

Tentamen i TDP004

Objektorienterad Programmering

Praktisk del

- Datum: 2008-12-16
- Tid: 14-18
- 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

1. I filen `shoesize.cpp` finns funktionen `Calculate_shoe_size()` som begär personers skostorlek som indata och beräknar sedan den genomsnittliga skostorleken som sedan skrivs ut på skärmen. Men det finns ett **allvarligt** fel i programmet. Du har två uppgifter.

1a) Hitta felet och beskriv ett testfall som visar när felet uppkommer. Förklara varför felet uppkommer. Testfallet behöver inte vara automatiskt och kan t.ex. bestå av den indata som orsakar felet, i den ordning som gör att felet uppkommer, tillsammans med det resultat du förväntar dig om koden är korrekt samt det resultat som den felaktiga koden ger.

1b) Rätta felet i koden och kontrollera att ditt testfall från A nu ger rätt resultat.

Lämna in testfallet och den rättade koden. (3+3p)

2. Skapa en klass `Zoo` som sparar pekare till olika `Animal` i någon container från standardbiblioteket. `Animal` är basklassen i en hierarki av djur. `Animal` har subclasserna `Cat` och `Dog`. Klassen `Animals` konstruktor är `Animal(const std::string name)`. `Animal` har en pekare till en `std::string` som sparar djurets namn, och har även en rent virtuell funktion `void Make_sound()` som skriver ut djurets namn " says " samt det ljud som den typen av djur gör ("meow!" respektive "wow!") på skärmen. Exempel: om en katt heter `Catbert` skriver `Make_sound` ut "Catbert says meow!" Implementera dessa klasser och funktioner. För att visa att det du har gjort fungerar, skapa några instanser av olika sorters djur i `Zoo` och iterera sedan igenom elementen i containern och anropa `Make_sound()` för alla instanserna. (8p)

3. En möbelfirma ska datorisera sin lagerhantering och du ska göra en del av systemet. Data om de olika möblerna sparas i objekt av typen `Furniture`. Din uppgift är att implementera `save`-funktionen i `Furniture`-klassen. Denna använder medlemsvariabler vars data är parametrar till konstruktorn, dessa sparas i medlemsvariabler av samma typ som konstruktorns inparametrar. Signaturen för klassen `Furniture`s konstruktor ser ut enligt:

```
Furniture(const std::string type,  
          const std::string name,  
          std::list<std::string>& colors);
```

Funktionen `bool Save() const` som du ska implementera ska skriva all information i en fil med namnet `type_name.txt`. Om öppnandet av filen misslyckas ska funktionen returnera `false`, annars ska den returnera `true` efter att informationen har skrivits. (9p)

Exempel:

```
std::string type = "Sofa";  
std::string name = "Ryd";  
std::string color1 = "white";  
std::string color2 = "red";  
std::string color3 = "black";
```

```
std::list<const std::string> colors;  
colors.push_back(color1);  
colors.push_back(color2);  
colors.push_back(color3);
```

```
Furniture f = Furniture(type, name, colors);
```

När funktionen **Save** anropas skrivs följande i filen `Sofa_Ryd.txt`

Type: Sofa

Name: Ryd

Colors: white, red, black

Observera att det inte ska vara något kommatecken efter den sista färgen.