

Omkontrollskrivning, Programmeringsteknik för D/C

2013-01-08, 14.00-19.00

Anvisningar: Preliminärt ger uppgifterna $10 + 24 + 6 = 40$ 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 anslagstavla när rättningen är klar.

Felrapporteringsystem används inom många programutvecklingsprojekt. När fel upptäcks av användare till ett program så rapporteras felen in i systemet, som sedan hjälper programutvecklarna att hålla reda på och följa upp fel så att de blir åtgärdade. Vi ska i denna tentamen skriva delar av programkoden till ett litet felrapporteringsystem. Systemet heter Bugs.

I Bugs representeras en felrapport (eng: ticket) av en klass `Ticket`. En rapport består av ett unikt id, en titel, en flagga som anger om rapporten är öppen eller stängd, datum då rapporten skapades och en lista med inlägg (eng: item). Ett objekt som innehåller dagens datum skapas med `new Date()`. Felrapporten är öppen innan problemet som rapporteras är löst och stängd då problemet har blivit löst. En nyskapad felrapport är alltid öppen.

Felrapporten innehåller också en lista med inlägg. Detta är kommentarer som utvecklare och användare skriver om problemet. Felet beskrivs vanligen i det första inlägget. Ytterligare inlägg görs därefter när felet åtgärdas eller när information relevant till det beskrivna felet uppdagas. En nyskapad felrapport har en tom lista med inlägg. Metoden `toString` returnerar felrapporten som en sträng. Denna ska innehålla titel, id, information om rapporten är öppen/stängd och datum på formen `titel (id) - öppen/stängd - datum`. Datum ska vara på den form som `Date.toString()` returnerar. Datum kan jämföras med `Date.compareTo`. Även inlägg som gjorts ska finnas med (ett inlägg per rad). Exempel på utskrift (raderna som börjar med Nisse och Olle är inlägg):

Programmet kraschar (ID 2) - ÖPPEN - Tis Dec 11 10:26:53 CET 2012

Nisse: Jag får `IOException` när jag vill öppna en fil.

Olle: Jag tittar på problemet

Java-programmet kompilerar inte (ID 1) - STÄNGD - Tis Dec 11 15:11:21 CET 2012

Nisse: Jag kan inte kompilera `int a = "Hello";`

Tips: `"\n"` betyder radbrytning i en sträng.

`Ticket`

```
/** Skapar en öppen felrapport med idnummer id och titeln title. */
Ticket(int id, String title)

/** Stänger felrapporten. */
void close()

/** Returnerar true om felrapporten är öppen, annars false. */
boolean isOpen()

/** Returnerar felrapportens id-nummer. */
int getId()

/** Lägger till inlägget item till felrapporten. */
void add(Item item)

/** Returnerar en lista med felrapportens inlägg. */
ArrayList<Item> getItems()

/** Jämför två felrapporter. Två felrapporter anses lika om de är båda öppna
    eller stängda och datum är lika. En öppen felrapport är alltid större
    än en stängd. Om båda rapporterna är öppna eller stängda avgör datum
    relationen. Returnerar 0 då this==t, 1 då this>t, -1 då this<t. */
public int compareTo(Ticket t)

/** Returnerar en sträng som beskriver felrapporten. */
String toString()
```

Inlägg representeras av klassen `Item`. Ett inlägg har alltid en ägare (namnet på personen som gjort inlägget) och en inläggstext (alltid en textrad). Metoden `toString` ger en sträng med inlägget på formen `ägare: meddelande`.

Item

```
/** Skapar ett nytt inlägg. */  
Item(String owner, String message)  
  
/** Returnerar ägaren till inlägget. */  
String getOwner()  
  
/** Returnerar en sträng som beskriver inlägget. */  
String toString()
```

Klassen Bugs lagrar och hanterar felrapporter. I nuvarande system går det att mata in en ny felrapport, uppdatera en befintlig rapport med ett nytt inlägg, få två sammanställningar utskrivna, samt stänga en rapport. Notera att rapporter identifieras med unika id, så metoder som tar id som parameter söker upp befintlig felrapport och applicerar sin operation på denna. För att skapa unika id används enklast en räknare som uppdateras för varje ny rapport. Metoden `printParticipantTickets` returnerar en lista med felrapporter där en viss person gjort inlägg. Om till exempel Alfred angivits så ska bara felrapporter där Alfred är ägare av inlägg returneras. Notera att en felrapport bara ska förekomma en gång även om Alfred gjort flera inlägg i denna. Listan som returneras av `printParticipantTickets` ska vara sorterad efter `Ticket.compareTo`. Även utskriften från `printTickets` ska vara sorterad.

Felrapporter som Alfred bidragit till:

Mitt program har slutat fungera (ID 4) - ÖPPEN - Tis Dec 11 16:36:43 CET 2012

Alfred: Mitt program har slutat fungera

Olle: På vilket sätt?

Bugs

```
/** Skapar ett nytt tomt felhanteringsobjekt. */  
Bugs()  
  
/** Skapar en ny felrapport med titeln title. Ger felrapporten ett unikt  
    id-nummer som returneras. */  
int newTicket(String title)  
  
/** Lägger till inlägget item i felrapporten med id-nummer id om det  
    finns en sådan felrapport. Om felrapporten finns returneras true,  
    annars false. */  
boolean updateTicket(int id, Item item)  
  
/** Stänger felrapporten med id-nummer id om den finns. */  
void closeTicket(int id)  
  
/** Skriver ut felrapporter enligt:  
    Om opened har värdet true skrivs alla öppna felrapporter ut.  
    Om closed har värdet true skrivs alla stängda felrapporter ut. */  
void printTickets(boolean opened, boolean closed)  
  
/** Returnerar en lista med de felrapporter där personen owner bidragit. */  
ArrayList<Ticket> getParticipantsTickets(String owner)
```

Uppgifter

De angivna maxpoängen nedan är preliminära.

1. (10 poäng): Implementera klassen Ticket.
 2. (24 poäng): Implementera klassen Bugs. Användning av färdiga rutiner i Javas klassbibliotek för sökning och sortering är inte tillåtet. Sortering implementeras enklast i en hjälpmetod som anropas när lagrade felrapporter uppdaterats eller modifierats.
 3. (6 poäng) Implementera ett huvudprogram Test som testat klassen Bugs enligt:
 1. Skapar två felrapporter med vardera ett respektive två inlägg. Personen Alfred ska stå som ägare till två av inläggen.
 2. Stänger en felrapport.
 3. Skriver ut stängda felrapporter.
 4. Skriver ut felrapporter där Alfred bidragit.
-