

Laborationsrapport

Författare:
Emil Åkerman

23emak01

Laboration nr.:
6

Datum:
2024-11-12

Problem 1

a) Vilka uppgifter kan man lösa med hjälp av objekt till klassen?

En arraylists antal element är dynamiskt. Det går att ta bort objekt och lägga till objekt. Detta skapar möjligheter till att hantera alla möjliga olika mängder och typer av data. Det går att sortera, söka, filtrera, utöka, det går att lagra komplexa objekt men även enklare typer av objekt.

b) Metoder för att få tag i information om objektet:

1. size()

.size() returnerar antalet element i listan. Detta kan vara användbart om man vill iterera över listan eller kontrollera om den är tom.

2. isEmpty()

.isEmpty() kontrollerar om listan är tom och returnerar en boolean true eller false beroende på om listan innehåller något element eller ej.

3. contains()

.contains() kontrollerar om ett angivet objekt finns med i listan och returnerar true eller false beroende på om objektet finns i listan eller ej. Objektet skickas med som argument.

c) Metoder för att ändra på objektet:

1. clear()

.clear() är en metod som tar bort alla element från listan. Listan kommer att vara tom efter att metoden körs.

2. remove()

.remove() tar bort den första förekomsten av elementet som man skickar med in som argument. Så här: `textList.remove("i");` - detta tar bort den första förekomsten av "i".

3. add()

.add() infogar angivet element som skickas med som argument till sista positionen(index) i listan. Så här: `textList.add("5");` - detta lägger till "5" längst sist i listan.

d) ArrayList vs Array.

Skillnader:

En Arrays antal element är statisk och antalet element i arrayen kan ej modifieras.
En Array är bättre att använda prestandamässigt om du av någon anledning inte behöver modifiera din samling element.
En ArrayList är mer flexibel och tillåter borttagning och infogning av nya element.
En ArrayLists storlek (antal element) behöver inte anges vid definition av variabeln. Medan en array måste få antalet index angivna vid definition.

```
Array:      String[] cars = new String[5];
ArrayList:  ArrayList<String> textList = new ArrayList<String>();
```

Likheter:

Båda är indexerade. Båda är iterable, det går alltså att loopa sig igenom båda typerna.
Både array och arraylist tillåter någon form av ersättning. Exempel:

```
cars[0] = "d";
textList.set(0, "d");
```

Båda dessa två, den övre är en array, och den undre är en arraylist, ersätter första elementet i sin samling med "d".

Problem 2

```
• → JavaDev git:(main) x java Problem62.java
Date:
Fri Nov 08 16:37:38 CET 2024
Day of week:
6
Day of month:
8
Day of year:
313
○ → JavaDev git:(main) x
```

```
/*
 * Modul 6
 * Problem62.java
 * Emil Åkerman
 * 2024-11-08
 */
public class Problem62 {
    Run | Debug
    public static void main(String[] args) {
        Map<String, Object> thatMap = Map.of(k1:"dateOfBirth", Calendar.getInstance());
        Calendar calendar = (Calendar) thatMap.get(key:"dateOfBirth");
        System.out.println(getDayOfWeekMonthYear(calendar));
    }

    public static String getDayOfWeekMonthYear(Calendar c) {
        StringBuilder sb = new StringBuilder();
        sb.append(str:"Date:\n");
        sb.append(c.getTime()).append(str:"\n");
        sb.append(str:"Day of week:\n");
        sb.append(c.get(Calendar.DAY_OF_WEEK)).append(str:"\n");
        sb.append(str:"Day of month:\n");
        sb.append(c.get(Calendar.DAY_OF_MONTH)).append(str:"\n");
        sb.append(str:"Day of year:\n");
        sb.append(c.get(Calendar.DAY_OF_YEAR));
        return sb.toString();
    }
}
```

Problem 3

a)

```
7 public class Problem63 {
    Run | Debug
8     public static void main(String[] args) {
9         int[] values = { 2, 1, 8, 0, 4 };
10        int[] clonedValues = values.clone();
11        swap(clonedValues, a:3, b:0);
12        for (int i = 0; i < clonedValues.length; i++) {
13            System.out.print(clonedValues[i] + " ");
14        }
15        System.out.println();
16    }
17
18    public static void swap(int[] values, int a, int b) {
19        int aa = values[a];
20        int bb = values[b];
21        values[a] = bb;
22        values[b] = aa;
23    }
24 }
25
```

PROBLEMS 2 OUTPUT POSTMAN CONSOLE TERMINAL DEBUG CONSOLE

```
→ JavaDev git:(main) x java Problem63.java
0 1 8 2 4
→ JavaDev git:(main) x
```

b)

Förklaring av "Void" som returtyp.

En funktion som har returtyp void, returnerar alltså inget värde.

Swap-funktionen kan ändra på värdena i clonedValues även fast funktionen inte returnerar något eftersom arrayer i java skickas med som referens och inte som värde.

Den "lokala" arrayen values inuti swap-funktionen och clonedValues pekar på samma på samma array i minnet. När modifikationen på den lokala arrayen values utförs, gör det samtidigt på arrayen clonedValues utanför funktionen.

Problem 4

Dot-operatorn:

Dot-operatorn används för att komma åt medlemmar ur klasser och objekt.

System.out.println(); är ett sätt att använda dot-operatorn för att komma åt metoden "println()" inuti klassen "System"

Detsamma kan göras med en array, som har inbyggda egenskaper direkt från Java-biblioteket. När vi till exempel skriver clonedValues.length, använder vi dot-operatorn för att komma åt egenskapen length på vår array. length ger oss antalet element i arrayen och är en inbyggd egenskap som är specifikt tillgänglig för arrayer i Java.

Problem 5

```
9 public class Problem65 {
10     public static void main(String[] args) {
11         System.out.println(split(text:"skriv en funktion som tar emot två String-värden",
12             marker:" "));
13         System.out.println(split(text:"132;44;255;18;129", marker:";"));
14     }
15
16     public static ArrayList<String> split(String text, String marker) {
17         StringTokenizer st = new StringTokenizer(text, marker);
18         ArrayList<String> list = new ArrayList<String>();
19         while (st.hasMoreTokens()) {
20             list.add(st.nextToken());
21         }
22         return list;
23     }
24 }
25
```

Run | Debug

PROBLEMS 1 OUTPUT POSTMAN CONSOLE TERMINAL DEBUG CONSOLE PORTS COMMENTS

● → JavaDev git:(main) x java Problem65.java
[skriv, en, funktion, som, tar, emot, två, String-värden]
[132, 44, 255, 18, 129]

○ → JavaDev git:(main) x