

Programmeringsuppgift 4

Avsikt

Avsikten med programmeringsuppgiften är att du ska träna på att använda arrayer. Uppgiften är uppdelad i fyra deluppgifter som hanterar:

- Endimensionell int-array
- Tvådimensionell int-array
- Objekt-array
- Skapa ett bibliotek som innehåller klasserna i deluppgifterna a-c.

Det är viktigt att du själv skriver all din kod och att du inte kopierar från någon studiekamrat. Men det är alltid bra att diskutera olika sätt att lösa ett problem med andra studenter.

I filen **DA339AP4HT14.zip** hittar du ett antal filer som används i programmeringsuppgiften. Dessa filer ska placeras i ett paket med namnet **p4**. Filerna är:

- *Exercise4a.java* Anropar metoder i klassen IntegerArrays
- *Exercise4b.java* Anropar metoder i klassen Integer2DArrays
- *Exercise4c.java* Anropar metoder i klassen ObjectArrays
- *Country.java* Används av Exercise4c

Redovisning

Inlämning

Din lösning av uppgiften lämnas in via It's learning *senast kl 09.00 måndagen den 17/11*. Du ska placera samtliga källkodsfiler i paketen *arrays* och *p4* i en zip-fil.

Zip-filen ska du ge namnet AAABBBP4.zip där AAA är de tre första bokstäverna i ditt efternamn och BBB är de tre första bokstäverna i ditt förnamn. Använd endast tecknen a-z när du namnger filen.

- Om Rolf Axelsson ska lämna in sina lösningar ska filen heta AxeRolP4.zip.
- Om Örjan Märkla ska lämna in sina lösningar ska filen heta MarOrjP4.zip.
- Är ditt förnamn eller efternamn kortare än tre bokstäver så ta med de bokstäver som är i namnet: Janet Ek lämnar in filen EkJanP4.zip

Granskning

Din granskning ska omfatta 1-2 A4-sidor. Granskningen ska vara som pdf-dokument och *lämnas in via It's Learning innan du redovisar tisdagen den 18/11*. Namnet på granskningen ska vara samma som zip-filen, dvs. AAABBBP4.pdf.

Granskning av Programmeringsuppgift 4

Lösning inlämnad av Eva Lind

Granskare: Einar Bok

Datum: 17/11-2014

Funktion, lösning:

Hur är funktionen i de olika klasserna, dvs hur är funktionen i de olika metoderna?
Hur bra är lösningarna? Finns det alternativa lösningar?

Indentering, metodnamn, variabelnamn mm:

Hur välskrivna är klasserna och metoderna? Är identifierarna väl valda?

Kommentarer:

Är klasser och metoder väl dokumenterade? Finns det delar i koden som borde varit kommenterade?

Redovisning

Redovisning sker *tisdagen den 18/11*. Redovisningstid publiceras på It´s learning under *måndagen den 17/11*. Kom väl förberedd till redovisningen. Kom i god tid till redovisningen så du är beredd då det är din tur. Se till att du är inloggad på en dator (eller har egen dator), att eclipse är igång på datorn och att det går att exekvera dina lösningar. Du ska också ha ett projekt vilket använder Arrays.zip.

En redovisning sker genom att:

- Studentens lösningar körs.
- Granskaren redogör för sina bedömningar
- Studenten svarar för sina lösningar
- Labhandledaren ställer kompletterande frågor
- De studenter i gruppen som inte redovisar är åhörare.

Godkänd uppgift signeras av läraren på lämpligt papper, t.ex. Redovisade uppgifter (se kurssidan). Du ska spara den signerade utskriften tills kursen är avslutad.

Om labhandledaren anser att det endast krävs *mindre komplettering för att lösningen ska godkännas* kan denna komplettering äga rum direkt efter redovisningen. Labhandledaren granskar kompletterad lösning då tiden medger.

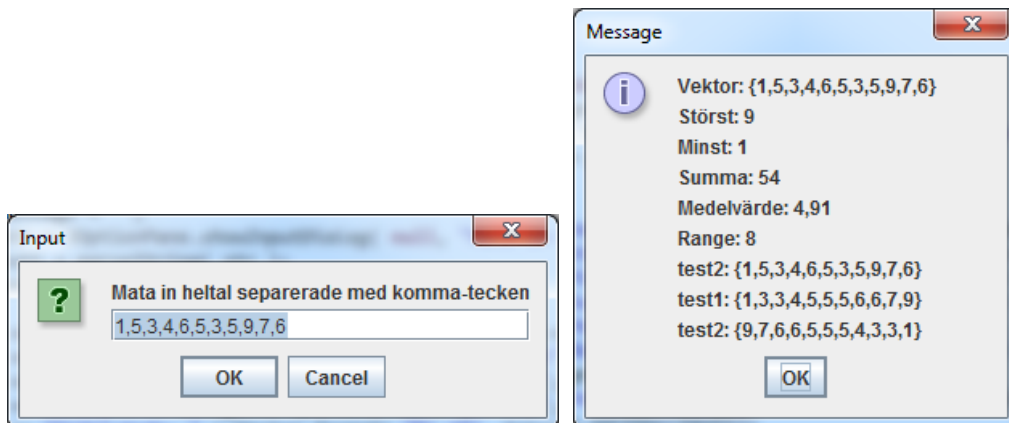
Om labhandledaren anser att det endast krävs *mindre komplettering för att granskningen ska godkännas* kan denna komplettering äga rum direkt efter redovisningen. Labhandledaren granskar kompletterad granskning då tiden medger.

Uppgift 4a

Du ska skriva klassen **IntegerArray**. Klassen ska placeras i paketet **arrays**. Klassen ska innehålla ett antal *klassmetoder*:

- **toString** vilken ska returnera en heltals-array som en sträng. Strängen ska vara på formen "{3,7,4,5}" dvs med inledande parentes, sedan talen uppräknade med komma-tecken som avgränsare, och slutligen en avslutande parentes.
Metodens huvud: `public static String toString(int[] array)`
- **max** vilken returnerar största värdet i en heltalsarray.
Metodens huvud: `public static int max(int[] array)`
- **min** vilken returnerar minsta värdet i en heltalsarray.
- **sum** vilken returnerar summan av talen i en heltalsarray.
- **average** vilken returnerar medelvärdet av talen i en heltalsarray. Observera att medelvärdet ska vara flyttalstyp.
- **range** vilken returnerar skillnaden mellan det största värdet och det minsta värdet i arrayen.
Exempel: {4,8,2,4,3,6,7,6}, min=2, max=8, returnera skillnaden: 6 (dvs $8 - 2 = 6$)
- **sortAsc** vilken sorterar talen i en heltalsarray växande, t.ex. {3,7,34,89}
- **sortDesc** vilken sorterar talen i en heltalsarray avtagande, t.ex. {89,34,7,3}
- **copy** vilken returnerar en kopia av en heltalsarray. Eftersom metoden ska returnera en ny array måste denna skapas inuti metoden.
Metodens huvud: `public static int[] copy(int[] array)`

Om du testkör din klass med programmet **Exercise4a.java** så får du ett körresultat liknande figuren nedan (då samtliga metoder är implementerade). Som du ser börjar programmet med att användaren ska mata in ett antal heltal som avgränsas med komma-tecken.



Extra metoder för hågade, ej obligatoriska

- **median** vilken returnerar medianen av talen i en heltalsarray. Medianen är det mittersta värdet i en sorterad array. Tänk på att skapa en ny array som sorteras (så den gamla ej fördärvas)

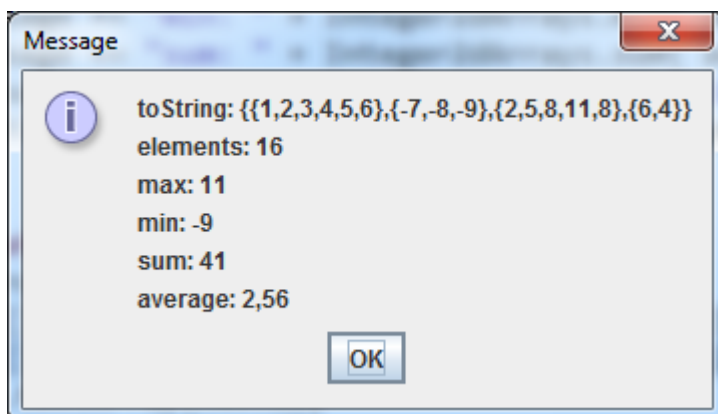
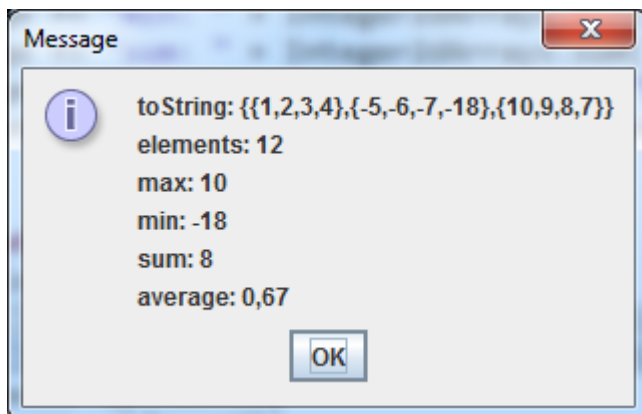
	Exempel 1	Exempel 2
1. Kopiera arrayen	{4,8,2,4,3,6,7,6}	{5,53,11,15,24,12,13}
2. Sortera arrayen:	{2,3,4,4,6,6,7,8}	{5,11,12,13,15,24,53}
3. Välj det mittersta värdet:	4 och 6	13
	Returnera medelvärde: $(4+6)/2$	Returnera 13
- **mode** vilken returnerar det mest frekventa värdet i arrayen. Om flera värden är lika frekventa så returnera ett av värdena.
Exempel: {4,8,2,4,6,3,6,7,6}, returnera 6 (3 st och därmed vanligast)

Uppgift 4b

Du ska skriva klassen **Integer2DArrays**. Klassen ska placeras i paketet **arrays**. Klassen ska innehålla ett antal *klassmetoder*:

- **toString** vilken ska returnera en heltals-array som en sträng. Strängen ska vara på formen
"`{{3,7,4,5},{4,2,10,8},{17,2,9,11}}`"
dvs med varje inre array mellan egna parenteser
Metodens huvud: `public static String toString(int[][] array)`
- **elements** vilken returnerar antalet element i en tvådimensionell int-array. Tänk på att varje rad i en array inte behöver innehålla lika många element.
Metodens huvud: `public static int elements(int[][] array)`
- **max** vilken returnerar största värdet i en tvådimensionell int-array.
Metodens huvud: `public static int max(int[][] array)`
- **min** vilken returnerar minsta värdet i en tvådimensionell int-array.
- **sum** vilken returnerar summan av talen i en tvådimensionell int-array.
- **average** vilken returnerar medelvärdet av talen i en tvådimensionell int-array. Observera att medelvärdet ska vara flyttalstyp.

Om du testkör din klass med programmet **Exercise4b.java** så får du ett körresultat liknande figurerna nedan.



Uppgift 4c

Du ska skriva klassen **ObjectArrays**. Klassen ska placeras i paketet **arrays**. Klassen ska innehålla ett antal *klassmetoder* vilka hanterar arrayer med objekt.

1. I metoderna *max* och *min* kan du förutsätta att objekten implementerar *Comparable*-interfacet.

Det innebär att metoden *compareTo* kan anropas med alla objekt i arrayen.

```
Object res = array[0];
Comparable comp;
:
comp = (Comparable)array[i];
if(comp.compareTo( res ) < 0 ) { // comp < res
    // kod då comp < res
}
```

2. Du kan i samtliga metoder anta att arrayen innehåller minst ett element.

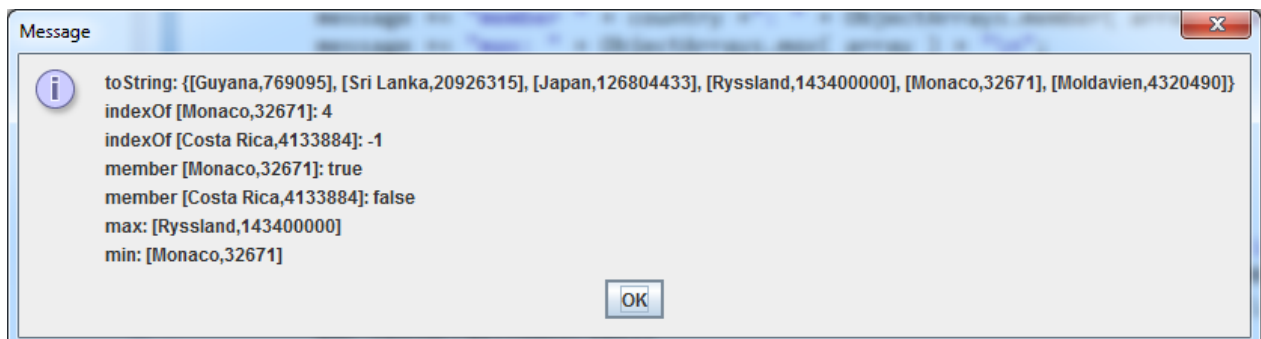
Metoder att skriva:

- **toString** vilken ska returnera en objekt-array som en sträng. Strängen ska vara på formen "{obj1, obj2, obj3}" dvs med inledande parentes, sedan objekten uppräknade med komma-tecken som avgränsare, och slutligen en avslutande parentes. obj1, obj2 och obj3 i exemplet ovan ska ersättas med resultatet vid anrop till respektive objekts *toString*-metod. Metodens huvud: `public static String toString(Object[] array)`
- **indexOf** vilken returnerar positionen för ett element (elem nedan) i en array. Om elementet inte finns i arrayen ska -1 returneras. Element ska jämföras med *equals*-metoden:

```
if( elem.equals( element i arrayen ) ) {
    // elem finns i arrayen
}
```

Metodens huvud: `public static int indexOf(Object[] array, Object elem)`
- **member** vilken returnerar *true* om ett element (elem nedan) finns i arrayen och annars *false*. Element ska jämföras med *equals*-metoden. Metodens huvud: `public static boolean member(Object[] array, Object elem)`
- **max** vilken returnerar största elementet i en objekt-array. Element ska jämföras med *compareTo*-metoden. Metodens huvud: `public static Object max(Object[] array)`
- **min** vilken returnerar minsta elementet i en objekt-array. Element ska jämföras med *compareTo*-metoden.

Om du testkör din klass med programmet **Exercise4c.java** så får du ett körresultat liknande:



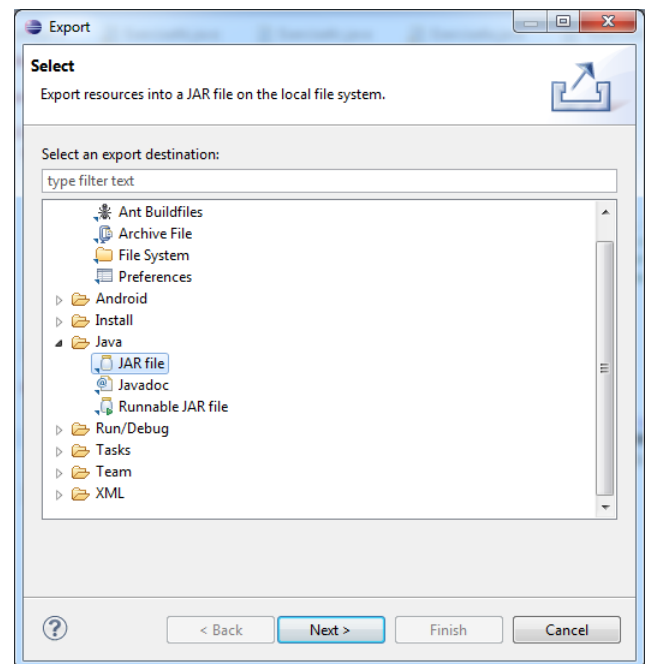
Uppgift 4d

I uppgift 4d ska du skapa en jar-fil som innehåller paketet *arrays* och de klasser som du skapat i uppgifterna 4a-4c. Eclipse sköter i huvudsak det hela åt dig.

När du gjort detta så ska du använda klasserna i paketet *arrays* i ett nytt projekt.

Högerklicka paketet *arrays* och välj **Export...** i Eclipse.

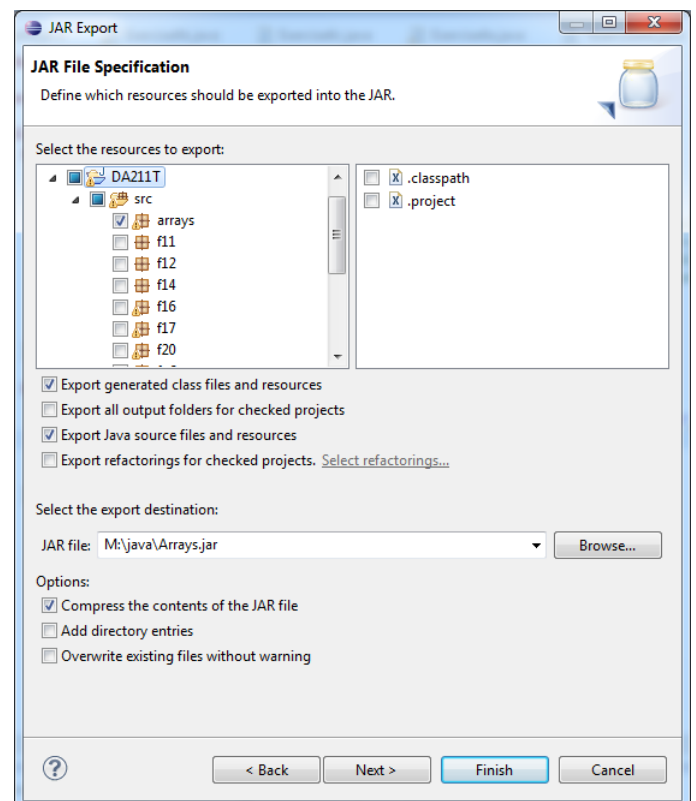
Markera **JAR file** (i mappen Java) och klicka på **Next >**.



Ange att jar-filen ska heta *Arrays.jar* och sparas i *M:\java*. Klicka sedan på **Finish**.

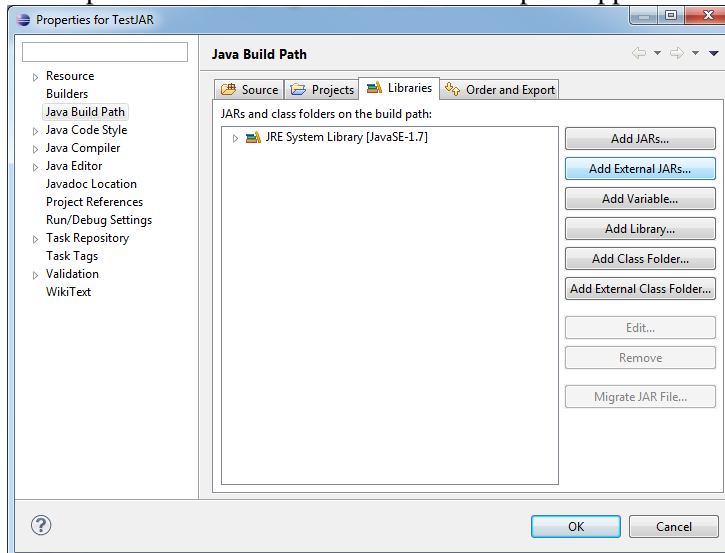
Eventuellt får du rapport om några varningar. Klicka om så är fallet på **OK**.

Nu ska du kontrollera att *Arrays.jar* finns i *M:\java*.

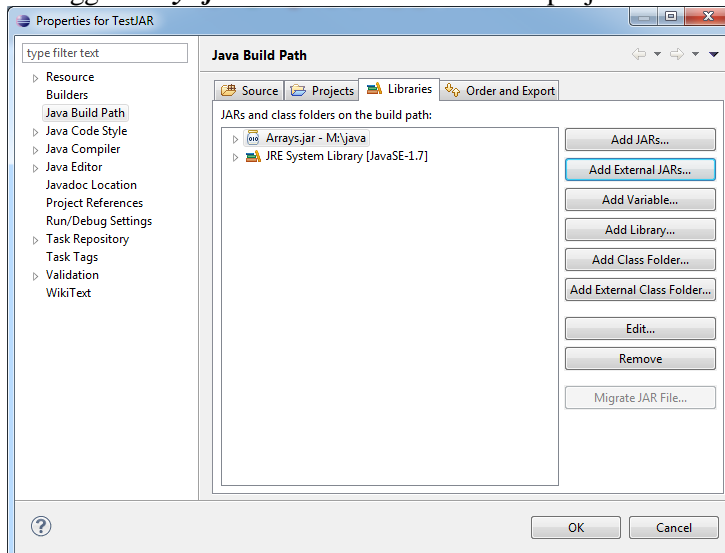


Nu är det dags att testa att klasserna i **Arrays.jar** är användbara.

1. Skapa ett nytt Java Projekt, **TestJAR**.
2. Högerklicka projektet och välj **Properties**.
3. Markera **Java Build Path** till vänster, klicka på fliken **Libraries** och klicka sedan på knappen **AddExternal JARs...**



4. En fildialog visar sig och du ska nu markera filen **M:\java\Arrays.jar** och slutligen klicka på **Open**.
5. Nu läggs **Arrays.jar** till som en biblioteksfil i projektet TestJAR.



6. Kopiera filerna i paketet **p4** till det nya projektet (kopiera hela paketet). Nu ska det gå bra att exekvera t.ex. **Exercise4a**. Nu importeras nämligen klassen **arrays.IntegerArrays** från biblioteksfilen **Arrays.jar**.