

Omkontrollskrivning, Programmeringsteknik för D/C

2014–04–29, 8.00–13.00

Anvisningar: Preliminärt ger uppgifterna $15 + 10 + 5 + 10 = 40$ poäng. För godkänt betyg krävs ungefär 25 poäng. Tillåtet hjälpmedel: Java-snabbreferens.

Betyget på denna skrivning utgör betyget på kontrollskrivningarna, oavsett resultatet på tidigare skrivningar (dock sänks inte ett tidigare godkänt betyg). Resultatlistan anslås på institutionens anslags-tavla när rättningen är klar.

Introduktion

Alla uppgifterna handlar om betygsbedömning i grundskolan. År 2011 ändrades betygssystemet till en betygsskala med sex steg med beteckningarna A, B, C, D, E och F. Betyg A är högsta betyg. A–E är godkända resultat medan F är ej godkänt.

Betygssättning på både enskilda prov och i hela ämnen sker med hjälp av betygsmatriser. En betygsmatris innehåller betygskraven för betygen A, C och E för ett antal aspekter av ett ämne (se tabell 1). Betygskraven för en aspekt är alltid inkluderande, dvs ett krav för ett lägre betyg är alltid en delmängd av kravet för ett högre betyg. Så om man uppfyller kravet för betyg A uppfyller man automatiskt kravet för betyg C och E, om man uppfyller kravet för betyg C uppfyller man automatiskt kravet för betyg E.

Betygsmatrisen används för att sätta betyg:

- För betyg A måste betygskravet för betyg A vara uppfyllt för samtliga aspekter.
- För betyg B måste betygskravet för betyg C vara uppfyllt för samtliga aspekter samt betygskravet för betyg A för mer än hälften av aspekterna.
- För betyg C måste betygskravet för betyg C vara uppfyllt för samtliga aspekter.
- För betyg D måste betygskravet för betyg E vara uppfyllt för samtliga aspekter samt betygskravet för betyg C för mer än hälften av aspekterna.
- För betyg E måste betygskravet för betyg E vara uppfyllt för samtliga aspekter.
- Är inte betygskravet för betyg E uppfyllt för samtliga aspekter ges betyg F.

Klassen Aspekt, som beskriver en aspekt, är färdigskriven. Du behöver bara känna till att varje aspekt har ett unikt id-nummer (används i uppgift 4).

<i>Aspekt</i>	<i>Betygskrav E</i>	<i>Betygskrav C</i>	<i>Betygskrav A</i>
Läsa med flyt	Eleven kan läsa skönlitteratur och sakprosatexter för barn och ungdomar med flyt genom att använda lässtrategier på ett i huvudsak fungerande sätt.	Eleven kan läsa skönlitteratur och sakprosatexter för barn och ungdomar med gott flyt genom att använda lässtrategier på ett ändamålsenligt sätt.	Eleven kan läsa skönlitteratur och sakprosatexter för barn och ungdomar med mycket gott flyt genom att använda lässtrategier på ett ändamålsenligt och effektivt sätt.
Skriva texter	Eleven kan skriva olika slags texter med begripligt innehåll, i huvudsak fungerande struktur och viss språklig variation.	Eleven kan skriva olika slags texter med relativt tydligt innehåll och relativt väl fungerande struktur samt förhållandevis god språklig variation.	Eleven kan skriva olika slags texter med tydligt innehåll, väl fungerande struktur samt god språklig variation.
Osv

Tabell 1: Betygsmatris för ämnet svenska.

1. I fortsättningen behandlar vi betygssättning av prov. Elever som deltar i prov beskrivs av den färdigskrivna klassen `Pupil`, som bland annat har metoder `String getPnbr()` och `String getName()` som tar reda på elevens personnummer och namn.

Vid ett prov har varje elev sin egen bedömningstabell där man noterar hur väl de olika aspekterna är uppfyllda. En sådan tabell beskrivs av klassen `GradeTable`:

`GradeTable`

```
/** Skapar en bedömningstabell för eleven pupil. Tabellen har nbrAspects
    rader */
GradeTable(Pupil pupil, int nbrAspects);

/** Tar reda på eleven */
Pupil getPupil();

/** Noterar att eleven klarat betygsnivå grade (A, C eller E) på aspekten
    med nummer a */
void mark(int a, char grade);

/** Ger true om eleven klarat betygsnivå grade (A, C eller E) på aspekten
    med nummer a */
boolean isMarked(int a, char grade);

/** Ger elevens sammanvägda betyg (A-F) */
char getTestGrade();
```

Implementera klassen `GradeTable`. Anvisningar:

- Aspekterna numreras 0, 1, 2, ...
- Bedömningstabellen ska implementeras med hjälp av en boolean-matris med en rad för varje aspekt och tre kolumner för betygsnivåerna E, C och A.
- Kom ihåg att avklarad betygsnivå A \Rightarrow nivå C och E, avklarad nivå C \Rightarrow nivå E.

2. Klassen `Test` hanterar bedömningen av ett prov. Olika prov testar olika aspekter.

`Test`

```
/** Skapar ett prov som ska bedöma aspekterna aspects */
Test(String title, String date, ArrayList<Aspect> aspects);

/** Skapar en bedömningstabell för eleven pupil och sparar tabellen i
    en lista */
void addPupil(Pupil pupil);

/** Markerar att eleven med personnumret pNbr klarat betygsnivå grade (A, C
    eller E) på aspekten med nummer a */
void mark(String pNbr, int a, char grade);

/** Skriver ut en resultatlista: en rubrikrad med provets namn och datum,
    sedan en rad för varje elev med namn, personnummer och betyg */
void printResults();
```

Implementera klassen.

3. Implementera en metod i klassen `Test` som tar reda på vilken aspekt som varit enklast att klara av. Metoden ska ha följande rubrik:

```
/** Tar reda på vilken aspekt som varit enklast på provet */
Aspect getEasiestAspect();
```

För att avgöra hur svår en aspekt är räknar man hur många betygsnivåer som eleverna totalt klarat av. En elev som klarat betygsnivå A ger tre "poäng", en elev med betygsnivå C två poäng, en elev med betygsnivå E en poäng. Den aspekt som fått flest poäng är enklast. Om flera aspekter är lika enkla så spelar det ingen roll vilken av dem som returneras.

4. En lärare som ger och bedömer ett prov skriver inte in resultaten direkt i ett `Test`-objekt. I stället skriver läraren in resultaten i ett ark i ett kalkylprogram. Arket exporteras sedan till en textfil. Textfilen innehåller:

- En rad med provets namn och datum,
- en rad med id-nummer för alla aspekter som bedöms i provet (se nedan),
- en rad för varje elev med personnummer och betygsnivå för varje aspekt (A, C, E som vanligt, streck för ej avklarad aspekt).

Exempel på textfil för ett prov där fem studenter deltog och fem aspekter bedömdes:

```
Svenska 2014-04-29
1123 1124 1131 1132 1140
010430-1234 A A C A C
010224-2345 A C C C C
001204-3456 C C - - C
010316-4567 E C E C C
010212-5678 A C C C E
```

Skriv en `main`-metod som läser in textfilen (som heter `testresult.txt`) och producerar en resultatlista för provet (med metoden `printResults` i klassen `Test`). Du kan förutsätta att textfilen följer det angivna formatet och att alla data är korrekta (orealistiskt, men det blir mindre att skriva).

Den färdigskrivna klassen `TestRegister` hanterar allt som har med prov att göra. Till exempel har klassen tillgång till alla aspekter och alla elever. Alla prov som görs måste registreras i registret. Metoder som utnyttjas i denna uppgift:

`TestRegister`

```
/** Skapar ett provregister */
TestRegister();

/** Ger aspekten med givet id-nummer, null om det inte finns någon
    sådan aspekt */
Aspect getAspect(int aspectId);

/** Ger eleven med givet personnummer, null om det inte finns någon
    sådan elev */
Pupil getPupil(String pNbr);

/** Registrerar ett nytt prov */
void register(Test test);
```

Tips om inläsning av betygsnivåer: det finns inte någon `Scanner`-metod för att läsa ett enskilda tecken. Men man kan läsa en sträng och hämta tecknet ur strängen med `charAt`.