

Laboration 11, Arrayer

Avsikten med laborationen är att du ska träna på att använda arrayer.

Skapa paketet **laboration11** i ditt laborationsprojekt innan du fortsätter med laborationen.

Grundläggande uppgifter

Uppgift 11a

Din uppgift är att ersätta raden `// Komplettera med kod` med en eller flera rader med kod. Det handlar i samtliga deluppgifter om att använda arrayen *numbers* för att få ett speciellt körresultat.

```
package laboration11;

public class Exercisella {
    public void action() {
        double[] numbers = { 23.2, 14.7, 17.0, -5.9, -11.1, 26.3, 8.3, 7.6 };
//        double[] numbers = { -1, 2, 5, 8, 11, 14, 10, 6, 2, -4 };
        int counter;
        double sum;
        String res;

        // Komplettera med kod
    }

    public static void main( String[] args ) {
        Exercisella ella = new Exercisella();
        ella.action();
    }
}
```

När du testkör din lösning ska du använda båda varianterna av arrayen *numbers* (en av dem ska vara avmarkerad och den andre ska användas). Det översta körresultatet är när den överst arrayen används och det undre körresultatet då den undre arrayen används.

Exempel

Skriv ut alla tal som är mindre än 10 och antal utskrivna tal.

Körresultat 1: -5.9 -11.1 8.3 7.6 , antal = 4

Körresultat 2: -1.0 2.0 5.0 8.0 6.0 2.0 -4.0 , antal = 7

Lösning

```
// Ersätter raden "// Komplettera med kod"
counter = 0;
for( int i = 0; i < numbers.length; i++ ) {
    if( numbers[ i ] < 10 ) {
        System.out.print( numbers[ i ] + " " );
        counter++;
    }
}
System.out.println( ", antal = " + counter );
```

1. Skriv ut antalet element i arrayen

Körresultat 2: Antal element = 10

Körresultat 2: Talens summa är 53.0

Körresultat 2: 11.0 14.0 10.0

Körresultat 2: Antal tal större än 8: 3

Körresultat 2: -1.0 -4.0

Körresultat 2: Summan av de negativa talen är: -5.0

Körresultat 2: Talen baklänges: -4.0 2.0 6.0 10.0 14.0 11.0 8.0 5.0 2.0
-1.0

Körresultat 2: Vart 3:e tal baklänges: -4.0 10.0 8.0 -1.0

Körresultat 2: -1.0 2.0 5.0 8.0 11.0 14.0 10.0 6.0 2.0 -4.0

Körresultat 2: Antal tal större än 8: 3

Uppgift 11b

I nedanstående klass ska fyra metoder kompletteras med kod så att körresultatet längst ner erhålles. Metoderna som ska kompletteras motsvarar lösningarna på deluppgifterna 3, 5, 7 och 8 i Uppgift 11a.

Börja med att skapa klassen Exercise11b och kopiera sedan koden nedan.

```
package laboration11;

public class Exercisel1b {

    public void greaterThan8( double[] array ) {
        // Komplettera med kod
    }

    public void negativeNumbers( double[] array ) {
        // Komplettera med kod
    }

    public void reverse( double[] array ) {
        // Komplettera med kod
    }

    public void everyThirdReverse( double[] array ) {
        // Komplettera med kod
    }

    public void action() {
//        double[] numbers={ 23.2, 14.7, 17.0, -5.9, -11.1, 26.3, 8.3, 7.6 };
        double[] numbers = { -1, 2, 5, 8, 11, 14, 10, 6, 2, -4 };

        greaterThan8(numbers);
        negativeNumbers(numbers);
        reverse(numbers);
        everyThirdReverse(numbers);
    }

    public static void main( String[] args ) {
        Exercisel1b e11b = new Exercisel1b();
        e11b.action();
    }
}
```

Körresultat om den första arrayen används:

```
23.2  14.7  17.0  26.3  8.3
-5.9  -11.1
Talen baklänges: 7.6  8.3  26.3  -11.1  -5.9  17.0  14.7  23.2
Vart 3:e tal baklänges: 7.6  -11.1  14.7
```

Körresultat om den andra arrayen används:

```
11.0  14.0  10.0
-1.0  -4.0
Talen baklänges: -4.0  2.0  6.0  10.0  14.0  11.0  8.0  5.0  2.0  -1.0
Vart 3:e tal baklänges: -4.0  10.0  8.0  -1.0
```

Uppgift 11c

I nedanstående klass ska tre metoder kompletteras med kod så att körresultatet längst ner erhålles. Metoderna som ska kompletteras motsvarar lösningarna på deluppgifterna 2,4/10 och 6 i Uppgift 11a.

Börja med att skapa klassen Exercise11c och kopiera sedan koden nedan.

```
public class Exercise11c {

    public double sum( double[] array ) {
    }

    public int count8( double[] array ) {
    }

    public double sumNegative( double[] array ) {
    }

    public void action() {
        double[] numbers = { 23.2, 14.7, 17.0, -5.9, -11.1, 26.3, 8.3, 7.6 };
//        double[] numbers = { -1, 2, 5, 8, 11, 14, 10, 6, 2, -4 };
        double sum;

        System.out.println( "Talens summa är " + sum(numbers) );
        System.out.println( "Antal tal större än 8: " + count8(numbers) );
        sum = sumNegative(numbers);
        System.out.println( "Summan av de negativa talen är: " + sum );
    }

    public static void main( String[] args ) {
        Exercise11c e11c = new Exercise11c();
        e11c.action();
    }
}
```

Körresultat om den första arrayen används:

```
Talens summa är 80.1
Antal tal större än 8: 5
Summan av de negativa talen är: -17.0
```

Körresultat om den andra arrayen används:

```
Talens summa är 53.0
Antal tal större än 8: 3
Summan av de negativa talen är: -5.0
```

Uppgift 11d

Skriv metoden

```
public void reverse(int[] numbers)
```

vilken skriver ut talen i en array med heltal baklänges. Talen ska separeras med komma-tecken och ett mellanslag. Om du testkör metoden med koden:

```
Exerciselld elld = new Exerciselld();  
int[] arr1 = { 1, 2, 3 };  
int[] arr2 = { 1000, 100, 10, 1 };  
System.out.print( "Array 1 baklänges: " );  
elld.reverse( arr1 );  
System.out.println();  
System.out.print( "Array 2 baklänges: " );  
elld.reverse( arr2 );  
System.out.println();
```

så ska du få resultatet till höger.

| |
|-------------------------------------|
| Array 1 baklänges: 3, 2, 1 |
| Array 2 baklänges: 1, 10, 100, 1000 |

Uppgift 11e

Skriv metoden

```
public int member(int nbr, int[] array)
```

vilken kontrollerar om **nbr** finns i arrayen **array**. Om **nbr** finns i arrayen så returneras positionen för **nbr** i arrayen (första förekomsten om flera gånger) och annars returneras -1.

Testprogram

```
Exerciselle elle = new Exerciselle();  
int[] array = { 1, 2, 3, 9, 5, 9 };  
int index;  
index = elle.member( 7, array );  
System.out.println( "Position för 7: " + index );  
index = elle.member( 9, array );  
System.out.println( "Position för 9: " + index );
```

Körresultat

| |
|--------------------|
| Position för 7: -1 |
| Position för 9: 3 |

Uppgift 11f

Skriv ett program vilket

- deklarerar en array som rymmer 10 heltal
- låter användaren mata in tio heltal vilka lagras i arrayen. Inmatningen sker på vanligt sätt med dialogfönster och användaren matar in ett heltal varje gång dialogen visas.
- skriver ut heltalen i den ordning de matades in
- skriver vart annat tal framlänges
- skriver ut samtliga tal baklänges

Programmet ska innehålla metoderna:

- **printNumbers(int[] array)**
- **printEverySecond(int[] array)**

Dessutom bör din lösning använda metoden **reverse**, vilken du skrev i uppgift 11d.

Körresultat vid inmatning av talen: 12 5 8 2 3 3 3 11 221 7:

Följande tal matades in: 12, 5, 8, 2, 3, 3, 3, 11, 221, 7

Vartannat tal: 12, 8, 3, 3, 221

Talen baklänges: 7, 221, 11, 3, 3, 3, 2, 8, 5, 12

Fördjupande uppgifter

Uppgift 11g

Skriv metoden

```
public double[] reverse(double[] array)
```

vilken tar en array som input och som returnerar en array med samma element men med elementen i omvänd ordning.

Testprogram

```
Exercisellg ellg = new Exercisellg();  
double[] array = { 3.2, 4.7, -13.1, 14, 17.25 };  
double[] copy = ellg.reverse( array );  
for( int i = 0; i < array.length; i++ )  
    System.out.print( array[ i ] + " " );  
System.out.println();  
for( int i = 0; i < copy.length; i++ )  
    System.out.print(copy[ i ] + " " );  
System.out.println();
```

Utskrifter

```
3.2  4.7  -13.1  14.0  17.25
```

```
17.25  14.0  -13.1  4.7  3.2
```

Tänkbar algoritm för reverse-metoden

1. Skapa en array med samma kapacitet som input-fältet
2. Kopiera elementen från input-arrayen till den nya arrayen. Men gör det så att elementen byter plats, ungefär med tilldelningen (i är loop-variabel)

```
newArray[i] = array[array.length-i-1];
```

3. Returnera referensen till den nya arrayen.

Uppgift 11h

Skriv metoden

```
public double[] interval(double[] array, double min, double max)
```

vilken tar en array och ett intervall (min-max) som input och som returnerar en ny array som innehåller de element i input-arrayen som är i intervallet ($\text{min} \leq \text{element} \leq \text{max}$).

Tänkbar algoritm för intervall-metoden

1. Räkna antalet element i input-arrayen som är i intervallet
2. Skapa en ny array med korrekt kapacitet
3. Kopiera över elementen i intervallet, från input-arrayen, till den nya arrayen
4. Returnera referensen till den nya arrayen.

Skriv dessutom ett program som testar din lösning.

Uppgift 11i

Skriv metoden

```
public int[] odd(int[] array)
```

vilken ska returnera en array som innehåller alla udda element i en input-array.

Om input-arrayen är {4, 3, 79, 55, 56, 34, 11, 17} så ska retur-arrayen vara {3, 79, 55, 11, 17}

Extrauppgifter

Uppgift 11j

Skriv ett program som simulerar 100000 kast med tre tärningar och som beräknar antalet gånger olika antal prickar infaller. Resultatet av de 100000 kasten ska skrivas ut i tabellform.

Vid varje kast så ska tärningarnas siffersumma beräknas. Lägsta siffersumma är 3 (tre ettor) och högsta är 18 (tre sexor). Därefter ska en räknare öka med 1. Eftersom det behövs en räknare för varje siffersumma som kan inträffa så är det lämpligt med en array:

```
int[] counter = new int[19];
```

där vardera facket summerar antalet förekomster av en speciell siffersumma. Eftersom den minsta möjliga siffersumman är 3 och den största är 18 så kommer positionerna 0, 1 och 2 ej att användas.

Exempel:

Om tärningarna visar 8 prickar tillsammans så ska `counter[8]` ökas med ett.
Om tärningarna visar 18 prickar tillsammans så ska `counter[18]` ökas med ett.
Om du testar din metod bör den ge ett resultat liknande figuren till höger.

| Summa | Antal |
|-------|-------|
| 3 | 466 |
| 4 | 1378 |
| 5 | 2757 |
| 6 | 4626 |
| 7 | 7001 |
| 8 | 9574 |
| 9 | 11509 |
| 10 | 12542 |
| 11 | 12573 |
| 12 | 11692 |
| 13 | 9780 |
| 14 | 6875 |
| 15 | 4661 |
| 16 | 2800 |
| 17 | 1326 |
| 18 | 440 |

Uppgift 11k

Skriv metoden **arrangeSocks(int[] socks)**. Arrayen *socks* representerar nytvättade strumpor och varje typ av strumpa representeras av ett positivt heltal. Två strumpor med samma värde utgör ett par. Metoden ska para ihop de nytvättade strumporna (om möjligt) och skriva ut när par påträffas och när udda strumpor påträffas. Dessutom ska sammanfattande information skrivas ut.

Exempel

```
socks = {3,7,5,7,1,2,2,3,3,2,3};
```

Arrayen innehåller 11 strumpor varav 1 st av typen 1, 3 st av typen 2, 4 st av typen 3, 1 st av typen 5 och 2 st av typen 7.

Om du anropar metoden *arrangeSocks* med arrayen *socks* som parameter ska du få en utskrift liknande (ordningen kan vara annorlunda och positionerna för par kan skilja):

```
Pair (3), positions: 0, 7  
Pair (7), positions: 1, 3  
Single (5), position: 2  
Single (1), position: 4  
Pair (2), positions: 5, 6  
Pair (3), positions: 8, 10  
Single (2), position: 9
```

Antal par: 4

Antal udda strumpor: 3

Förslag till lösningar

Uppgift 11a

```
1
System.out.println( "Antal element = " + numbers.length );

2
sum = 0;
for( int i = 0; i < numbers.length; i++ ) {
    sum += numbers[ i ];
}
System.out.println( "Talens summa är " + sum );

3
for( int i = 0; i < numbers.length; i++ ) {
    if( numbers[ i ] > 8 ) {
        System.out.print( numbers[ i ] + " " );
    }
}
System.out.println();

4
counter = 0;
for( int i = 0; i < numbers.length; i++ ) {
    if( numbers[ i ] > 8 ) {
        counter++;
    }
}
System.out.println( "Antal tal större än 8: " + counter );

5
for( int i = 0; i < numbers.length; i++ ) {
    if( numbers[ i ] < 0 ) {
        System.out.print( numbers[ i ] + " " );
    }
}
System.out.println();

6
sum = 0;
for( int i = 0; i < numbers.length; i++ ) {
    if( numbers[ i ] < 0 ) {
        sum += numbers[ i ];
    }
}
System.out.println( "Summan av de negativa talen är: " + sum );

7
res = "";
for( int i = numbers.length - 1; i >= 0; i-- ) {
    res += numbers[ i ] + " ";
}
System.out.println( "Talen baklänges: " + res );

8
System.out.print( "Vart 3:e tal baklänges: " );
for( int i = numbers.length - 1; i >= 0; i-=3 ) {
    System.out.print( numbers[ i ] + " " );
}
System.out.println();

9
System.out.print( "Samtliga tal: " );
for( double elem : numbers ) {
    System.out.print( elem + " " );
}
System.out.println();

10
for( double elem : numbers ) {
    if( elem > 8 ) {
        antal++;
    }
}
System.out.println( "Antal tal större än 8: " + antal );
```

Uppgift 11b

```
package laboration11;

public class Exercisellb {

    public void greaterThan8( double[] array ) {
        String res = "";
        for( int i = 0; i < array.length; i++ ) {
            if( array[ i ] > 8 ) {
                System.out.print( array[ i ] + " " );
            }
        }
        System.out.println( res );
    }

    public void negativeNumbers( double[] array ) {
        for( int i = 0; i < array.length; i++ ) {
            if( array[ i ] < 0 ) {
                System.out.print( array[ i ] + " " );
            }
        }
        System.out.println();
    }

    public void reverse( double[] array ) {
        String res = "";
        for( int i = array.length - 1; i >= 0; i-- ) {
            res += array[ i ] + " ";
        }
        System.out.println( "Talen baklänges: " + res );
    }

    public void everyThirdReverse( double[] array ) {
        System.out.print( "Vart 3:e tal baklänges: " );
        for( int i = array.length - 1; i >= 0; i-=3 ) {
            System.out.print( array[ i ] + " " );
        }
        System.out.println();
    }

    public void action() {
//        double[] numbers = { 23.2, 14.7, 17.0, -5.9, -11.1, 26.3, 8.3, 7.6 };
        double[] numbers = { -1, 2, 5, 8, 11, 14, 10, 6, 2, -4 };

        greaterThan( numbers );
        negativeNumbers( numbers );
        reverse( numbers );
        everyThirdReverse( numbers );
    }

    public static void main( String[] args ) {
        Exercisellb ellb = new Exercisellb();
        ellb.action();
    }
}
```

Uppgift 11c

```
package laboration11;

public class Exercisel1c {

    public double sum( double[] array ) {
        double sum = 0;
        for( int i = 0; i < array.length; i++ ) {
            sum += array[ i ];
        }
        return sum;
    }

    public int count8( double[] array ) {
        int counter = 0;
        for( int i = 0; i < array.length; i++ ) {
            if( array[ i ] > 8 ) {
                counter++;
            }
        }
        return counter;
    }

    public double sumNegative( double[] array ) {
        double sum = 0;
        for( int i = 0; i < array.length; i++ ) {
            if( array[ i ] < 0 ) {
                sum += array[ i ];
            }
        }
        return sum;
    }

    public void action() {
//        double[] numbers = { 23.2, 14.7, 17.0, -5.9, -11.1, 26.3, 8.3, 7.6 };
        double[] numbers = { -1, 2, 5, 8, 11, 14, 10, 6, 2, -4 };
        double sum;

        System.out.println( "Talens summa är " + sum( numbers ) );
        System.out.println( "Antal tal större än 8: " + count8(numbers) );
        sum = sumNegative ( numbers );
        System.out.println( "Summan av de negativa talen är: " + sum );
    }

    public static void main( String[] args ) {
        Exercisel1c el1c = new Exercisel1c();
        el1c.action();
    }
}
```

Uppgift 11d

```
public void reverse( int[] numbers ) {
    for( int i = numbers.length-1 ; i >= 0; i-- ) {
        System.out.print( numbers[ i ] );
        if( i > 0 ) {
            System.out.print( ", " );
        }
    }
}
```

Uppgift 11e

```
public int member( int nbr, int[] array ) {
    for( int i = 0; i < array.length; i++ ) {
        if( nbr == array[ i ] ) {
            return i;
        }
    }
    return -1;
}
```

Uppgift 11f

```
public class Exercisellf {
    public void input( int[] array ) {
        for( int i = 0; i < array.length; i++ ) {
            array[ i ] = Integer.parseInt( JOptionPane.showInputDialog( "Ange tal " +
( i + 1 ) ) );
        }
    }

    public void printNumbers( int[] array ) {
        System.out.print( "Följande tal matades in: " );
        for( int i = 0; i < array.length; i++ ) {
            System.out.print( array[ i ] );
            if( i < array.length-1 )
                System.out.print( ", " );
        }
        System.out.println();
    }

    public void printEverySecond( int[] array ) {
        System.out.print( "Vartannat tal: " );
        for( int i = 0; i < array.length; i += 2 ) {
            System.out.print( array[ i ] );
            if( i < array.length-2 )
                System.out.print( ", " );
        }
        System.out.println();
    }

    public void program() {
        Exerciselld elld = new Exerciselld();
        int[] numbers = new int[ 10 ];
        input( numbers );
        printNumbers( numbers );
        printEverySecond( numbers );
        System.out.print( "Talen baklänges: " );
        elld.reverse( numbers );
        System.out.println();
    }

    public static void main(String[] args) {
        Exercisellf uppl1f = new Exercisellf();
        uppl1f.program();
    }
}
```

Uppgift 11g

```
public double[] reverse( double[] array ) {
    double[] newArray = new double[ array.length ];
    for( int i = 0; i < array.length; i++ ) {
        newArray[ i ] = array[ array.length-i-1 ];
    }
    return newArray;
}
```

Uppgift 11h

```
public double[] interval( double[] array, double min, double max ) {
    int counter = 0, index=0;
    double[] res;
    for( int i = 0; i < array.length; i++ ) {
        if( ( array[ i ] >= min ) && ( array[ i ] <= max ) )
            counter++;
    }
    res = new double[ counter ];
    for( int i = 0; i < array.length; i++ ) {
        if( ( array[ i ] >= min ) && ( array[ i ] <= max ) ) {
            res[ index ] = array[ i ];
            index++;
        }
    }
    return res;
}
```

Uppgift 11i

```
public int[] odd( int[] array ) {
    int counter=0, index=0;
    int[] newArray;
    for( int i = 0; i < array.length; i++ ) {
        if( array[ i ] % 2 == 1 )
            counter++;
    }
    newArray = new int[ counter ];
    for( int i = 0; i < array.length; i++ ) {
        if( array[ i ] % 2 == 1 ) {
            newArray[ index ] = array[ i ];
            index++;
        }
    }
    return newArray;
}
```

Uppgift 11j

```
package laboration11;
import java.util.*;

public class Exercisellj {
    private Random rand = new Random();

    public int roll() {
        return rand.nextInt(6) + 1;
    }

    public void rollDices( int[] result, int nbrOfRolls ) {
        int dice1, dice2, dice3;
        for( int i = 0; i < nbrOfRolls; i++ ) {
            dice1 = roll();
            dice2 = roll();
            dice3 = roll();
            result[ dice1 + dice2 + dice3 ]++;
        }
    }

    public void printResult( int[] result ) {
        System.out.println( "Summa    Antal" );
        for( int i = 3; i < result.length; i++ ) {
            System.out.printf( "%5d%5d", i, result[ i ] );
        }
    }

    public void program() {
        int[] counter = new int[e];
        rollDices( counter, 100000 );
        printResult( counter );
    }

    public static void main( String[] args ) {
        Exercisellj ellj = new Exercisellj ();
        ellj.program();
    }
}
```

Uppgift 11k

```
public void arrangeSocks(int[] socks) {
    int i = 0, j, id, singles=0, pairs=0;
    if (socks != null) { //Det är möjligt att anropa metoden med argumentet null
        while (i < socks.length) {
            id = socks[i];
            if (id != -1) {
                j = i + 1;
                while (j < socks.length && socks[j] != id) {
                    j++;
                }
                if (j < socks.length) {
                    System.out.println("Pair (" + id + "), positions: " + i + ", " + j);
                    socks[j] = -1;
                    pairs++;
                } else {
                    System.out.println("Single (" + id + "), position: " + i);
                    singles++;
                }
            }
            i++;
        }
    }
    System.out.println("\nAntal par: " + pairs + "\nAntal udda strumpor: " + singles);
}
```