

Tentamen i TDP004

Objektorienterad Programmering

Praktisk del

- Datum: 2009-04-15
- Tid: 8-12
- Plats: SU-salar i B-huset.
- Jour: Per-Magnus Olsson, tel 285607
Jourhavande kommer att besöka skrivsalarna ungefär varje timme under skrivtiden.
- Hjälpmedel: Teoretisk del: Inga.
Praktisk del: Den C++ information som finns i systemet.
- Betygsättning: Max antal poäng: 46 med 23 poäng vardera på teori och praktikdel.
- | Poäng | Betyg |
|-------|----------|
| 39-46 | 5 |
| 31-38 | 4 |
| 24-30 | 3 |
| 0-23 | U |
- Anvisningar: Börja med den teoretiska delen. När du är klar med den lämnar du in den och får den praktiska delen. När du har lämnat in den teoretiska delen kan du inte återvända till den.
Skriv svaret på varje uppgift på ett separat blad.
Uppgifterna är inte ordnade i svårighetsgrad.

Lycka till!

TDP004 Objektorienterad Programmering

Praktisk del

Den här delen av tentamen bygger på uppgift 1 och 2, vilka bör lösas i ordning. Uppgift 3 och 4 kan göras i valfri ordning. Om du inte klarar av uppgift 2 innan du gör uppgift 3 eller 4, skapa själv lämplig indata att testa din implementation med.

1. I mappen givenfiles finns filen medaljer.txt. På varje rad i filen finns en landsförkortning på tre bokstäver samt tre stycken tal. Det första talet är antalet guldmedaljer, det andra talet är antalet silvermedaljer och det tredje är antalet bronsmedaljer i ett stort mästerskap. Uppgiften går ut på att välja ett lämpligt sätt att modellera informationen. Ditt svar ska innehålla koden samt en kort motivering/diskussion i filen (max ca en sida) som behandlar t.ex. följande:
 - Hur ska nya instanser skapas?
 - Hur ska medlemsvariabler modifieras?
 - Hur hanteras åtkomst av medlemsvariabler? Finns det felkontroll av medlemsdatan?
 - Är ditt val förberett för att återanvändas genom arv?Poäng ges för vettiga val och motiveringar till dessa. (10p)
2. Läs in information från filen medaljer.txt och lagra informationen i instanser av din datastruktur från uppgift 1. Om öppningen av filen misslyckas ska det meddelas till användaren, annars antas det att all information i filen är korrekt inläst. Lagra instanserna på lämpligt sätt. (5p)
3. Gör en funktion `find_countries` som tar inparametrar av lämpliga typer och returnerar en STL container (av valfri och passande typ) innehållande de länder som uppfyller kraven att antalet medaljer av olika valörer har de värdena som inparametrarna anger. Exempel: Om funktionen får inparametrarna 1,2,3 så ska returvärdet innehålla de instanser som representerar de länder som har tagit 1 guldmedalj, 2 silvermedaljer och 3 bronsmedaljer. Om det inte finns något land som uppfyller kraven ska en tom container returneras. Välj lämpliga inparametrar och returtyp till funktionen samt implementera funktionen. Motivera dina val kortfattat i filen. (3p)

4. Ta informationen från uppgift 2, sortera den så att ett land med fler medaljer av en viss typ rankas än ett land med färre medaljer. Om antalet medaljer av en viss typ är lika mellan två länder, använd medaljtypen närmast under för att avgöra ordning. Guld rankas högst, silver näst högst och brons lägst. Någon av de inbyggda **sort**-funktionerna i **STL** ska användas, d.v.s. du ska *inte* implementera någon egen sorteringsalgoritm. Efter att informationen är sorterad, skriv ut den på skärmen med först landsförkortningen, sedan antalet guldmedaljer, antalet silvermedaljer och sist på raden antalet bronsmedaljer. Det land med högst ranking ska stå först och ha siffran 1 framför sig, näst högst ranking skrivs ut därefter med siffran 2 före sig och så vidare. Se exempel nedan. (5p)

Exempel på indata:

USA 0 1 0

Nor 0 1 1

Swe 2 1 0

Utskrift efter sortering:

1 Swe 2 1 0

2 Nor 0 1 1

3 USA 0 1 0