

Tentamen Programmeringsteknik I 2020-08-08, 14:00 – 21:00

Observera att tiden är förlängd!

Klasserna handlar om att representera vinddata från olika väderstationer.

Klassen `Observation` representerar en enskild observation bestående av uppgift om vindstyrka och vindriktning (reella tal).

Klassen `Station` representerar en specifik väderstation med namn (en textsträng) och en lista (dvs en `ArrayList`) av observationer (dvs `Observation`-objekt).

Klassen `StationList` innehåller en lista (en `ArrayList`) av stationer (dvs `Station`-objekt).

I alla klasserna finns uppgifter av både A- och B-typ.

Klasserna innehåller `main`-metoder som med utskrifter demonstrerar användning av metoderna. Efter varje klass finns ett kommentarfält som visar förväntade utskrifter.

Det finns också en textfil `stations.txt` med indata till demokörningen av klassen `StationList`.

Uppgifter för A-delen

I klassen `Observation`:

- A1. Deklarera de instansvariabler som behövs i klassen för att representera en observation bestående av en uppgift om vindstyrka och vindriktning.
- A2. Skriv en konstruktor som tar emot dessa två värden.
- A3. Skriv en `toString`-metod. Körexemplet visar vad den ska producera
- A4. Skriv metoden `getSpeed()` som returnerar observationens vindstyrka.
- A5. Skriv metoden `boolean aboveSpeed(double limit)` som returnerar `true` om observationens vindhastighet är över `limit`, annars `false`.

I klassen `Station`

- A6. Skriv de instansvariabler som behövs. Läs den givna koden för att se vilka namn instansvariablerna måste ha!
- A7. Skriv klart konstruktorn. Läs den givna koden för att se vad konstruktorn behöver för parametrar!
- A8. Skriv metoden `int numberOfObservations()` som returnerar antal lagrade observationer.
- A9. Skriv metoden `addObservation(Observation obs)` som lägger till en observation till listan med observationer.
- A10. Skriv metoden `double meanSpeed()` som beräknar och returnerar medelvärdet av de lagrade vindobservationerna.
- A11. För att ett vindkraftverk skall fungera får det varken blåsa för lite eller för mycket. Skriv en metod `double meanUsable(double low, double high)` som returnerar medelvinden för de vindobservationer som ligger i intervallet `[low, high]`.

I klassen `StationList`:

- A12. Skriv metoden `print` som skriver ut en lista innehållande namn, antal observationer och medelvind. Se körexemplet!

Uppgifter för B-delen

- B1. Skriv metoden `double[] speedArray()` som skapar och returnerar en *array* med alla vindvärden från en station.
- B2. Skriv klart metoden `Station addNameSorted(String name)` som letar i listan efter en station med angivet namn. Om stationen inte hittas skall den skapas och läggas in i listan så att den förblir sorterad på namn. Metoden skall returnera en referens till (det funna eller skapade) objektet.
- B3. I klassen `Observation` finns en metod `read` som används för inläsning av en observationspost dvs hastighet och riktning. Metoden har en referens till ett `Scanner`-objekt som parameter. Om det som står på tur att läsas med `Scanner`-objektet *inte* är ett tal returneras `null`. I annat fall läses hastighet och riktning och ett objekt med dessa skapas och returneras.

Om det inte går att läsa en riktning efter hastigheten så avbryts programmet med ett *RuntimeException* dvs med koden

```
throw new RuntimeException("Observation could not be read");
```

Se senare delen av `main`-metoden i klassen `Observation`

Skriv klart metoden!

- B4. Metoden `void load(String filename)` i klassen `StationList` skall läsa rader från en fil med uppgifter om stationer och observationsdata. Varje rad börjar med ett stationsnamn följt av ett antal observationsposter. Metoden skall använda `read` i `Observation` för att läsa enskilda observationer och metoden `addNameSorted` för att få in namn och observationer på rätt plats i listan. Om samma stationsnamn kommer igen skall dess observationer läggas till den redan lagrade stationen. I listan skall alltså ingen station förekomma flera gånger. Se körexemplet!

Skriv klart metoden!

- B5. a) Metoden `read` i klassen `Observation` är deklarerad som `static`. Vad innebär det?
b) Varför skriver man den inte som en ”vanlig” metod dvs utan `static`?

Svaret på denna sista fråga kan skrivas direkt i mejlet.