T-106.1207 Grundkurs i Programmering
Tent 3.1.2007

Skriv i övre kanten på varje svarspapper kursens namn och kod, tentdatum, studentends namn, studentnummer och namnteckning.

Uppgift 1 (24 poäng)

Definiera kort, högst med tre rader, följande programmerings- och Java-termer enligt det som undervisats på kursen:

1. Abstrakt klass
2. Undantag (exception)
3. Rekursion
4. Konstruktor-metod
5. Kompilator
6. UML
7. `null`
8. Automatisk typkonversion (upcast, widening). Ge exempel.

Uppgift 2 (10 poäng)

Skriv en kort beskrivning om metoder i Java. Vad är de? Vad används de till? Hur deklarerar man metoder? Skriv en exempelmetod och förklara hur den fungerar.

Vad menas med överlagring av metoder (method overloading)? Vad menas med ersättning av metoder (method overriding)? Vad innebär det att en metod deklareras vara "abstract", "final" eller "static"? Vad betyder "dynamic bindning"?

Uppgift 3 (18 poäng)

Objekt av följande klass hanterar listan av de bästa resultaterna i något spel:

```
public class HighScores {
  private int n_scores;

  class Score {
    String name;
    int points;
  }
  private Score[] scores;

  HighScores(int n_scores) {
    n_scores = n_scores;
    scores = new Score[n_scores];
  }

  public void insert(String name, int points) {
    Score score = new Score();
    score.name = name;
    score.points = points;

    for (int i = 0; i < n_scores; i++)
      if (scores[i] == null) {
        scores[i] = score;
        return;
      } else if (scores[i].points < score.points) {
        Score tmp = scores[i];
        scores[i] = score;
        score = tmp;
      }
  }

  public String toString() {
    String s = "";
    for (int i = 0; i < n_scores; i++)
      if (scores[i] != null)
        s += i + 1 + ". " + scores[i].name
             + " " + scores[i].points + "\n";
    return s;
  }
}
```

Svara de följande frågorna:

1. Klassen fungerar inte rätt. Vilket programmeringsfel finns i klassens konstruktor? Berätta hur det leder till ett fel då programmet körs, och hur programmeringsfelet kunde korrigeras.

2. Efter att du korrigerat programmet, vad skrivs på skärmen då de följande satserna körs

```
HighScores topten = new HighScores(10);
topten.insert("Markus", 20);
topten.insert("Kenneth", 30);
System.out.println(topten);
```

3. Vad skulle hända om `return`-satsen skulle lämnas bort i metoden `insert`? Vad skulle de föregående satserna skriva på skärmen?

4. Ändra på metoden `toString` så, att om samma spelare har två eller flera resultat efter varann, skrivs `-''-` istället för att upprepa spelarens namn.

5. Skriv en metod som kan användas för att fråga om ett givet poängantal räcker till att komma på listan. Förutom programsatserna i metoden, måste du själv hitta på ett beskrivande namn och passligt returvärdetyp för metoden.

6. Följande metoden läser in listan från en fil.

```
public static HighScores readFromFile(String filename)
    throws HighScoreFormatError, java.io.FileNotFoundException {
  Scanner sc = new Scanner(new File(filename));
  String name;

  if (!sc.hasNextInt())
    throw new HighScoreFormatError("Missing number of scores");
  HighScores scores = new HighScores(sc.nextInt());
  for (int i = 0; i < scores.n_scores && sc.hasNextLine(); i++) {
    name = sc.nextLine();
    if (!sc.hasNextInt())
      throw new HighScoreFormatError("Missing points");
    scores.insert(name, sc.nextInt());
  }
  return scores;
}
```

   Vad händer om filen inte innehåller poängantalen i rätt ordning? Vad händer om filen innehåller överlops rader av spelare och poäng? Varför är metoden deklarerad `static`? Skriv ett exempel hur metoden anropas, inklusive hantering av undantag.

Den andra frågan är värd två poäng, de två sista frågorna är värda fyra poäng, alla andra frågor är värt tre poäng.

Uppgift 4 (8 poäng)

Meteorologiska institutet samlar data från många olika källor: Automatiska observationsstationer skickar varje timme ett meddelande som innehåller information om luftens temperatur, lufttrycket, och luftfuktigheten samt vindens hastighet och riktning. Väderleksballonger skickar samma information från olika höjder. Från bemannade observationsstationer kan man dessutom på skild begäran få information om snöns höjd och mängd, samt pollenmätningar. All denna information sparas ett dygn i en meddelandekö för att kunna användas i olika prognosprogram.

Rita ett UML-diagram som innehåller klasserna MeddelandeKö, ObservationsMeddelande, SnöMeddelande, PollenMeddelande och BallongMeddelande. Diagrammet bör förutom ärvningen också visa associationerna mellan klasserna, samt associationernas kardinalitet (= hur många objekt av klassen som hör till de olika "ändorna" av kopplingen).

Använd en abstrakt basklass för de olika meddelandetyperna. Förklara varför detta är vettigt.

Räkna upp BallongMeddelande-klassens fält och fältens typ. Mätningar gjorda på olika höjder bör sparas i en tabell som indexeras enligt höjden.