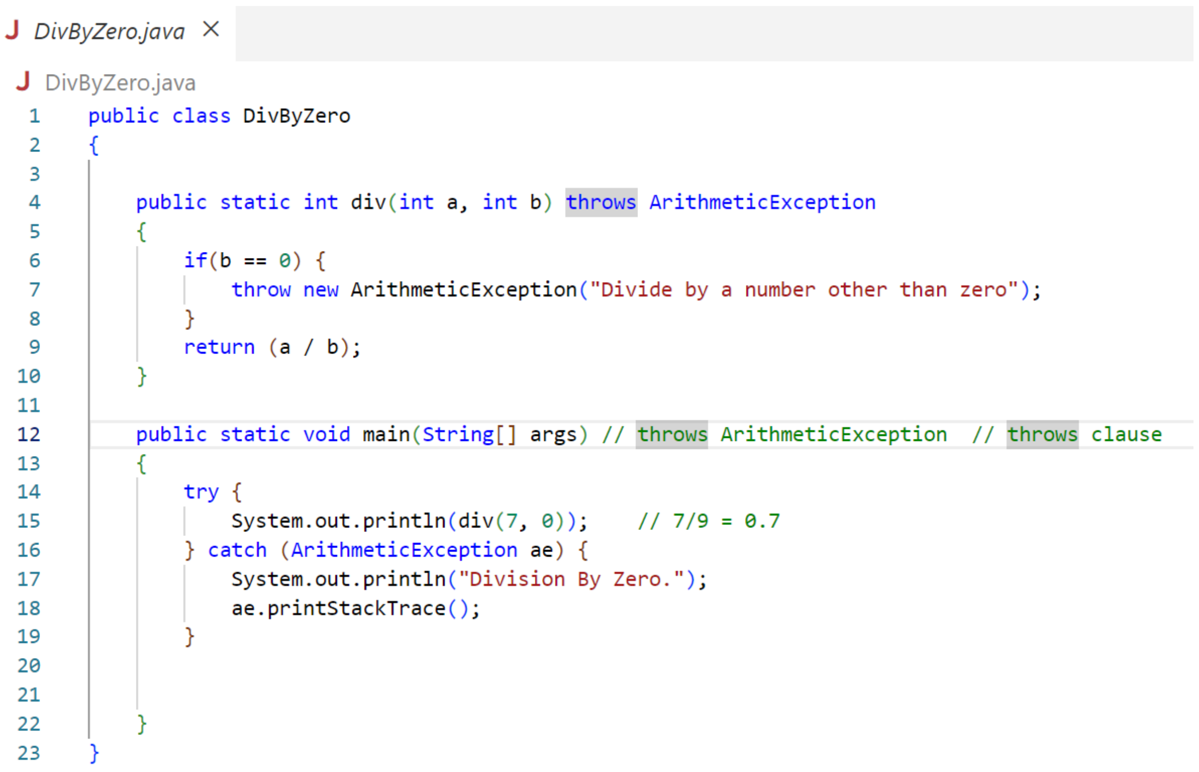

ArrayException.java

ArrayException.java

```java
public class ArrayException
{
    public static void main(String[] args)
    {
        int[] numbers = {15, 45, 69, 78};

        try {
            System.out.println(numbers[0]);
        } catch(Exception e) {
            System.out.println("Please enter a lesser index.");
        } finally {
            System.out.println("I am in finally.");
        }

    }
}
```


CheckedException.java

```java
public class CheckedException
{
    public static void main(String[] args) //throws InterruptedException
    {
        try {
            Thread.sleep(3000);
        } catch (InterruptedException ie) {
            System.out.println("Interrupted.");
        }

    }
}
```


DivByZero.java

```java
public class DivByZero
{

    public static int div(int a, int b) throws ArithmeticException
    {
        if(b == 0) {
            throw new ArithmeticException("Divide by a number other than zero");
        }
        return (a / b);
    }

    public static void main(String[] args) // throws ArithmeticException  // throws clause
    {
        try {
            System.out.println(div(7, 0));    // 7/9 = 0.7
        } catch (ArithmeticException ae) {
            System.out.println("Division By Zero.");
            ae.printStackTrace();
        }


    }
}
```