

SI-pass 5 OOP

Handledare: Oscar Söderlund

Mail: soscar@student.chalmers.se

1. Som du vet kan man inte ändra ett tecken i ett String-objekt. Därför skall du här göra en egen (bättre) klass `OurString` där detta är möjligt. (Att det redan finns en annan färdig klass i Java för detta får du givetvis inte utnyttja).

Klassen skall internt representera texten med ett fält (array) av tecken. Klassen skall ha en konstruktor med en text (av typen `String`) som parameter. Konstruktorn skall initiera det interna fältet. Dessutom skall klassen ha instansmetoderna:

- `int antal()` Ger antalet tecken i strängen.
- `void byt(int i, char c)` Ersätter det `i`te tecknet med `c`. Om `i` ligger utanför gränserna för strängen skall en exception av standardklassen `IndexOutOfBoundsException` genereras.
- `void utökaMed(String s)` Läger in tecknen i `s` efter dem som redan finns i texten. Detta kräver att du bildar ett längre fält.
- `String toString()` Returnerar innehållet i objektet som en `String`.

2. Man har i en textfil samlat uppgifter om ett antal personer. För varje person finns två rader i filen. På första raden står personens namn och adress och på andra raden finns personens ålder, längd och vikt. Längden anges i cm och vikten i kg.

Man vill göra en medicinsk studie av överviktiga personer och söker därför personer vilkas s.k. body mass index (förkortat BMI) överstiger värdet 30. BMI beräknas enligt formeln m/h^2 där m är vikten i kg och h längden i m.

Skriv ett program som läser filen med personuppgifter. Programmet skall skapa en ny textfil som bara innehåller uppgifterna för de personer vilkas BMI överstiger 30. Filernas namn skall läsas in från kommandoraden.

3. Ett primtal är ett tal som endast är jämnt delbart med 1 och med sig själv. En teknik att finna primtal går under namnet Eratosthenes såll. Den går till på följande sätt:

Man använder ett fält där elementen har typen `boolean`. Från början sätts element nummer 0 och 1 till `false` och alla övriga element i fältet till `true`. Därefter löper man igenom fältet med början på index 2. När man stöter på ett element (låt oss säga att det har nummer `i`) som har värdet `true` gör man följande. Löp igenom resten av fältet (fr.o.m. `i+1` till fältets slut) och sätt varje element som har den egenskapen att dess index är en jämn multipel av `i` till `false`. När man t.ex. kommer till element nr. 5 så skall elementen 10, 15, 20 etc. i fältet ges värdet `false`. När denna process är avslutad visar de element i fältet som fortfarande har värdet `true` vilka index som är primtal.

Konstruera ett program som använder denna teknik för att beräkna och skriva ut alla primtal som är mindre än eller lika med 1000. Utskriften skall ske i ett kommandofönster.

4. Antag att alla resultat från en tenta finns sparade i en textfil. Varje person står på en egen rad i filen och informationen är skriven på följande form:

```
personnummer efternamn förnamn u1 u2 u3 u4 u5 u6
```

u1, u2, ... står för poängen på de enskilda uppgifterna. (Antag att bara hela poäng delas ut.) På tentan kunde man få något av betygen U, 3, 4, eller 5. För att få betyget 5 krävdes minst 48 poäng, för en 4:a minst 36 poäng och för en 3:a minst 24 poäng.

Nu vill man ha informationen i filen uppdelad så att de som tentat delas upp i fyra separata filer. De med betyget 5 skall hamna i en fil, de med betyget 4 i en annan fil, de med betyget 3 i en tredje fil och de som blev underkända i en fjärde fil. Raderna i de fyra filerna skall ha samma utseende som i den ursprungliga filen, fast sist på varje rad *skall den totala poängen för eleven också skrivas ut*. Uppgiften är att skriva ett program som gör denna uppdelning.

Namnet på den ursprungliga filen skall läsas in från kommandoraden. (Om du inte vet hur man gör detta får du, mot ett poängavdrag, läsa in filnamnet från en dialogruta.) Du får alltså inte i ditt program förutsätta att den ursprungliga filen har ett visst, förutbestämt namn.

De fyra filer som skall skapas av programmet skall ha samma namn som den ursprungliga filen fast med tilläggen 5, 4, 3 resp. u omedelbart före punkten framför suffixet. Om den ursprungliga filen t.ex. heter **resultat.txt** så skall alltså de fyra filerna heta **resultat5.txt**, **resultat4.txt**, **resultat3.txt** resp. **resultatu.txt**.

Om den ursprungliga filens namn inte innehåller någon punkt skall tilläggen 5, 4, 3 resp. u finnas sist i filnamnen. Om den ursprungliga filen t.ex. heter **resultat** så skall alltså de fyra filerna heta **resultat5**, **resultat4**, **resultat3** resp. **resultatu**.