Objektorienterad Programmering (TDDC77)

Föreläsning V: arrayer, metoder, räckvidd (scope), eclipse

Ahmed Rezine

IDA, Linköpings Universitet

Höstermin 2016

Outline

Arrayer

Metoder

Räckvidd (Scope)

Eclipse

Vända om inlästa värden

```java
/* Reverse.java
 * Programmet demonstrera hur man kan läsa
 * in verden och skriva ut de i omvänd ordning
 */
import java.util.Scanner;
class Reverse{
    /* Be användaren om två tal och skriver ut de
     * i omvänd ordning
     */
    public static void main(String[] args){
     Scanner in = new Scanner (System.in);
     int x0, x1, x2, x3;

     System.out.println ("Mata in ett heltal: ");
     x0 = in.nextInt();
     System.out.println ("Mata in ett heltal: ");
     x1 = in.nextInt();
     System.out.println ("Mata in ett heltal: ");
     x2 = in.nextInt();
     System.out.println ("Mata in ett heltal: ");
     x3 = in.nextInt();
     System.out.print("Du matade in:");
     System.out.print(" "+ x3);
     System.out.print(" "+ x2);
     System.out.print(" "+ x1);
     System.out.print(" "+ x0);
     System.out.println();
    }
}
```

Vända om inlästa värden

- Vi kan lätt göra samma sak för fyra variabler ...
- Hur gör vi för 400 inlästa värden ?
- Ofta behöver man flera likadana variabler
- Kallas "Array" på engleska, ibland "Fält" på svenska
- En jätte bra tutorial `http://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html`

```java
int[] numbers = new int[4];
// ... eller ...
int[] numbers;
numbers = new int[4];
// ... eller ...
int[] numbers = {3,5,8,13};

System.out.println("Första elementet är: " + numbers[0]);
System.out.println("Sista elementet är: " + numbers[numbers.Length - 1]);

float[] ratios = new float[12];
String[]  names = new String[MAX];
boolean[] flags = new boolean[numberOfFlags];
String[] names = {"TDDC77", "TDDD39", "TDDC75"};
```

Att vända om 400 inlästa värden

```
int nrOfInts = 400;
int[] värden = new int[nrOfInts];
int pos = 0;

while(pos <nrOfInts){
    System.out.println("Mata in ett heltal: ");
    värden[pos] = in.nextInt();
    pos = pos + 1;
}

pos = 399;

while(pos >= 0){
    System.out.println(värden[pos] + " ");
    pos = pos - 1;
}
```

Att vända om 400 inlästa värden med For slingan

```
int nrOfInts = 400;
int[] vaerden = new int[nrOfInts];


for(int pos=0; pos<nrOfInts; pos++){
    System.out.println("Mata in ett heltal: ");
    vaerden[pos] = in.nextInt();
}

for(int pos=nrOfInts-1; pos>=0; pos--){
    System.out.println(vaerden[pos] + " ");
}
```

Outline

Arrayer

Metoder

Räckvidd (Scope)

Eclipse

Metoder

- En bit kod med ett eget namn
- Kan anropas flera gånger från olika delar av programmet
- När de körts färdigt fortsätter programmet där man var innan

Method Summary

Methods

Modifier and Type	Method and Description
char	`charAt(int index)` Returns the `char` value at the specified index.
int	`codePointAt(int index)` Returns the character (Unicode code point) at the specified index.
int	`codePointBefore(int index)` Returns the character (Unicode code point) before the specified index.
int	`codePointCount(int beginIndex, int endIndex)` Returns the number of Unicode code points in the specified text range of this `String`.
int	`compareTo(String anotherString)` Compares two strings lexicographically.
int	`compareToIgnoreCase(String str)` Compares two strings lexicographically, ignoring case differences.
`String`	`concat(String str)` Concatenates the specified string to the end of this string.
boolean	`contains(CharSequence s)` Returns true if and only if this string contains the specified sequence of char values.

Att skapa en metod

public static int nextInt(){...}	får ett heltal i retur
public static void println(){...}	får inte något i retur alls
public static int nextInt(){...}	använder inte några parametrar
public static void println(String s){...}	kräver en **String** som parameter

```java
/* Monolitisk.java
 * Programmet illustrerar hur man kan skriva ett monolitisk
 * kod istället för att brytta ner problemet i mindre bittar.
 */
import java.util.Scanner;
class Monolitisk{

    /* Läser in antalet element, sen läser in själva elementerna,
     * räkna upp medelvärdet och skriver ut den.
     */
    static public void main(String[] args){
     int size;
     Scanner scan = new Scanner(System.in);

     do{ // läser in ett heltal som är minst 1
         System.out.print("Enter an int that is larger or equal to 1: ");
         size=scan.nextInt();
     }while(size < 1);

     int[] numbers = new int[size];
     for(int k=0; k<numbers.length; k++){
         do{ // skriver om varje gång jag vill läser in ett heltal ??
          System.out.print("Enter an int that is larger or equal to 0: ");
          numbers[k]=scan.nextInt();
         }while(numbers[k] < 0);
     }

     long sum = 0;
     for(int p=0; p < numbers.length; p++)
         sum = sum + numbers[p];

     System.out.println("The average value is: " + ((double)sum / size));
    }
}
```

Medelvärdet

- Skriv ett program som ska fråga hur många tal som ska läsas: godtyckligt många (fast minst ett tal) mella 0 och maxvärdet för int
- Gör inläsningar till en array
- Beräkna medelvärdet; summan av talen antas rymmas i en long
- Skriv ut resultatet som bör vara av flyttalstyp

```java
/* NerBryten.java
 * Illustrerar hur programmet Monolitisk.java kan brytas ner
 */
import java.util.Scanner;
public class NerBryten{

    // Få en int array som parameter och returnera summan av dess element
    static public long computeSum(int[] nums){
     long result = 0;
     for(int p=0; p<nums.length; p++)
         result += nums[p];
     return result;
    }

    //Få en int array som parameter och returnerar medelvärdet av dess element
    static public double computeAverage(int[] nums){
     return (double)computeSum(nums)/nums.length;
    }

    //läser in ett heltal som är större eller lik med parametern "min"
    static public int readIntLargerOrEqualTo(int minimal){
     Scanner scan = new Scanner(System.in);
     int result;
     do{
         System.out.print("Enter an int that is larger or equal to " + minimal + ": ");
         result=scan.nextInt();
     }while(result<minimal);
     return result;
    }

    //läser in antalet element och deras värden, skriver upp deras medelvärde
    static public void main(String[] args){
     int[] numbers = new int[readIntLargerOrEqualTo(1)];
     for(int k=0; k<numbers.length; k++)
         numbers[k] = readIntLargerOrEqualTo(0);
     System.out.println("The average value is: " + computeAverage(numbers));
    }
}
```

Outline

Arrayer

Metoder

Räckvidd (Scope)

Eclipse

```java
/* NerBryten.java
 * Illustrerar hur programmet Monolitisk.java kan brytas ner
 */
import java.util.Scanner;
public class NerBryten{

    // Få en int array som parameter och returnera summan av dess element
    static public long computeSum(int[] nums){
     long result = 0;
     for(int p=0; p<nums.length; p++)
         result += nums[p];
     return result;
    }

    //Få en int array som parameter och returnerar medelvärdet av dess element
    static public double computeAverage(int[] nums){
     return (double)computeSum(nums)/nums.length;
    }

    //läser in ett heltal som är större eller lik med parametern "min"
    static public int readIntLargerOrEqualTo(int minimal){
     Scanner scan = new Scanner(System.in);
     int result;
     do{
         System.out.print("Enter an int that is larger or equal to " + minimal + ": ");
         result=scan.nextInt();
     }while(result<minimal);
     return result;
    }

    //läser in antalet element och deras värden, skriver upp deras medelvärde
    static public void main(String[] args){
     int[] numbers = new int[readIntLargerOrEqualTo(1)];
     for(int k=0; k<numbers.length; k++)
         numbers[k] = readIntLargerOrEqualTo(0);
     System.out.println("The average value is: " + computeAverage(numbers));
    }
}
```

```java
/* ...
 */
import java.util.Scanner;
class NerBryten{


    // ...
    static public long computeSum(int[] nums){
     long result = 0;
     for(int p=0; p<nums.length; p++)
         result += nums[p];
     return result;
    }

    //...
    static public double computeAverage(int[] nums){
     return (double)computeSum(nums)/nums.length;
    }

    //...
    static public int readIntLargerOrEqualTo(int min){
     Scanner scan = new Scanner(System.in);
     int result;
     do{
         System.out.print("Enter an int that is larger or equal to " + min + ": ");
         result=scan.nextInt();
     }while(result<min);
     return result;
    }

    //...
    static public void main(String[] args){
     int numbers = new int[readIntLargerOrEqualTo(1)];
     for(int k=0; k<numbers.length; k++)
         numbers[k] = readIntLargerOrEqualTo(0);
     System.out.println("The average value is: " + computeAverage(numbers));
    }
}
```

```java
/* ...
 */
import java.util.Scanner;
class NerBryten{
    public static int[] numbers; //ny variabel som alla kan "se"

    // ...
    static public long computeSum(/*int[] nums*/){
     long result = 0;
     for(int p=0; p<numbers.length; p++)
         result += numbers[p];
     return result;
    }

    //...
    static public double computeAverage(/*int[] nums*/){
     return (double)computeSum(numbers)/numbers.length;
    }

    //...
    static public int readIntLargerOrEqualTo(int min){
     Scanner scan = new Scanner(System.in);
     int result;
     do{
         System.out.print("Enter an int that is larger or equal to " + min + ": ");
         result=scan.nextInt();
     }while(result<min);
     return result;
    }

    //...
    static public void main(String[] args){
     numbers = new int[readIntLargerOrEqualTo(1)];
     for(int k=0; k<numbers.length; k++)
         numbers[k] = readIntLargerOrEqualTo(0);
     System.out.println("The average value is: " + computeAverage(numbers));
    }
}
```

```java
/* ...
 */
import java.util.Scanner;
class NerBryten{
    public static int[] numbers; //Klass variabel numbers

    // ...
    static public long computeSum(/*int[] nums*/){
     long result = 0;
     for(int p=0; p<numbers.length; p++)
         result += numbers[p]; //läser från klass variabeln numbers
     return result;
    }

    //...
    static public double computeAverage(/*int[] nums*/){
     return (double)computeSum(numbers)/numbers.length; //läser från klass variabeln numbers
    }

    //...
    static public int readIntLargerOrEqualTo(int min){
     Scanner scan = new Scanner(System.in);
     int result;
     do{
         System.out.print("Enter an int that is larger or equal to " + min + ": ");
         result=scan.nextInt();
     }while(result<min);
     return result;
    }

    //...
    static public void main(String[] args){
     numbers = new int[readIntLargerOrEqualTo(1)]; //tilldela ett värde till klass variabeln
     for(int k=0; k<numbers.length; k++)
         numbers[k] = readIntLargerOrEqualTo(0);
     System.out.println("The average value is: " + computeAverage(numbers));
    }
}
```

```java
/* ...
 */
import java.util.Scanner;
class NerBryten{
    public static int[] numbers; //Klass variabel numbers

    // ...
    static public long computeSum(/*int[] nums*/){
     long result = 0; //lokal variabel result
     for(int p=0; p<numbers.length; p++)
         result += numbers[p]; //läser från klass variabeln numbers
     return result;
    }

    //...
    static public double computeAverage(/*int[] nums*/){
     return (double)computeSum(numbers)/numbers.length; //läser från klass variabeln numbers
    }

    //...
    static public int readIntLargerOrEqualTo(int min){
     Scanner scan = new Scanner(System.in); //lokal variabel scan
     int result; //lokal variabel result
     do{
         System.out.print("Enter an int that is larger or equal to " + min + ": ");
         result=scan.nextInt();
     }while(result<min);
     return result;
    }

    //...
    static public void main(String[] args){
     numbers = new int[readIntLargerOrEqualTo(1)]; //tilldela ett värde till klass variabeln
     for(int k=0; k<numbers.length; k++)
         numbers[k] = readIntLargerOrEqualTo(0);
     System.out.println("The average value is: " + computeAverage(numbers));
    }
}
```

Vad blir resultatet?

```java
/* SkuggningLokalt.java
 * Programmet illustrerar hur lokala variabler
 * kan skugga klass variabler.
 */
class SkuggningLokalt{

    public static int a = 6;

    public static int addition (int c){
     int a = 2;
     return a + c;
    }

    public static void main(String[] args){
     a = addition(b);
     System.out.println(a);
    }

    public static int b = 5;
}
```

Nu då?

```java
/* SkuggningParam.java
 * Programmet illustrerar hur parametrar
 * kan skugga klass variabler.
 */
class SkuggningParam{

    public static int a = 6;

    public static int addition (int a){
     return a + b;
    }

    public static void main(String[] args){
     a = addition(b);
     System.out.println(a);
    }

    public static int b = 5;
}
```

Outline

Arrayer

Metoder

Räckvidd (Scope)

Eclipse

Eclipse

- Eclipse är en Integrated Development Environement (IDE)
- Är helt gratis
- Till alla operativsystem
- finns i flera varianter för olika språk/tillämpningar
- Sköter organisationen av filer, editering av filer, kompilering, exekvering och felsökning

Hemma

- `http://eclipse.org`
- Klicka på "Download Eclipse"
- Välj "Eclipse IDE for Java Developers"

Starta

- Starta Eclipse (på universitetet med kommandot **eclipse&**)
- Eclipse vill att du ska välja workspace, välj en mapp som alltid är tillgänglig och som kan användas hela kursen. Här kommer alla dina projekt och filer att sparas
- Om du starta fe för första gången får du upp en välkomstskärm. Klicka på den böjda pilen till höger för att bli av med den
- Du bör nu ha Eclipse igång med sisådär fyra delfönster

Översikt

- Till vänster bör du se **Package Explorer**. Här ser du dina olika projekt, och dess filer (tom första gången)
- I mitten ligger **Kodfönstret**
- Till höger ligger **Outline** här syns en översikt över dina funktioner och globala variabler
- Längst ner finns en mängd tabbar, bland andra **Problems** för felmeddelanden om din kod och **Console** där in-/utmatning till/från programmet sker när du kör det

Skriv ut array

- Skriv en metod printArray(int[] list) som givet en array av heltal, list, skriver ut dess element separerade med kommatecken
- Skriv en likadan för flyttal och strängar