

Laboration 19, Arrayer

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

Skapa paketet **laboration19** i ditt laborationsprojekt innan du fortsätter med laborationen.

Grundläggande uppgifter

Uppgift 19a

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 laboration19;

public class Exercise19a {
    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 ) {
        Exercise19a e19a = new Exercise19a();
        E19a.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 );
```

Deluppgifter

1. Skriv ut antalet element i arrayen
Körresultat 1: Antal element = 8
Körresultat 2: Antal element = 10
2. Skriv ut talens summa
Körresultat 1: Talens summa är 80.1
Körresultat 2: Talens summa är 53.0
3. Skriv ut alla tal som är större än 8
Körresultat 1: 23.2 14.7 17.0 26.3 8.3
Körresultat 2: 11.0 14.0 10.0
4. Skriv ut antalet tal som är större än 8
Körresultat 1: Antal tal större än 8: 5
Körresultat 2: Antal tal större än 8: 3
5. Skriv ut alla negativa tal (tal mindre än noll)
Körresultat 1: -5.9 -11.1
Körresultat 2: -1.0 -4.0
6. Skriv ut summan av de negativa talen
Körresultat 1: Summan av de negativa talen är: -17.0
Körresultat 2: Summan av de negativa talen är: -5.0
7. Skriv ut talen i arrayen med start från slutet (dvs. baklänges)
Körresultat 1: Talen baklänges: 7.6 8.3 26.3 -11.1 -5.9 17.0 14.7 23.2
Körresultat 2: Talen baklänges: -4.0 2.0 6.0 10.0 14.0 11.0 8.0 5.0 2.0
-1.0
8. Skriv ut vart 3:e tal i arrayen med start från slutet
Körresultat 1: Vart 3:e tal baklänges: 7.6 -11.1 14.7
Körresultat 2: Vart 3:e tal baklänges: -4.0 10.0 8.0 -1.0
9. Skriv ut samtliga tal i arrayen med start från början. Använd en förenklad for-loop.
Körresultat 1: 23.2 14.7 17.0 -5.9 -11.1 26.3 8.3 7.6
Körresultat 2: -1.0 2.0 5.0 8.0 11.0 14.0 10.0 6.0 2.0 -4.0
10. Skriv ut antalet tal som är större än 8. Använd en förenklad for-loop i din lösning.
Körresultat 1: Antal tal större än 8: 5
Körresultat 2: Antal tal större än 8: 3

Uppgift 19b

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 19a.

Börja med att skapa klassen Exercise19b och kopiera sedan koden nedan.

```
package laboration19;

public class Exercise19b {

    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 ) {
        Exercise19b e19b = new Exercise19b();
        e19b.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 19c

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 19a.

Börja med att skapa klassen Exercise19c och kopiera sedan koden nedan.

```
public class Exercise19c {  
  
    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 ) {  
        Exercise19c e19c = new Exercise19c();  
        e19c.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
```

Fördjupande uppgifter

Uppgift 19d

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:

```
Exercise19d e19d = new Exercise19d();  
int[] arr1 = { 1, 2, 3 };  
int[] arr2 = { 1000, 100, 10, 1 };  
System.out.print( "Array 1 baklänges: " );  
e19d.reverse( arr1 );  
System.out.println();  
System.out.print( "Array 2 baklänges: " );  
e19d.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 19e

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

```
Exercise19e e19e = new Exercise19e();  
int[] array = { 1, 2, 3, 9, 5, 9 };  
int index;  
index = e19e.member( 7, array );  
System.out.println( "Position för 7: " + index );  
index = e19e.member( 9, array );  
System.out.println( "Position för 9: " + index );
```

Körresultat

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

Uppgift 19f

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 19d.

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

Uppgift 19g

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

```
Exercise19g e19g = new Exercise19g();
double[] array = { 3.2, 4.7, -13.1, 14, 17.25 };
double[] copy = e19g.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 19h

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 19i

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 19j

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.

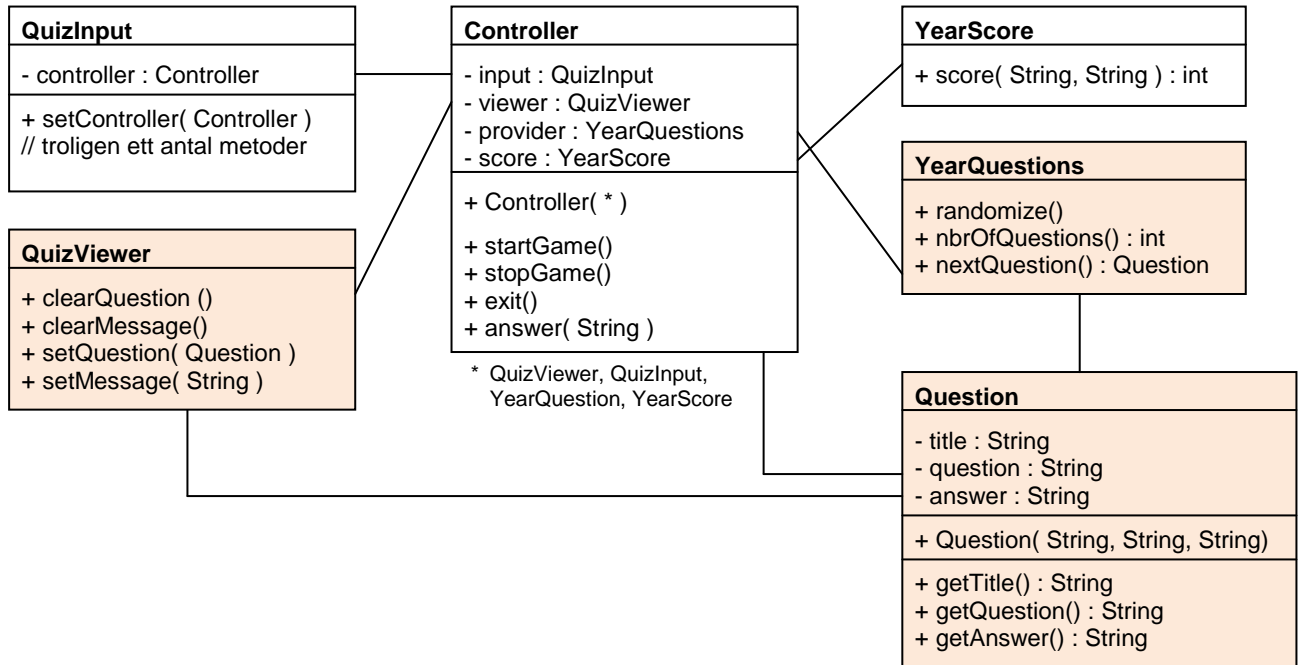
Om tärningarna visar 18 prickar tillsammans så ska `counter[18]` ökas med ett.

Om du testar din metod bör den ge ett resultat liknande figuren till höger.

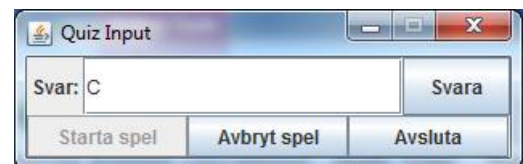
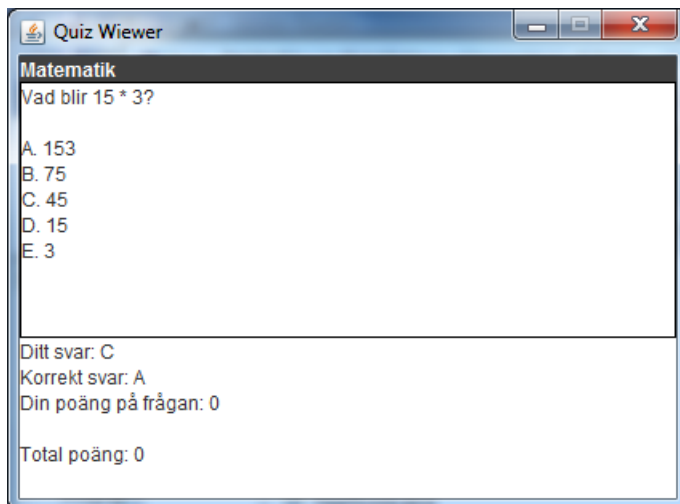
| Summa | Antal |
|-------|-------|
| 3 | 466 |
| 4 | 1378 |
| 5 | 2757 |
| 6 | 4626 |
| 7 | 7001 |
| 8 | 9574 |
| 9 | 11509 |
| 10 | 12542 |
| 11 | 12573 |
| 12 | 11692 |
| 13 | 9780 |
| 14 | 6875 |
| 15 | 4661 |
| 16 | 2800 |
| 17 | 1326 |
| 18 | 440 |

Uppgift 19k

Denna uppgift förutsätter att du redovisat och fått godkänt på programmeringsuppgift 3. Den förutsätter också att du följt klassdiagrammet i uppgiften:



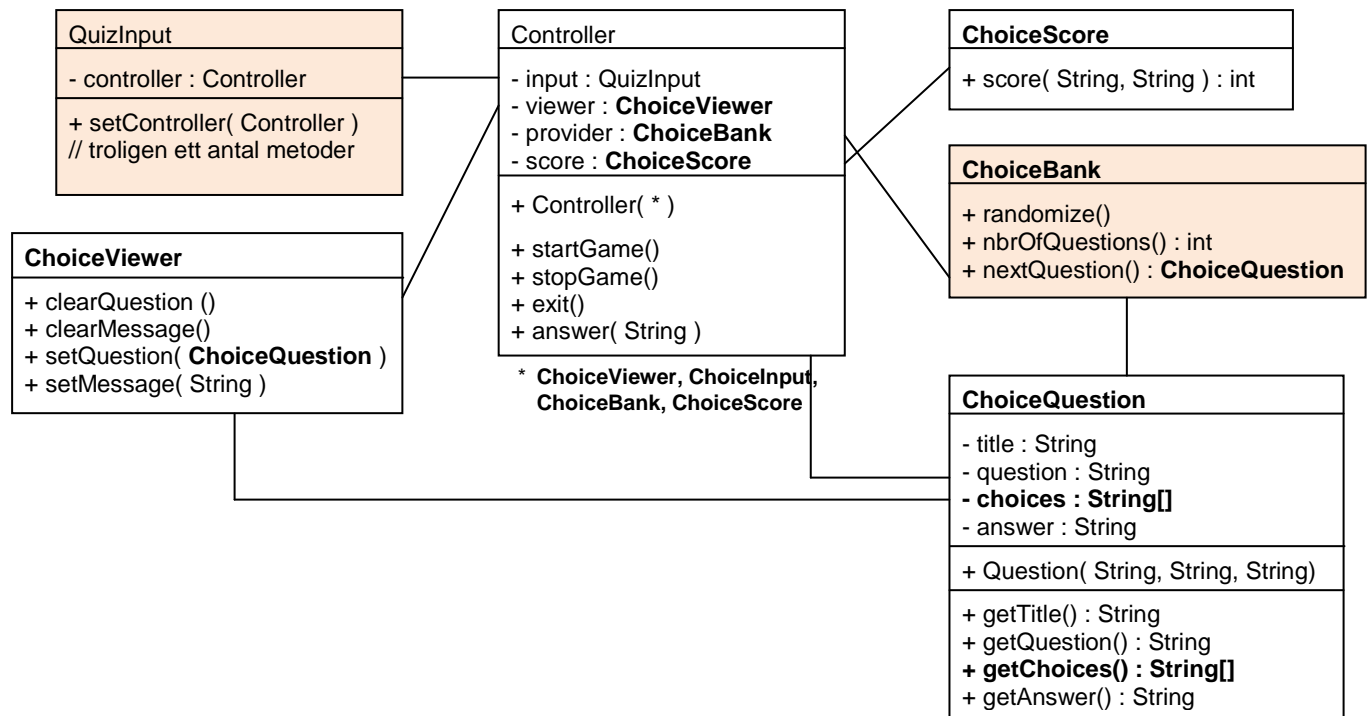
En ny typ av fråga ska användas i systemet, sk alternativfråga. I en sådan fråga ges användaren ett antal alternativ av vilket ett är korrekt.



Förändringen innebär att de flesta klasserna i systemet ersätts med nya klasser, speciellt anpassade till den nya frågetypen:

- *QuizViewer* ska ersättas med **ChoiceViewer**
- *Question* ska ersättas med **ChoiceQuestion**
- *YearScore* ska ersättas med **ChoiceScore**
- *YearQuestions* ska ersättas med **ChoiceBank** (given)

Dessutom ska *Controller* modifieras lite grand.



Om du jämför klassdiagrammen ser du att det är ganska små förändringar som krävs. Det är endast i *ChoiceQuestion* som en instansvariabel och en get-metod tillkommit. I övrigt har referensers typ förändrats, framför allt i *Controller*-klassen.

Förslag till arbetsordning

- Kopiera filerna i *quiz*-paketet till paketet *labquiz* (gör det i Eclipse så ändras paket-instruktionen automatiskt).
- Ett bekvämt sätt att få alla referenser ändrade på är att använda *Refactor – Rename*.
Exempel:
Högerklicka *Question.java* i Eclipse och välj Copy.
Högerklicka paketet *laboration19* och välj Paste.
Kopian får default namnet *CopyOfQuestion* vilket duger bra.
Högerklicka *Question.java* på nytt och välj *Refactor – Rename...* och ange namnet *ChoiceQuestion*.
Nu är referenserna i övriga klasser ändrade och du kan göra nödvändiga modifieringar i *ChoiceQuestion*.
- Ändra sedan namnet på *YearQuestions* till *ChoiceBank* (så referenserna i systemet ändras) och ersätt sedan *ChoiceBank.java* i eclipse med versionen som finns i DA211TL19HT12.zip.
- Kvar är klasserna *ChoiceViewer* och *ChoiceScore*. Metoden *ChoiceScore-score* ska fungera så här:
Är korrekt alternativ valt så utdela 5 poäng, annars 0 poäng.

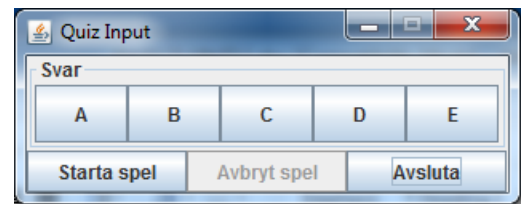
Uppgift 19I

Denna uppgift förutsätter att du löst Laboration19 - Uppgift 19k.

Ett problem med inmatningen av svar är att användaren kan lockas att skriva in själva svaret eller ge flera alternativ eller... Genom att ersätta klassen *QuizInput* med klassen *ChoiceInput* i systemet så blir svaret från användaren kontrollerat. Klicket på knapp ger svaret.

I din lösning kan du utgå från att det är 5 alternativ på varje fråga. På kommande laborationer kommer vi troligen göra systemet flexiblare.

Panelen med svarsknapparna har en `TitledBorder` i figuren ovan.



| ChoiceInput |
|--|
| - controller : Controller |
| + setController(Controller) // övriga metoder |

Förslag till lösningar

Uppgift 19a

```
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 19b

```
package laboration19;

public class Exercisel9b {

    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 ) {
        Exercisel9b e19b = new Exercisel9b();
        e19b.action();
    }
}
```

Uppgift 19c

```
package laboration19;

public class Exercise19c {

    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 ) {
        Exercise19c e19c = new Exercise19c();
        e19c.action();
    }
}
```

Uppgift 19d

```
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 19e

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

Uppgift 19f

```
public class Exercisel9f {
    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() {
        Exercisel9d e19d = new Exercisel9d();
        int[] numbers = new int[ 10 ];
        input( numbers );
        printNumbers( numbers );
        printEverySecond( numbers );
        System.out.print( "Talen baklänges: " );
        e19d.reverse( numbers );
        System.out.println();
    }

    public static void main(String[] args) {
        Exercisel9f uppl9f = new Exercisel9f();
        uppl9f.program();
    }
}
```

Uppgift 19g

```
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 19h

```
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 19i

```
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 19j

```
package laboration19;
import java.util.*;

public class Exercise19j {
    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[ 19 ];
        rollDices( counter, 100000 );
        printResult( counter );
    }

    public static void main( String[] args ) {
        Exercise19j e19j = new Exercise19j ();
        e19j.program();
    }
}
```

Uppgift 19k

```
public class ChoiceQuestion {
    :
    private String[] choices;

    public ChoiceQuestion(String title, String question, String[] choices, String
answer) {
        this.title = title;
        this.question = question;
        this.choices = choices;
        this.answer = answer;
    }

    public String getTitle() {...}

    public String getQuestion(){...}

    public String[] getChoices() {
        return choices;
    }

    public String getAnswer(){...}
}
```



```
public class ChoiceViewer extends JPanel {
    :
    private String labels = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

    public ChoiceViewer() {...}

    public void clearQuestionArea(){...}

    public void clearMessageArea(){...}

    public void setQuestion(ChoiceQuestion question) {
        String res = question.getQuestion() + "\n";
        lblTitle.setText(question.getTitle());
        String[] choices = question.getChoices();
        for(int i=0; choices!=null && i<choices.length; i++) { // Ä
            res += "\n" + labels.charAt(i) + ". " + choices[i];
        }
        taQuestion.setText(res);
    }

    public void setMessage(String message) {...}
}

public class ChoiceScore {
    public int score(String solution, String player) {
        if( solution.equals(player))
            return 5;
        return 0;
    }
}
```

Uppgift 191

```
public class ChoiceInput extends JPanel {
    // div instansvariabler
    private JButton btnA = new JButton("A");
    private JButton btnB = new JButton("B");
    private JButton btnC = new JButton("C");
    private JButton btnD = new JButton("D");
    private JButton btnE = new JButton("E");

    public ChoiceInput() {
        JPanel panelCenter = new JPanel(new GridLayout(1,5));
        JPanel panelSouth = new JPanel(new GridLayout(1, 3));
        setLayout(new BorderLayout());
        panelCenter.setBorder(BorderFactory.createTitledBorder("Svar"));
        // design + add i paneler
        addListeners();
    }

    private void addListeners() {
        ButtonListener bl = new ButtonListener();
        AnswerListener al = new AnswerListener();
        btnStartGame.addActionListener(bl);
        btnStopGame.addActionListener(bl);
        btnExit.addActionListener(bl);
        btnA.addActionListener(al);
        btnB.addActionListener(al);
        btnC.addActionListener(al);
        btnD.addActionListener(al);
        btnE.addActionListener(al);
    }

    public void setController(Controller controller) {...}

    public void enableAnswer(boolean enabled) {
        btnA.setEnabled(enabled);
        btnB.setEnabled(enabled);
        btnC.setEnabled(enabled);
    }
}
```

```
        btnD.setEnabled(enabled);  
        btnE.setEnabled(enabled);  
    }  
  
    public void enableStart(boolean enabled) {...}  
  
    public void enableStop(boolean enabled) {...}  
  
    private class ButtonListener implements ActionListener {  
        public void actionPerformed(ActionEvent e) {  
            // händelsehantering för Starta spel, Avbryt spel och Avsluta  
        }  
    }  
  
    private class AnswerListener implements ActionListener {  
        public void actionPerformed(ActionEvent e) {  
            JButton btn = (JButton) e.getSource();  
            controller.answer(btn.getText());  
        }  
    }  
}
```