

DV017A
Java-programmering
Laboration 2

Jonas Sjöberg
860224
Högskolan i Gävle,
`tel12jsg@tudent.hig.se`

Datum: 28. juni 2015
Kursansvarig lärare: Atique Ullah

Sammanfattning

Innehåll

1	Uppgift 1	3
1.1	Instruktioner	3
1.2	Källkod	3
1.3	Skärmdump	4
2	Uppgift 2	5
2.1	Instruktioner	5
2.2	Källkod	5
2.3	Skärmdump	6
3	Uppgift 3	7
3.1	Instruktioner	7
3.2	Källkod	7
3.3	Skärmdump	8
4	Uppgift 4	8
4.1	Instruktioner	8
4.2	Källkod	9
4.3	Skärmdump	11
5	Uppgift 5	11
5.1	Instruktioner	11
5.2	Källkod	13
5.3	Skärmdump	13
6	Uppgift 6	13
6.1	Instruktioner	13
6.2	Källkod	15
6.3	Skärmdump	16
7	Uppgift 7	17
7.1	Instruktioner	17
7.2	Källkod	17
7.3	Skärmdump	19
8	Uppgift 8	19
8.1	Instruktioner	19
8.2	Källkod	19
8.3	Skärmdump	24
9	Uppgift 9	24
9.1	Instruktioner	24
9.2	Källkod	24
9.3	Skärmdump	28

1 Uppgift 1

1.1 Instruktioner

1. Vad är det för speciellt med en konstant variabel och visa hur man deklarerar en sådan? Skriv ett litet program på några rader där du använder dig av en konstant variabel.

1.2 Källkod

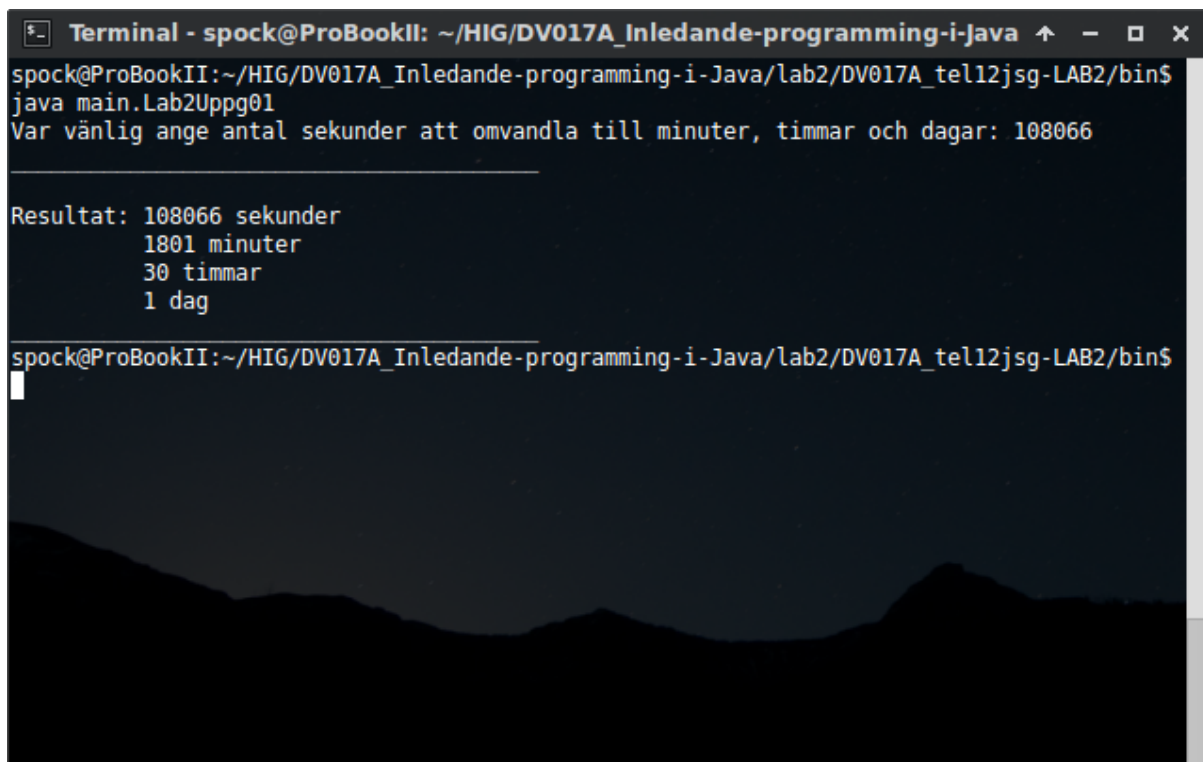
```
1 /**
2  * DV017A :: Grundläggande programmering i Java
3  * 860224 Jonas Sjöberg
4  * Högskolan i Gävle
5  * tel12jsg@student.hig.se
6  *
7  * Labb #2    Uppgift 1
8  */
9
10 package main;
11
12 public class Lab2Uppg01
13 {
14     /* Konstanta variabler håller konstanter som förblir oförändrade under
15      * programmets exekvering. Även om antal sekunder per minut med största
16      * sannolikhet alltid kommer att vara 60 så är det "good practice" att
17      * undvika att "hårdkoda magiska siffror" direkt i koden. */
18     private static final int    SECONDS_PER_MINUTE = 60;
19     private static final int    MINUTES_PER_HOUR   = 60;
20     private static final int    HOURS_PER_DAY      = 24;
21
22     /* Konstant variabel QUERY skrivs ut då användaren ombeds mata in sekunder. */
23     private static final String QUERY =
24         "Var vänlig ange antal sekunder att omvandla till minuter, timmar och dagar: ";
25
26     public static void main(String[] args)
27     {
28         //@formatter:off
29         UserInputFilter filter = new UserInputFilter();
30
31         int sekunder = filter.getPositiveInt(QUERY);
32         int minuter  = numberOfMinutesForSeconds(sekunder);
33         int timmar   = numberOfHoursForMinutes(minuter);
34         int dagar    = numberOfDaysForHours(timmar);
35
36         System.out.println("-----");
37         System.out.println("");
38         System.out.println("Resultat: " + sekunder + " sekunder");
39         System.out.println("          " + minuter + " minuter");
40         System.out.println("          " + timmar + (timmar > 1 ? " timmar" : " timme"));
41         System.out.println("          " + dagar + (dagar > 1 ? " dagar" : " dag"));
42         System.out.println("-----");
43         //@formatter:on
44     }
45
46     /**
47      * Beräknar antal minuter för ett givet antal sekunder.
48      * @param seconds    antal sekunder att beräkna
49      * @return           antal minuter
```

```

50     */
51     static int numberOfMinutesForSeconds(int seconds)
52     {
53         return seconds / SECONDS_PER_MINUTE;
54     }
55
56     /**
57      * Beräknar antal timmar för ett givet antal minuter.
58      * @param minutes    antal minuter att beräkna
59      * @return            antal timmar
60      */
61     static int numberOfHoursForMinutes(int minutes)
62     {
63         return minutes / MINUTES_PER_HOUR;
64     }
65
66     /**
67      * Beräknar antal dagar för ett givet antal timmar.
68      * @param hours       antal timmar att beräkna
69      * @return            antal dagar
70      */
71     static int numberOfDaysForHours(int hours)
72     {
73         return hours / HOURS_PER_DAY;
74     }
75 }

```

1.3 Skärmdump



```

Terminal - spock@ProBookII: ~/HIG/DV017A_Inledande-programming-i-Java
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
java main.Lab2Uppg01
Var vänlig ange antal sekunder att omvandla till minuter, timmar och dagar: 108066

Resultat: 108066 sekunder
          1801 minuter
          30 timmar
          1 dag

spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$

```

Figur 1: Körning av koden till Uppgift 1

2 Uppgift 2

2.1 Instruktioner

```
int x=4;
int y=2;

a. y=x++;
b. y=++x;
c. y\*=x;
d. y+=(x+2);
e. y+=--x;
f. y+=x++
g. x=-y;
```

2.2 Källkod

```
1 /**
2  * DV017A :: Grundläggande programmering i Java
3  * 860224 Jonas Sjöberg
4  * Högskolan i Gävle
5  * tel12jsg@student.hig.se
6  *
7  * Labb #2    Uppgift 2
8  */
9
10 package main;
11
12 public class Lab2Uppg02
13 {
14     private static int x, y;
15
16     //@formatter:off
17     public static void main(String[] args)
18     {
19         x = 4;
20         y = 2;
21
22         showResults("start"); // Beräknar uttrycken A till G och
23         /* A */ y = x++;      // skriver ut värdet av 'x' och 'y'
24         showResults("A");     // efter varje uttryck med hjälp av
25         /* B */ y = ++x;      // metoden 'showResults()'.
26         showResults("B");
27         /* C */ y *= x;
28         showResults("C");
29         /* D */ y += (x + 2);
30         showResults("D");
31         /* E */ y += --x;
32         showResults("E");
33         /* F */ y += x++;
34         showResults("F");
35         /* G */ x = -y;
36         showResults("G");
37
38         System.exit(0);
39     }
40     //@formatter:on
```

```

41  /**
42   * Skriv ut värdet hos variabler x och y.
43   * @param label    bokstav som representerar aktuellt uttryck
44   */
45  private static void showResults(String label)
46  {
47      System.out.println("Efter " + label + ": ");
48      System.out.println("    int x = " + x);
49      System.out.println("    int y = " + y);
50      System.out.println();
51  }
52 }

```

Lab2Uppg02.java

2.3 Skärmdump

```

Terminal - spock@ProBookII: ~/HIG/DV017A_Inledande-programming-i-Java/la
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
java main.Lab2Uppg02
Efter start:
    int x = 4
    int y = 2

Efter A:
    int x = 5
    int y = 4

Efter B:
    int x = 6
    int y = 6

Efter C:
    int x = 6
    int y = 36

Efter D:
    int x = 6
    int y = 44

Efter E:
    int x = 5
    int y = 49

Efter F:
    int x = 6
    int y = 54

Efter G:
    int x = -54
    int y = 54

spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$

```

Figur 2: Körning av koden till Uppgift 2

3 Uppgift 3

3.1 Instruktioner

3. Skriv ett program som räknar ut vilket av tre heltal som är det mellersta talet och sedan skriver ut detta tal. Användaren ska skriva in det tre heltalen, exempel på utskrift:

```
Skriv in första talet: *7*
Skriv in andra talet: *10*
Skriv in tredje talet: *4*
Det mellersta talet är 7
```

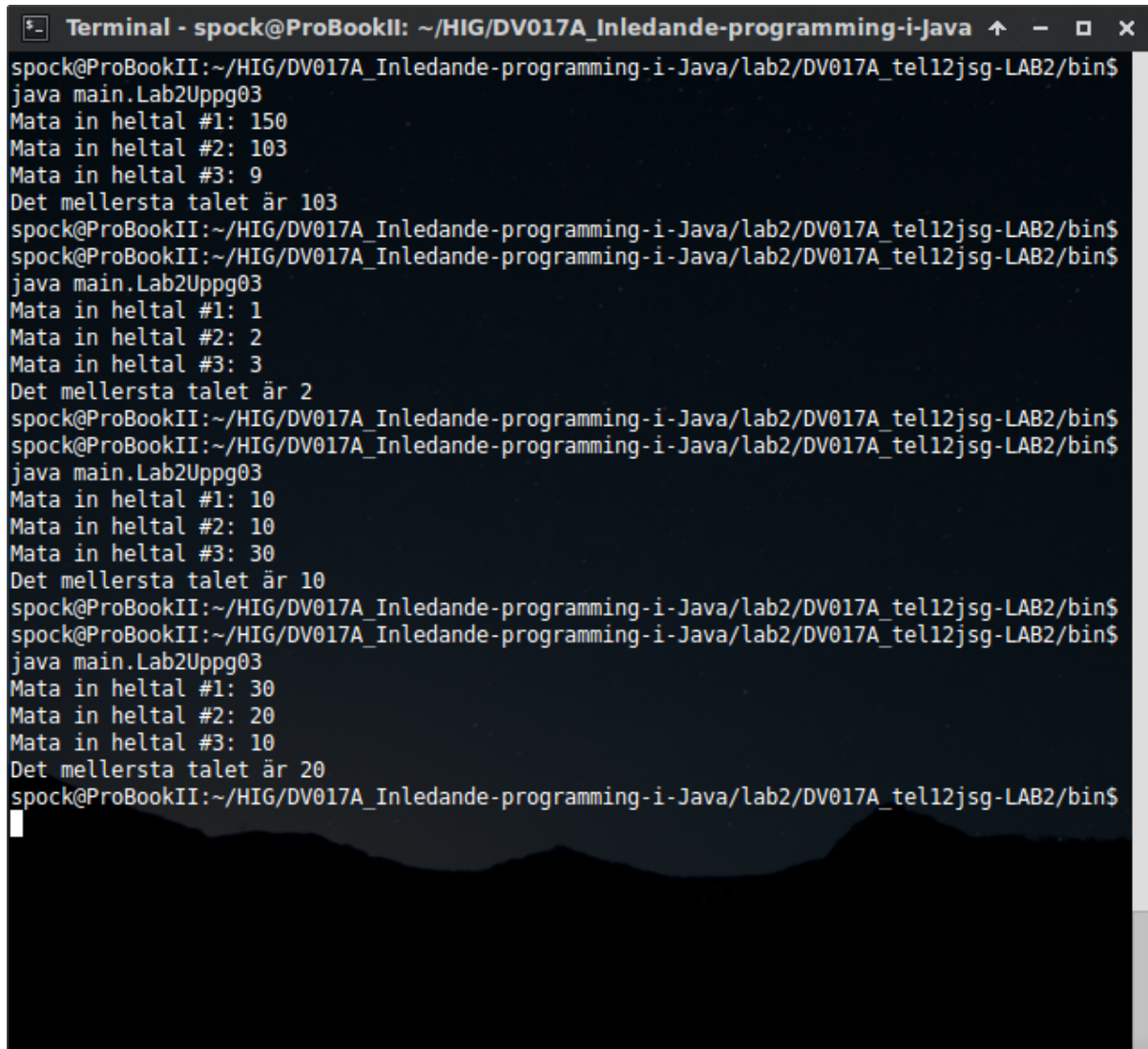
3.2 Källkod

```
1 /**
2  * DV017A :: Grundläggande programmering i Java
3  * 860224 Jonas Sjöberg
4  * Högskolan i Gävle
5  * tel12jsg@student.hig.se
6  *
7  * Labb #2    Uppgift 3
8  */
9
10 package main;
11
12 public class Lab2Uppg03
13 {
14     private static int heltal[] = new int[3];
15
16     public static void main(String[] args)
17     {
18         UserInputFilter filter = new UserInputFilter();
19         Lab2Uppg03 uppg3 = new Lab2Uppg03();
20
21         for (int i = 0; i < heltal.length; i++) {
22             heltal[i] =
23                 filter.getPositiveInt( "Mata in heltal #" + (i + 1) + ": ");
24         }
25
26         System.out.println("Det mellersta talet är " + uppg3.mellerstaTalet(heltal));
27     }
28
29     public int mellerstaTalet(int[] tal)
30     {
31         int A = tal[0];
32         int B = tal[1];
33         int C = tal[2];
34
35         if (A > B) {
36             if (B > C) return B;
37             else if (A > C) return C;
38             else return A;
39         } else {
40             if (A > C) return A;
41             else if (B > C) return C;
42             else return B;
```

```
43     }  
44 }  
45 }
```

Lab2Uppg03.java

3.3 Skärmdump



```
Terminal - spock@ProBookII: ~/HIG/DV017A_Inledande-programming-i-Java  
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$  
java main.Lab2Uppg03  
Mata in heltal #1: 150  
Mata in heltal #2: 103  
Mata in heltal #3: 9  
Det mellersta talet är 103  
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$  
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$  
java main.Lab2Uppg03  
Mata in heltal #1: 1  
Mata in heltal #2: 2  
Mata in heltal #3: 3  
Det mellersta talet är 2  
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$  
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$  
java main.Lab2Uppg03  
Mata in heltal #1: 10  
Mata in heltal #2: 10  
Mata in heltal #3: 30  
Det mellersta talet är 10  
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$  
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$  
java main.Lab2Uppg03  
Mata in heltal #1: 30  
Mata in heltal #2: 20  
Mata in heltal #3: 10  
Det mellersta talet är 20  
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
```

Figur 3: Körning av koden till Uppgift 3

4 Uppgift 4

4.1 Instruktioner

4. Skriv programmet som spelar det klassiska gissa-talet spelet. Programmet ska först slumpa ut ett tal mellan 1 och 100. Detta tal ska den som spelar klura ut. Om man gissat för lågt eller för högt så ska "Du har gissat för

lågt!" resp "Du har gissat för högt!" skrivs ut på skärmen. Man ska få fortsätta gissa tills man gissat rätt, "Rätt gissat!" skrivs då ut. Även antal gissningar ska skrivas ut. Varje gång man gissat rätt ska man få välja om man vill spela spelet igen eller avsluta. Så här kan exempelvis en körning se ut:

```
Välkommen till gissa-talet spelet!
Du ska gissa på ett tal mellan 1 och 100
gissa\> *75*
Du har gissat för högt!
gissa\> *35*
Du har gissat för lågt!
gissa\> *60*
Du har gissat för högt!
gissa\> *40*
Rätt gissat! Du har gissat 4 gånger.
Spela en gång till (j/n): *n*
Adjö!
```

4.2 Källkod

```
1 /**
2  * DV017A :: Grundläggande programmering i Java
3  * 860224 Jonas Sjöberg
4  * Högskolan i Gävle
5  * tel12jsg@student.hig.se
6  *
7  * Labb #2    Uppgift 4
8  */
9
10 package main;
11
12 import java.util.Random;
13
14 public class Lab2Uppg04
15 {
16     /* Gissa ett tal mellan RANGE_MIN och RANGE_MAX. */
17     private final static int    RANGE_MIN    = 1;
18     private final static int    RANGE_MAX    = 100;
19
20     private static int          rightAnswer;
21     private static int          noGuesses    = 0;
22     private static boolean      hasWon       = false;
23
24     /* Meddelanden till spelaren. */
25     private final static String MSG_PROMPT   = "gissa\> ";
26     private final static String MSG_WIN      = "Rätt gissat! Du har gissat ";
27     private final static String MSG_TOLOW    = "Du har gissat för lågt!";
28     private final static String MSG_TOOHIGH = "Du har gissat för högt!";
29     private final static String MSG_AGAIN    = "Spela en gång till (j/n): ";
30     private final static String MSG_BYE      = "Adjö!";
31
32     public static void main(String[] args)
33     {
34         resetGame();
```

```

35
36     do {
37         int guess = getGuess(MSG_PROMPT);
38         noGuesses++;
39
40         hasWon = checkGuess(guess, rightAnswer);
41
42         if (hasWon) {
43             if (!keepPlaying()) {
44                 System.out.println(MSG_BYE);
45                 System.exit(0);
46             } else {
47                 resetGame();
48             }
49         }
50
51     } while (!hasWon);
52 }
53
54 /**
55  * Skriver ut ett välkomstmeddelande.
56  */
57 public static void printWelcome()
58 {
59     System.out.println("-----");
60     System.out.println("Välkommen till gissa-talet spelet!");
61     System.out.println("Du ska gissa på ett tal mellan "
62         + RANGE_MIN + " och " + RANGE_MAX);
63 }
64
65 /**
66  * Hämtar en gissning från spelaren. Måste vara ett positivt heltal.
67  * @param prompt skrivs ut då spelaren omdöms mata in gissning
68  * @return ett positivt heltal
69  */
70 public static int getGuess(String prompt)
71 {
72     UserInputFilter filter = new UserInputFilter();
73     return filter.getPositiveInt(prompt);
74 }
75
76 /**
77  * Genererar ett slumpstal mellan 'min' och 'max'.
78  * @param min talets lägsta möjliga värde
79  * @param max talets högsta möjliga värde
80  * @return ett slumpmässigt heltal mellan 'min' och 'max'
81  */
82 public static int getRandomNumber(int min, int max)
83 {
84     int range = Math.abs(((max - min) + 1) + min);
85     Random rng = new Random();
86     /* '(range + 1)' kompenserar för beräkningen av range.
87     * rng.nextInt(100) ger ett tal mellan 0-99, därav en ytterligare '+1'.*/
88     return rng.nextInt(range);
89 }
90
91 /**
92  * Undersök om det gissade talet är rätt svar.
93  * @param guess gissat tal
94  * @param answer rätt svar

```

```

95      * @return          sant om gissning stämmer, annars falskt
96      */
97      public static boolean checkGuess(int guess, int answer)
98      {
99          if (guess == answer) {
100              System.out.println(MSG_WIN + noGuesses + " gånger.");
101              return true;
102          } else if (guess < answer) {
103              System.out.println(MSG_TOLOW);
104              return false;
105          } else if (guess > answer) {
106              System.out.println(MSG_TOOHIGH);
107              return false;
108          } else {
109              return false;
110          }
111      }
112
113      /**
114       * Fråga om användaren vill fortsätta spela.
115       * @return          sant om spelet ska fortsätta
116       */
117      public static boolean keepPlaying()
118      {
119          UserInputFilter filter = new UserInputFilter();
120          boolean cont = filter.getYesNoAnswer(MSG_AGAIN);
121          return cont;
122      }
123
124      /**
125       * Nollställer variabler och startar en ny spelomgång.
126       */
127      public static void resetGame()
128      {
129          hasWon = false;
130          noGuesses = 0;
131          rightAnswer = getRandomNumber(RANGE_MIN, RANGE_MAX);
132          printWelcome();
133      }
134  }

```

Lab2Uppg04.java

4.3 Skärmdump

5 Uppgift 5

5.1 Instruktioner

5. Skriv ett program som räknar förekomsterna av en viss bokstav i en sträng (String- objekt), och som sedan skriver ut summan förekomster. Strängen och vilken bokstav som ska räknas skrivs in av den som kör programmet. Tips: använd metoderna `charAt` och `length` i klassen `String`.

Exempel på hur utskriften kan se ut:

```

Skriv in den sträng du vill leta i: *Kalle har en banan*
Vilken bokstav vill du räkna: *a*

```

```
Terminal - spock@ProBookII: ~/HIG/DV017A_Inledande-programming-i-Java
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
java main.Lab2Uppg04
-----
Välkommen till gissa-talet spelet!
Du ska gissa på ett tal mellan 1 och 100
gissa\> 50
Du har gissat för högt!
gissa\> 25
Du har gissat för lågt!
gissa\> 30
Du har gissat för högt!
gissa\> 26
Du har gissat för lågt!
gissa\> 27
Rätt gissat! Du har gissat 5 gånger.
Spela en gång till (j/n): j
-----
Välkommen till gissa-talet spelet!
Du ska gissa på ett tal mellan 1 och 100
gissa\> 50
Du har gissat för lågt!
gissa\> 75
Du har gissat för lågt!
gissa\> 90
Du har gissat för högt!
gissa\> 80
Du har gissat för lågt!
gissa\> 85
Du har gissat för högt!
gissa\> 84
Rätt gissat! Du har gissat 6 gånger.
Spela en gång till (j/n): n
Adjö!
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
```

Figur 4: Körning av koden till Uppgift 4

Strängen innehåller 4 st a:n.

5.2 Källkod

```
1 /**
2  * DV017A :: Grundläggande programmering i Java
3  * 860224 Jonas Sjöberg
4  * Högskolan i Gävle
5  * tel12jsg@student.hig.se
6  *
7  * Labb #2    Uppgift 5
8  */
9
10 package main;
11
12 public class Lab2Uppg05
13 {
14     public static void main(String[] args)
15     {
16         UserInputFilter filter = new UserInputFilter();
17
18         /* Hämta textsträng från användaren. */
19         String text = filter.getString("Skriv in den sträng du vill leta i: ");
20         /* Saknar skydd mot att användaren matar in annat än en bokstav. */
21         char ch = filter.getString("Vilken bokstav vill du räkna: ").charAt(0);
22
23         int chCount = 0;
24
25         /* Gå igenom textsträngen 'text' ett tecken i taget och jämför tecknet
26          * vid position 'i' med 'ch'. Öka räknaren om tecknet matchar. */
27         for (int i = 0; i < text.length(); i++) {
28             if (text.charAt(i) == ch) chCount++;
29         }
30
31         /* Skriv ut resultatet. */
32         System.out.println(
33             "Strängen innehåller " + chCount + " st " + ch + ":n.");
34     }
35 }
```

Lab2Uppg05.java

5.3 Skärmdump

6 Uppgift 6

6.1 Instruktioner

6. Skriv ett program som räknar och skriver ut hur många förekomster det finns av vokaler a, e, i, o, u och y i en sträng. Strängen ska matas in av den som kör programmet. Du ska deklarera en räknar-variabel för varje vokal och använda dig av en switch-sats.

Exempel på körning:

```
Mata in strängen: *Du har min kursbok*
I strängen finns: 1 st a:n
```

```
Terminal - spock@ProBookII: ~/HIG/DV017A_Inledande-programming-i-Java
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
java main.Lab2Uppg05
Skriv in den sträng du vill leta i: Kissekatt mjao mjao mjao kissekatt mjao
Vilken bokstav vill du räkna: o
Strängen innehåller 4 st o:n.
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
java main.Lab2Uppg05
Skriv in den sträng du vill leta i: Fem kissar sa mjao mjao mjao mjao mjao
Vilken bokstav vill du räkna: a
Strängen innehåller 7 st a:n.
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
java main.Lab2Uppg05
Skriv in den sträng du vill leta i: aaaaaaaaaaaaaabbc
Vilken bokstav vill du räkna: a
Strängen innehåller 16 st a:n.
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
```

Figur 5: Körning av koden till Uppgift 5

```
0 st e:n
1 st i:n
1 st o:n
2 st u:n
0 st y:n
```

6.2 Källkod

```
1 /**
2  * DV017A :: Grundläggande programmering i Java
3  * 860224 Jonas Sjöberg
4  * Högskolan i Gävle
5  * tel12jsg@student.hig.se
6  *
7  * Labb #2    Uppgift 6
8  */
9
10 package main;
11
12 public class Lab2Uppg06
13 {
14     private static int count_a = 0;
15     private static int count_e = 0;
16     private static int count_i = 0;
17     private static int count_o = 0;
18     private static int count_u = 0;
19     private static int count_y = 0;
20
21     public static void main(String[] args)
22     {
23         UserInputFilter filter = new UserInputFilter();
24
25         String text = filter.getString("Mata in strängen: ");
26
27         /* Gå igenom textsträngen 'text' ett tecken i taget och gör jämförelsen
28          * i en switch-sats. Öka respektive räknare om tecknet matchar. */
29         for (int i = 0; i < text.length(); i++) {
30             switch (text.charAt(i)) {
31                 case 'a':
32                     count_a++;
33                     break;
34                 case 'e':
35                     count_e++;
36                     break;
37                 case 'i':
38                     count_i++;
39                     break;
40                 case 'o':
41                     count_o++;
42                     break;
43                 case 'u':
44                     count_u++;
45                     break;
46                 case 'y':
47                     count_y++;
48                     break;
49                 default:
50                     break;
```

```

51     }
52 }
53
54     /* Skriv ut resultatet. */
55     System.out.println("I strängen finns: " + count_a + " st a:n");
56     System.out.println("                " + count_e + " st e:n");
57     System.out.println("                " + count_i + " st i:n");
58     System.out.println("                " + count_o + " st o:n");
59     System.out.println("                " + count_u + " st u:n");
60     System.out.println("                " + count_y + " st y:n");
61 }
62 }

```

Lab2Uppg06.java

6.3 Skärmdump

```

Terminal - spock@ProBookII: ~/HIG/DV017A_Inledande-programming-i-Java
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
java main.Lab2Uppg06
Mata in strängen: Min katt heter Gibson
I strängen finns: 1 st a:n
                  2 st e:n
                  2 st i:n
                  1 st o:n
                  0 st u:n
                  0 st y:n

spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
java main.Lab2Uppg06
Mata in strängen: Gibson heter min katt
I strängen finns: 1 st a:n
                  2 st e:n
                  2 st i:n
                  1 st o:n
                  0 st u:n
                  0 st y:n

spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
java main.Lab2Uppg06
Mata in strängen: Beowulf (/ˈbeɪ.əwʊlf/; in Old English [ˈbeːo̯.wulf]) is an Old English e
pic poem consisting of 3182 alliterative long lines. It is possibly the oldest surviving
long poem in Old English and is commonly cited as one of the most important works of Old
English literature.[1] It was written in England some time between the 8th[2][3] and the
early 11th century.[4] The author was an anonymous Anglo-Saxon poet, referred to by schol
ars as the "Beowulf poet".[5]
I strängen finns: 18 st a:n
                  34 st e:n
                  24 st i:n
                  29 st o:n
                  8 st u:n
                  6 st y:n

spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$

```

Figur 6: Körning av koden till Uppgift 6

7 Uppgift 7

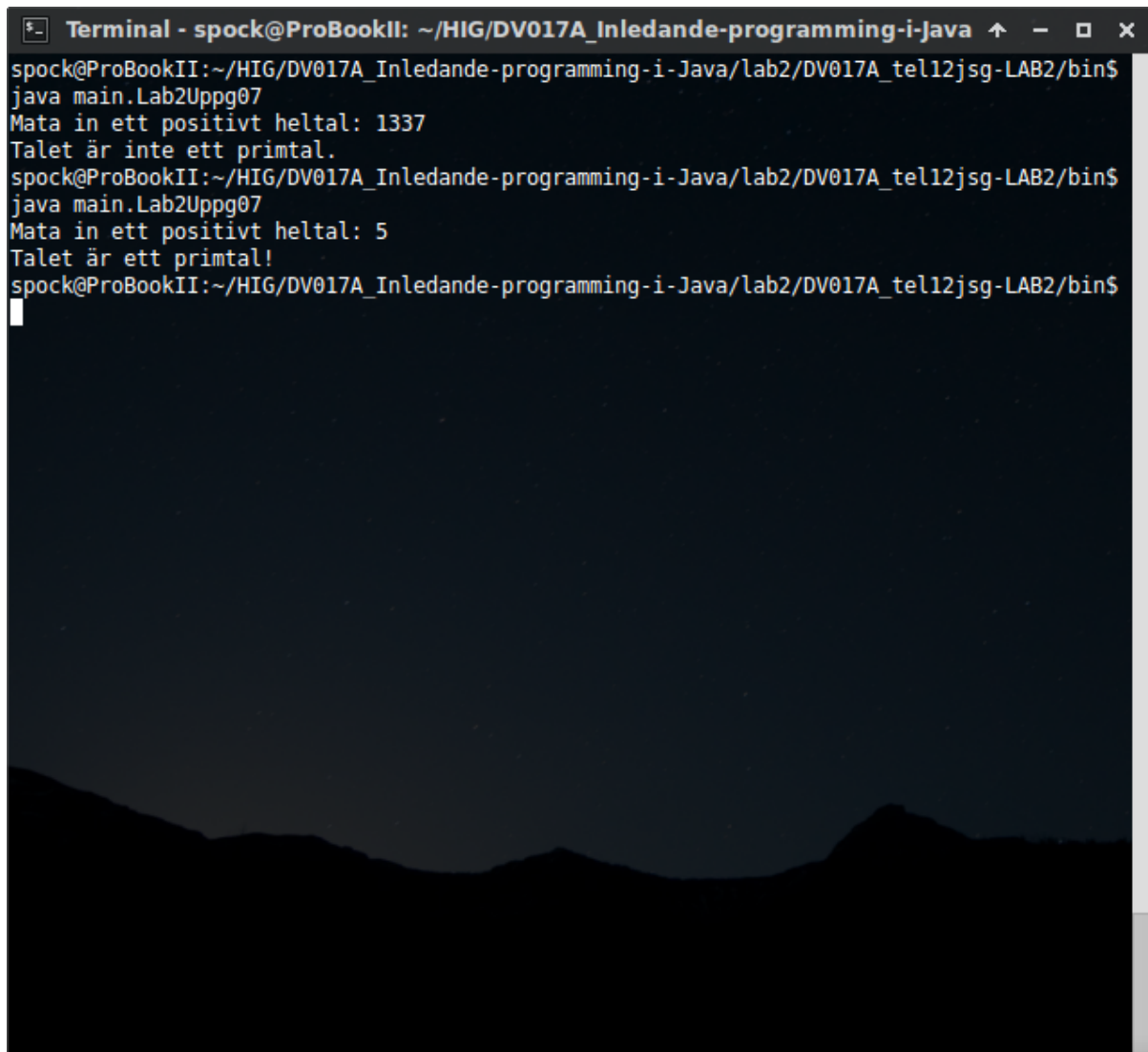
7.1 Instruktioner

7. Ett heltal är ett primtal om det bara är delbart med 1 och sig självt. Exempelvis så är 2, 3, 5 och 7 primtal, men 4, 6, 8 och 9 är ej primtal. Skriv ett program som kontrollerar och skriver ut om ett tal är primtal eller ej. Den som kör programmet ska skriva in talet. Tips: använd modulus-operatören %.

7.2 Källkod

```
1 /**
2  * DV017A :: Grundläggande programmering i Java
3  * 860224 Jonas Sjöberg
4  * Högskolan i Gävle
5  * tel12jsg@student.hig.se
6  *
7  * Labb #2    Uppgift 7
8  */
9
10 package main;
11
12 public class Lab2Uppg07
13 {
14     private static final String MSG_ISPRIME = "Talet är ett primtal!";
15     private static final String MSG_NOTPRIME = "Talet är inte ett primtal.";
16     private static int number;
17
18     public static void main(String[] args)
19     {
20         UserInputFilter filter = new UserInputFilter();
21
22         number = filter.getPositiveInt("Mata in ett positivt heltal: ");
23         System.out.println(isPrimeNumber(number) ? MSG_ISPRIME : MSG_NOTPRIME);
24     }
25
26     /**
27      * Undersöker om talet 'n' är ett primtal eller ej.
28      * @param n    talet att undersöka
29      * @return     sant om 'n' är ett primtal, annars falskt
30      */
31     public static boolean isPrimeNumber(int n)
32     {
33         for (int i = 2; i < n; i++ ) {
34             if (n % i == 0) return false;
35         }
36
37         return true;
38     }
39 }
40 }
```

Lab2Uppg07.java

A terminal window titled "Terminal - spock@ProBookII: ~/HIG/DV017A_Inledande-programming-i-Java" with standard window controls. The terminal shows a user running a Java program. The prompt is "spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin\$". The user enters "java main.Lab2Uppg07". The program prompts "Mata in ett positivt heltal: 1337" and outputs "Talet är inte ett primtal.". The user then enters "5" and the program outputs "Talet är ett primtal!". The terminal ends with the prompt "spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin\$".

```
Terminal - spock@ProBookII: ~/HIG/DV017A_Inledande-programming-i-Java
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
java main.Lab2Uppg07
Mata in ett positivt heltal: 1337
Talet är inte ett primtal.
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
java main.Lab2Uppg07
Mata in ett positivt heltal: 5
Talet är ett primtal!
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
```

Figur 7: Körning av koden till Uppgift 7

7.3 Skärmdump

8 Uppgift 8

8.1 Instruktioner

8. Skriv en klass `Person` som innehåller följande instansvariabler, deklarerera med passande datatyper:

`namn, personnummer, adress, ålder.`

I klassen ska följande metoder ingå:

- * konstruktorn `Person`, som initierar samtliga instansvariabler.
- * `byterNamn`, en metod som via parameter ändrar namnet.
- * `byterAdress`, en metod som via parameter ändrar adressen.
- * `fyllerÅr`, en metod som lägger till 1 till åldern.
- * `hamtaNamn`, en metod som returnerar namnet.
- * `hamtaPersnr`, en metod som returnerar personnumret.
- * `hamtaÅlder`, en metod som returnerar åldern.
- * `hamtaAdress`, en metod som returnerar adressen.
- * `toString`, en metod som i en snyggt formaterad sträng returnerar ett `Person`-objekts *samtliga* data. Metoden anropas genom att skriva objektets (referensens) namn.

Skriv ett testprogram som testar samtliga metoder i klassen `Person`. Programmet ska i tur och ordning utföra följande:

- * Skapa två `Person`-objekt, `person1` och `person2`. Den som kör ditt program ska mata in startvärden till de båda objektens samtliga data (instansvariabler).
- * Skriv ut de båda objektens samtliga data.
- * Låt den som kör ditt program byta namn och adress på `person1`.
- * Låt `person2` fylla år.
- * Skriv ut namnet, personnr och adressen på `person1` (skriv alltså inte ut åldern).
- * Skriv ut namnet, personnr och åldern på `person2` (skriv alltså inte ut adressen).

8.2 Källkod

```
1 /**
2  * DV017A :: Grundläggande programmering i Java
3  * 860224 Jonas Sjöberg
4  * Högskolan i Gävle
5  * tel12jsg@student.hig.se
6  *
7  * Labb #2    Uppgift 8
8  */
9
10 package main;
```

```

11
12 public class Lab2Uppg08
13 {
14     public static void main(String[] args)
15     {
16         UserInputFilter filter = new UserInputFilter();
17
18         /* Skapa nya 'Person'-objekt. */
19         prompt("Ange startvärden för \"person1\": ");
20         Person person1 = createNewPerson();
21         prompt("Ange startvärden för \"person2\": ");
22         Person person2 = createNewPerson();
23
24         /* Skriv ut information om 'Person'-objekten. */
25         prompt("\nperson1\n-----" + person1.toString());
26         prompt("person2\n-----" + person2.toString());
27
28         /* Modifiera 'person1'. */
29         person1.byterNamn(filter.getString("Ange nytt namn för person1: "));
30         person1.byterAdress(filter.getString("Ange ny adress för person1: "));
31
32         /* Låt 'person2' fylla år. */
33         prompt("\n** person2 fyller år!\n");
34         person2.fyllerAr();
35
36         /* Skriv ut information om 'Person'-objekten. */
37         // Använder '\n' newlines trots att det inte är särskilt portabelt ..
38         prompt ("\nperson1\n-----"
39             + "\nNamn:          " + person1.hamtaNamn()
40             + "\nPersonnummer: " + person1.hamtaPersnr()
41             + "\nAdress:          " + person1.hamtaAdress());
42
43         prompt ("\nperson2\n-----"
44             + "\nNamn:          " + person2.hamtaNamn()
45             + "\nPersonnummer: " + person2.hamtaPersnr()
46             + "\nÅlder :          " + person2.hamtaAlder());
47     }
48
49     /**
50      * Skapar ett nytt 'Person'-objekt.
51      * Användaren matar in samtliga startvärden.
52      * @return ett nytt 'Person'-objekt
53      */
54     public static Person createNewPerson()
55     {
56         UserInputFilter filter = new UserInputFilter();
57
58         String namn = filter.getString("Ange namn: ");
59         long persnr = filter.getPositiveLong("Ange personnummer: ");
60         String adress = filter.getString("Ange adress: ");
61         int alder = filter.getPositiveInt("Ange ålder: ") ;
62
63         return new Person(namn, persnr, adress, alder);
64     }
65
66     /**
67      * "Wrapper" runt System.out.println() för mindre skrivande.
68      * @param s textsträng att skriva ut
69      */
70     public static void prompt(String s)

```

```

71     {
72         System.out.println(s);
73     }
74 }

```

Lab2Uppg08.java

```

1  /**
2   * DV017A :: Grundläggande programmering i Java
3   * 860224 Jonas Sjöberg
4   * Högskolan i Gävle
5   * tel12jsg@student.hig.se
6   *
7   * Labb #2    Uppgift 8
8   */
9
10 package main;
11
12 public class Person
13 {
14     private String namn;
15     private long personnummer;
16     private String adress;
17     private int alder;
18
19     /**
20      * Konstruktör för klassen 'Person' som representerar en person.
21      * @param namn      personens namn
22      * @param ID        personens personnummer
23      * @param adress    personens adress
24      * @param alder     personens ålder
25      */
26     public Person(String namn, long ID, String adress, int alder)
27     {
28         this.namn = namn;
29         this.personnummer = ID;
30         this.adress = adress;
31         this.alder = alder;
32     }
33
34     /**
35      * Byter namn på personen.
36      * @param namn      personens nya namn
37      */
38     public void byterNamn(String namn)
39     {
40         this.namn = namn;
41     }
42
43     /**
44      * Ändrar personens adress.
45      * @param adress    personens nya adress
46      */
47     public void byterAdress(String adress)
48     {
49         this.adress = adress;
50     }
51
52     /**

```

```

53     * Gör personen ett år äldre.
54     */
55     public void fyllerAr()
56     {
57         alder++;
58     }
59
60     /**
61     * Hämta personens namn
62     * @return      personens namn
63     */
64     public String hamtaNamn()
65     {
66         return namn;
67     }
68
69     /**
70     * Hämta personens personnummer
71     * @return      personens personnummer
72     */
73     public long hamtaPersnr()
74     {
75         return personnummer;
76     }
77
78     /**
79     * Hämta personens ålder
80     * @return      personens ålder
81     */
82     public int hamtaAlder()
83     {
84         return alder;
85     }
86
87     /**
88     * Hämta personens adress
89     * @return      personens adress
90     */
91     public String hamtaAdress()
92     {
93         return adress;
94     }
95
96     /**
97     * Returnerar objektets data i "human-readable" format
98     */
99     public String toString()
100    {
101        // Använder '\n' newlines trots att det inte är särskilt portabelt ..
102        return ("Namn:      " + this.namn
103            + "\nPersonnummer: " + this.personnummer
104            + "\nAdress:      " + this.adress
105            + "\nÅlder:      " + this.alder + "\n");
106    }
107 }

```

Person.java

```
Terminal - spock@ProBookII: ~/HIG/DV017A_Inledande-programming-i-Java/lab2
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
java main.Lab2Uppg08
Ange startvärderna för "person1":
Ange namn: Gibson
Ange personnummer: 0705280000
Ange adress: Stora Eskissekattgatan 25A
Ange ålder: 9
Ange startvärderna för "person2":
Ange namn: Jonas
Ange personnummer: 8602101010
Ange adress: Vägenvägen 2
Ange ålder: 29

person1
-----
Namn:      Gibson
Personnummer: 705280000
Adress:    Stora Eskissekattgatan 25A
Ålder:     9

person2
-----
Namn:      Jonas
Personnummer: 8602101010
Adress:    Vägenvägen 2
Ålder:     29

Ange nytt namn för person1: Rymdhunden
Ange ny adress för person1: Orbitalgatan 0

** person2 fyller år!

person1
-----
Namn:      Rymdhunden
Personnummer: 705280000
Adress:    Orbitalgatan 0

person2
-----
Namn:      Jonas
Personnummer: 8602101010
Ålder :    30
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
```

Figur 8: Körning av koden till Uppgift 8

8.3 Skärmdump

9 Uppgift 9

9.1 Instruktioner

9. Skriv en klass `Artikel` som ska representera ett artikelslag på ett varulager. Information som ska finnas om varje artikelslag är:

`artikelnr (int)`, `artikelnamn (String)`, `lagerantal (int)`, `pris (double)`

Denna information är alltså klassens instansvariabler, passande datatyp står inom parentes. I klassen ska även finnas en **klassvariabel** som håller räkning på hur många artikelslag det finns totalt, alltså som innehåller totala antalet skapade `Artikel`-objekt:

`totArtiklar (int)`

I klassen ska följande metoder ingå:

- * konstruktorn `Artikel`, som initierar samtliga instansvariabler samt uppdaterar klassvariabelns värde.
- * `ändraNamn`, metod som via parameter ändrar på `artikelnamnet`.
- * `hamtaNamn`, metod som returnerar `artikelnamnet`.
- * `säljaArtikel`, metod som minskar lagerantalet med parameterns värde.
- * `fyllaLagret`, metod som ökar lagerantalet med parameterns värde.
- * `ändraPris`, metod som ändrar priset till parameterns värde.
- * `hamtaPris`, metod som returnerar `artikelns pris`.
- * `hamtaTotAntal`, klassmetod som returnerar antalet artikelslag.
- * `skrivInfo`, metod av returtypen `void` som skriver ut alla data om en artikel.

Skriv också ett testprogram, där du testat klassens alla metoder.

9.2 Källkod

```
1 /**
2  * DV017A :: Grundläggande programmering i Java
3  * 860224 Jonas Sjöberg
4  * Högskolan i Gävle
5  * tel12jsg@student.hig.se
6  *
7  * Labb #2    Uppgift 9
8  */
9
10 package main;
11
12 public class Lab2Uppg09
13 {
14     public static void main(String[] args)
15     {
16         UserInputFilter filter = new UserInputFilter();
17         Artikel a1 = createNewArtikel();
```



```

18     Artikel a2 = createNewArtikel();
19
20     a1.skrivInfo();
21     a2.skrivInfo();
22
23     prompt("\n* Ändrar artikeln 'a1's namn ..");
24     a1.andraNamn(filter.getString("Ange nytt artikelnamn: "));
25     prompt("Artikeln 'a1' har nu namnet: " + a1.hamtaNamn());
26
27     prompt("\n* Säljer ett antal artiklar ..");
28     a2.saljaArtikel(filter.getPositiveInt("Ange antal 'a1' att sälja: "));
29     a2.saljaArtikel(filter.getPositiveInt("Ange antal 'a2' att sälja: "));
30
31     prompt("\n* Ändrar pris på artiklar ..");
32     a1.andraPris(filter.getPositiveInt("Ange nytt pris för 'a1': "));
33     prompt("Artikeln 'a1's pris ändrat till " + a1.hamtaPris());
34
35     prompt("\n* Fyller lagret ..");
36     a1.fyllaLagret(filter.getPositiveInt("Ange antal av artikel 'a1'"));
37     a2.fyllaLagret(filter.getPositiveInt("Ange antal av artikel 'a2'"));
38
39     prompt("Totalt antal artiklar: " + Artikel.hamtaTotAntal());
40     prompt("\n* Skapar ny artikel 'a3' ..");
41     prompt("Totalt antal artiklar: " + Artikel.hamtaTotAntal());
42
43     a1.skrivInfo();
44     a2.skrivInfo();
45 }
46
47 /**
48  * Skapar ett nytt 'Artikel'-objekt.
49  * Användaren matar in samtliga startvärden.
50  * @return     ett nytt 'Artikel'-objekt
51  */
52 public static Artikel createNewArtikel()
53 {
54     UserInputFilter filter = new UserInputFilter();
55
56     int artikelnr      = filter.getPositiveInt("Ange artikelnummer: ");
57     String artikelnamn = filter.getString("Ange artikelnamn: ");
58     int lagerantal     = filter.getPositiveInt("Ange lagerantal: ");
59     int pris           = filter.getPositiveInt("Ange artikelns pris: ");
60
61     return new Artikel(artikelnr, artikelnamn, lagerantal, pris);
62 }
63
64 /**
65  * "Wrapper" runt System.out.println() för mindre skrivande.
66  * @param s      textsträng att skriva ut
67  */
68 public static void prompt(String s)
69 {
70     System.out.println(s);
71 }
72 }

```

Lab2Uppg09.java

1 /**

```

2  * DV017A :: Grundläggande programmering i Java
3  * 860224 Jonas Sjöberg
4  * Högskolan i Gävle
5  * tel12jsg@student.hig.se
6  *
7  * Labb #2    Uppgift 9
8  */
9
10 package main;
11
12 public class Artikel
13 {
14     private static int totArtiklar = 0;
15
16     private int artikelnr;
17     private String artikelnamn;
18     private int lagerantal;
19     private double pris;
20
21     /**
22      * Klass 'Artikel' representerar ett artikelslag.
23      * @param artikelnr      artikelns artikelnummer
24      * @param artikelnamn    artikelns namn
25      * @param lagerantal     antal i lager
26      * @param pris           artikelns pris
27      */
28     public Artikel(int artikelnr, String artikelnamn, int lagerantal, double pris)
29     {
30         this.artikelnr = artikelnr;
31         this.artikelnamn = artikelnamn;
32         this.lagerantal = lagerantal;
33         this.pris = pris;
34
35         totArtiklar++;
36     }
37
38     /**
39      * Ändra artikelns namn.
40      * @param artikelnamn    nytt namn
41      */
42     public void andraNamn(String artikelnamn)
43     {
44         if (artikelnamn != null) this.artikelnamn = artikelnamn;
45     }
46
47     /**
48      * Hämta artikelns namn.
49      * @return               artikelns namn
50      */
51     public String hamtaNamn()
52     {
53         return artikelnamn;
54     }
55
56     /**
57      * Sälj 'antal' exemplar av artikel och minska lagerantalet.
58      * @param antal          antal artiklar att sälja
59      */
60     public void saljaArtikel(int antal)
61     {

```

```

62     if (lagerantal - antal > 0) lagerantal -= antal;
63 }
64
65 /**
66  * Fyller lagret med 'antal' exemplar av artikeln.
67  * @param antal    antal artiklar som placeras i lagret
68  */
69 public void fyllaLagret(int antal)
70 {
71     lagerantal += antal;
72 }
73
74 /**
75  * Ändrar artikelns pris.
76  * @param pris     artikelns nya pris
77  */
78 public void andraPris(double pris)
79 {
80     if (pris > 0) this.pris = pris;
81 }
82
83 /**
84  * Hämtar artikelns pris.
85  * @return         artikelns pris
86  */
87 public double hamtaPris()
88 {
89     return pris;
90 }
91
92 /**
93  * Hämtar totalt antal artiklar.
94  * @return         totalt antal artiklar
95  */
96 public static int hamtaTotAntal()
97 {
98     return totArtiklar;
99 }
100
101 /**
102  * Skriver ut information om artikeln likt toString().
103  */
104 public void skrivInfo()
105 {
106     prompt ("\nArtikelInfo\n-----"
107             + "\nArtikelnummer: " + artikelnr
108             + "\nArtikelnamn:    " + artikelnamn
109             + "\nLagerantal:    " + lagerantal
110             + "\nPris:         " + pris + "\n");
111 }
112
113 /**
114  * Wrapper runt System.out.println() för mindre skrivande.
115  * @param s        textsträng att skriva ut
116  */
117 public static void prompt(String s)
118 {
119     System.out.println(s);
120 }
121 }

```

Artikel.java

9.3 Skärmdump

```
Terminal - spock@ProBookII: ~/HIG/DV017A_Inledande-programming-i-Java/lab: ↑ - □ ×
spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$ java main.Lab2Uppg09
Ange artikelnummer: 100
Ange artikelnamn: Weller S100
Ange lagerantal: 52
Ange artikelns pris: 1999
Ange artikelnummer: 101
Ange artikelnamn: Uni-T UT61D
Ange lagerantal: 13
Ange artikelns pris: 599

ArtikelInfo
-----
Artikelnummer: 100
Artikelnamn: Weller S100
Lagerantal: 52
Pris: 1999.0

ArtikelInfo
-----
Artikelnummer: 101
Artikelnamn: LUni-T UT61D
Lagerantal: 13
Pris: 599.0

* Ändrar artikeln 'a1's namn ..
Ange nytt artikelnamn: Lödstationen stationen
Artikeln 'a1' har nu namnet: Lödstationen stationen

* Säljer ett antal artiklar ..
Ange antal 'a1' att sälja: 4
Ange antal 'a2' att sälja: 7

* Ändrar pris på artiklar ..
Ange nytt pris för 'a1': 199
Artikeln 'a1's pris ändrat till 199.0

* Fyller lagret ..
Ange antal av artikel 'a1':10
Ange antal av artikel 'a2':50
Totalt antal artiklar: 2

* Skapar ny artikel 'a3' ..
Totalt antal artiklar: 2

ArtikelInfo
-----
Artikelnummer: 100
Artikelnamn: Lödstationen stationen
Lagerantal: 62
Pris: 199.0

ArtikelInfo
-----
Artikelnummer: 101
Artikelnamn: LUni-T UT61D
Lagerantal: 52
Pris: 599.0

spock@ProBookII:~/HIG/DV017A_Inledande-programming-i-Java/lab2/DV017A_tel12jsg-LAB2/bin$
```

Figur 9: Körning av koden till Uppgift 9