

T-106.1207 Grundkurs i Programmering

Tent 9.5.2007

Skriv i övre kanten på varje svarpapper kursens namn och kod, tentdatum, studentens 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. `public`
2. `boolean`
3. Källkod
4. Abstrakt klass
5. Gränssnittsklass
6. Överlagring av metoder (method overloading)
7. Klassvariabel (och hur den skiljer sig från instansvariabel)
8. Event-driven programmering (och hur den skiljer sig från batch-orienterad programmering)

Uppgift 2 (9 poäng)

Beskriv kort de olika sätten att få programmet att bete sig olika på basen av olika villkor.

Vilka olika uttryck finns det för att göra val? När lönar det sig att använda ett uttryck framom ett annat?

Illustrera hur de olika uttrycken fungerar genom att skriva följande exempelmetod (på de olika sätten):

Metoden tar som en parameter ett heltal som motsvarar en veckodag (0 = måndag, 1 = tisdag, 2 = onsdag, 3 = torsdag, 4 = fredag, 5 = lördag, 6 = söndag). Metoden returnerar en teckensträng som berättar butikernas öppethållningstider (måndag-fredag = "9-21", lördag = "9-18", söndag = "12-21").

Uppgift 3 (19 poäng)

I denna uppgift har givits ett program för att hålla reda på resultaten i olika matcher (t.ex. fotboll).

```
class Match {
    String homeTeam;
    String awayTeam;
    int homeGoals;
    int awayGoals;
    boolean isFinished;

    public Match(String homeTeam, String awayTeam,
                 int homeGoals, int awayGoals,
                 boolean isFinished) {
        this.homeTeam = homeTeam;
        this.awayTeam = awayTeam;
        this.homeGoals = homeGoals;
        this.awayGoals = awayGoals;
        this.isFinished = isFinished;
    }

    public String toString() {
        if (isFinished) {
            return homeTeam + " - " + awayTeam + "\t" +
                homeGoals + " - " + awayGoals;
        } else {
            return homeTeam + " - " + awayTeam;
        }
    }
}

class MatchAlreadyExistsException extends Exception {
    public MatchAlreadyExistsException(String m) {
        super(m);
    }
}
```

```
public class MatchLog {

    public int numberOfMatches;
    private Match matches[];

    public void addMatch(int matchNumber, Match m) {
        matches[matchNumber - 1] = m;
    }

    public int countWins(String teamName) {
        Match m;
        int nWins = 0;
        for (int i = 0; i < numberOfMatches; i++) {
            m = matches[i];
            if ((m != null) && m.isFinished)
                if ((m.homeTeam.equals(teamName) &&
                    (m.homeGoals > m.awayGoals)) ||
                    (m.awayTeam.equals(teamName) &&
                    (m.awayGoals > m.homeGoals)))
                    nWins++;
        }
        return nWins;
    }

    public String toString() {
        String s = "";
        for (int i = 0; i < numberOfMatches; i++)
            if (matches[i] != null)
                s += matches[i].toString() + "\n";
        return s;
    }

    public static void main(String[] args) {
        MatchLog log = new MatchLog(26);
        Match m = new Match("EIF", "VG-62", 0, 1, true);
        log.addMatch(1, m);
        m = new Match("GrIFK", "EIF", 2, 1, true);
        log.addMatch(2, m);
        m = new Match("GrIFK", "EIF", 1, 4, true);
        log.addMatch(2, m);
        System.out.print(log);
    }
}
```

Svara på följande frågor:

1. Programmet kommer att krascha. Berätta varför och förklara hur man kan fixa problemet.
2. Om vi antar att felet har korrigerats, vad kommer main-metoden att skriva ut då den körs?
3. Hur skulle du ändra på metoden som räknar antalet vinster så att den istället räknar antalet förluster?
4. Skriv en metod för att räkna ut hur många poäng ett givet lag har samlat (antag att man för en vinst får 3 poäng, för oavgjort 1 poäng, samt för förlust 0 poäng). Rekommenderad metod-signatur: `public int getPointsForTeam(String teamName)` (Det är tillåtet att skriva en hjälpmetod för att räkna antalet oavgjorda matcher ett lag spelat, även om det innebär att man "i onödan" går igenom tabellen två gånger.)
5. Skriv en metod som berättar hur lång den längsta obesegrade sviten för ett givet lag är. (Antag att matcherna är ordnade i kronologisk ordning. Rekommenderad metod-signatur: `public int longestUnDefeatedStreak(String teamName)`)
6. Ändra på `addMatch`-metoden så att den kastar ett `MatchAlreadyExistsException` om det redan finns en match med det givna numret. Ändra också på `main`-metoden så att den fångar undantaget och skriver ut "Matchen existerar redan."

Första frågan ger 3 poäng. Andra och tredje frågan ger två poäng. Övriga frågor ger fyra poäng.

Uppgift 4 (8 poäng)

Ett förenklat program för att hålla reda på filmer i en videouthyrningsfirma kunde eventuellt se ut på följande sätt:

Det finns olika filmer, vilka har en titel, ett utgivningsår, och en genre (t.ex. drama, romantisk komedi, skräckfilm). Av dessa filmer kan det sedan finnas ett eller flera uthyrningsexemplar (programmet håller inte reda på filmer som inte finns att hyra).

Dessa exemplar kan vara antingen VHS-kassetter eller DVD-skivor. För varje exemplar håller vi reda på huruvida det är uthyrt, och i så fall till vilken kund och när den skall lämnas tillbaka.

Det finns två olika typer av kunder: vanlig kund och stamkund. För varje kund har vi ett namn och någon form av kontaktuppgifter (t.ex. adress, telefonnummer).

För stamkunder håller vi dessutom reda på hur många filmer de hyrt, samt hur många bonuspoäng de har till godo. En kund kan hyra ett eller flera uthyrningsexemplar, men ett uthyrningsexemplar kan bara vara uthyrt åt en kund i gången.

Rita ett UML-diagram som beskriver ovan nämnda program.

Diagrammet skall innehålla klasshierarkierna för uthyrningsexemplaren och kunderna. Det skall också innehålla alla associationer mellan de olika klasserna. Associationernas kardinalitet (hur många av vardera klassen som ingår i associationen) skall också synas i diagrammet.

Skriv också ut klassernas attribut (åtminstone de som nämns i beskrivningen) samt deras typer.

Svara också på följande fråga:

Varför gör vi skillnad på filmer och uthyrningsexemplar?

