

Instuderingsfrågor

Föreläsning 3

Frågor kopplade till föreläsning 3. För att hitta svaren behöver du olika strategier:

- Titta igenom föreläsningbilder
- Leta på kurshemsidan
- Skriv kod och prova kompilera och köra
- Förkunskaper alt. tänka till om vad som helt enkelt verkar vara en förnuftig strategi
- Sök informationen på nätet
- Läs igenom hela dokumentet innan du börjar

Klasser och objektorienterat tänkande

1. Vad skiljer en klass från ett aggregat?
2. Vad skiljer en imperativt tankesätt från objektorienterat?
3. Går det att tänka objektorienterat i ett språk (t.ex. "C") som bara stödjer imperativ programmering?
4. Går det att tänka imperativt i ett språk (t.ex. "C++") som är byggt för objektorienterad programmering?
5. Vilka problem försöker vi lösa med objektorienterad programmering (e.g. objektorienterat tankesätt)?
6. Du har konverterat `length(Vector const& v)` till `Vector::length() const` och ändrat motsvarande funktionsanrop från `length(x)` till `x.length()`. Hur kommer du åt parametern "x" i det andra anropet?
7. Vilka delar av en klass ska vara "private" och vilka ska vara "public"?
8. Varför ska "private" användas i så stor utsträckning som möjligt?
9. Varför ska "const" användas i så stor utsträckning som möjligt?
10. Vad betyder det att en medlemsfunktion markeras som "const"?
11. Extern kod råkar få fatt i var en privat datamedlem är lagrad i minnet (via referens eller adress). Kan den externa koden ändra datamedlemmen?
12. Medlemsfunktionen `complex::add(complex const& rhs)` som anropats på objektet "lhs" behöver komma åt de privata datamedlemmarna "a" och "b" i objektet "rhs". Blir det kompileringfel?
13. En klassdefinition deklarerar den privata datamedlemmen "name". Hur ger du den ett värde?
14. Du deklarerar två nya objekt (variabler) "a" och "b" av klassen (typen) "X". Hur ändrar du datamedlemmen "m" i a (utan att röra "m" i "b"). Vi förutsätter att "m" är publik.

15. Besvara föregående fråga igen, men utgå nu från att “m” korrekt gjorts privat.
16. Kompilatorn skriver felmeddelandet “error: passing ‘const Foo’ as ‘this’ argument discards qualifiers [-fpermissive]” angående en rad som anropar medlemsfunktionen “bar()” på objektet “fum”. Hur ser deklarationen av “fum” ut, hur ser anropet ut och hur är medlemsfunktionen “bar()” deklarerad?
17. Varför är anropet i ovan fråga ett problem?
18. Vad behöver ändras för att lösa problemet i ovan två frågor. Utgå från att “bar()” inte har behov att ändra på sitt objekt.
19. Du har konverterat från “struct Vector” där du använde aggregatinitiering `Vector obj{1,2}` till `class Vector` med privata datamedlemmar (bra där!). Nu får du felet `no matching function for`. Vad saknas?
20. När används en datamedlemsinitieringslista?
21. En klass har en “const” datamedlem. Kan den initieras (ändras) i konstruktorns funktionsblock (inom konstruktorns)?
22. En klass har en referens som datamedlem. Kan du i konstruktorns funktionsblock (inom konstruktorns klammerparenteser { }) ändra vilken variabel referensen refererar till (är alias för)?
23. Din klass har datamedlemmarna x, y och z i den ordningen. Vad får du för fel när din datamedlemsinitieringslista är skriven som `: z{5}, x{z}`?

Abstraktion

1. Du har gjort klassen “Skiplist” för att representera en effektivt sökbar lista med tal. Just nu implementerar du medlemsfunktionen “find” med hjälp av en “while”-loop. Identifiera så många abstraktioner du kan i scenariot.
2. Tänk igenom din dag. Hur många abstraktioner har du interagerat med sedan du vaknade?
3. Varför abstraktion användbart och viktigt?
4. Programspråket erbjuder funktionsöverlagring, men inte default-argument. Hur skapar du ändå ett default-argument?

Filuppdelning, filer och kompilering

1. Vad tjänar du på att lägga deklarationer och klassdefinition i en inkluderingfil (*.h) och motsvarande definitioner i en implementationsfil (*.cc)?
2. Du implementerar klassen `Rock_Band` och har i din implementationsfil skrivit medlemsfunktionen `int followers() { return 0; }` när du får kompileringsfelet “undefined reference to” rörande just “followers”. Du har redan kontrollerat att du sparat filen, och att du verkligen kompilerar rätt implementationsfil. Vad har du glömt?
3. Varför står det `#ifndef` och `#define` överst i inkluderingsfiler, följt av `#endif` i slutet?
4. Varför ska du inte skriva “using namespace std” i en inkluderingsfil?
5. Vilken typ av filer ska du ta bort från din arbetamapp?

6. Vilken typ av filer ska du kompilera och vilken typ ska du inkludera?
7. Vilken typ av filer ska du aldrig ta med i din versionshantering (git) och är säkra att städa bort?
8. I ditt spelprojekt får du kompileringsfelet “redefinition of class Boss” när du kompilerar en ny fil som inkluderar “boss.h” och “enemies.h”. All tidigare kod, inklusive klassen “Boss” och kod som använder den kompilerar fortfarande, det är bara när du tar med den nya filen problemet uppstår. Vad har du troligen för problem i “boss.h” och “enemies.h”?
9. Varför finns det flera varianter på inkluderingsfiler (*.h, *.hh, *.hpp) och implementationsfiler (*.cc, *.cpp)?
10. Vilka tre faser ingår i det vi slentrianmässigt bara kallar “kompilering”?
11. Vad är det för skillnad på `#include <...>` och `#include "..."`?
12. Vad betyder kompileringsflaggan “-c”?

Operatoröverlagring

1. Du har deklarerat objekten “Vector v1, v2;”. Nu vill du göra “v1 + v2”,
2. Vad är fördelen med en fristående operatorfunktion?
3. Vad är fördelen med en operatorfunktion som klassmedlem?
4. Du har bara möjlighet att göra en fristående operatorfunktion, men får kompileringsfelet “is private within this context”. Hur går du tillväga för att lösa problemet?
5. Du får felet “bool X::operator<(X, X) must have exactly one argument”. Vad har du gjort för misstag?
6. Du vill överlagra utmatningsoperatören <<. Vad ska det stå för datatyp till vänster (datatypen för strömmen)?
7. Varför måste utmatningsoperatören returnera en referens till strömmen?
8. Vilket syntax används för att returnera en referens till hela objektet “x” från funktionen “fun()” i anropet “x.fun()”?
9. Är det en bra idé att jag implementerar operatören “++” för min nya flyttalstyp?
10. Jag har implementerat alla jämförelseoperatorer till min klass och varit så pass smart att jag anropar < för att lösa >, jag anropar > och == för att lösa >= och jag anropar >= för att lösa <. Nu får jag “segmentation fault” när jag testar programmet. Vad är problemet?

Undantag

1. Du och två kollegor slår vad om vem som kan lösa var sitt problem snabbast. Ni drar sticka om vem som får vilket problem. Det är ett mystiskt kompileringsfel, det är en programkrasch till följd av ett mystiskt undantag och det är en programkörning som ger mystisk (fel) utdata i terminalen. vilket problem hoppas du på för att vinna vadet? Vilket problem väljer du in andra hand?

2. Utgående från rätt svar på förra frågan och utgående från att olika implementationsstrategi leder tenderar att ge respektive typ av fel oftare, vilken implementationsstrategi ska du välja i första hand?
3. Du upptäcker i konstruktorn till din klass "Weekday" att du fått in veckodagen "Mandog". Vilket alternativ är bäst?
 - (a) skriv ut ett felmeddelande och returnera från konstruktorn
 - (b) kasta ett undantag eftersom det är en felaktig veckodag
 - (c) ändra till "Måndag" eftersom det antagligen är vad användaren menade
 - (d) ändra till "Måndag" som standard vid fel angiven dag
 - (e) kör på men sätt objektets veckodag till "Day error"
 - (f) kör på och sätt objektets veckodag "Mandog"
 - (g) skapa en uppräkningsstyp som bara kan anta värdena "MON, TUE, WED, THU, FRI, SAT, SUN"?
4. Måste du fånga undantag som referenser?
5. Vilken sats används för att kasta undantag?
6. Vilken sats används för att detektera kastade undantag?
7. Vilken sats används för att välja vilken typ av undantag som ska fångas?
8. Vad händer med programkörningen om ett undantag inte fångas?
9. Min funktion `add_result` söker ut en `Student` ur ett set med registrerade när jag får undantaget `std::out_of_range`. I koden som anropar `add_result` vet jag att problemet uppstår när en student saknar kursregistrering. Vad bör jag göra?
 - (a) fånga (kväva) felet redan i `add_result` så jag slipper få det senare
 - (b) strunta i undantaget helt och hållet, felet upptäcks när main kraschar med `terminate called after`
 - (c) fånga undantaget i main och skriva ut ett mer begripligt felmeddelande?
 - (d) fånga undantaget i anropande kod och istället kasta ett `Student_Not_Registered`
 - (e) fånga undantaget i kod där jag har möjlighet att lägga till den felande studenten på en undantagslista med resultat som inte gick igenom
 - (f) valfri kombination av ovan strategier
10. I ovan fråga har du identifierat ett antal dåliga strategier. Varför är de dåliga?

Testdriven utveckling och `catch.hpp`

1. Varför är det viktigt att se nyskrivna tester misslyckas innan de går igenom?
2. Vad betyder det att "refaktorera" sin kod?
3. Vad får vi ut av att skriva testfall till kod vi ännu ej skrivit?
4. Hur kan vi få testfallen att kompilera innan vi skrivit all kod de ska testa?
5. Hur bekräftar vi med Catch att en jämförelse ger "sant" resultat?
6. Hur bekräftar vi med Catch att en jämförelse ger "falskt" resultat?
7. Hur bekräftar vi med Catch att ett felaktigt anrop verkligen kastar ett undantag?

8. Varför måste vi kontrollera *både* att en funktion ger rätt resultat med rätt indata *och* fel resultat med fel indata?
9. Jag vill kontrollera om en funktion ger “sant” för värden mindre än 12 och “falskt” för värden 12 eller större. Vilka 3-5 testfall är prioriterade?
10. Varför placeras Catch huvudprogram i en egen fil `test_main.cc`?

Svar

Svar ännu ej skrivna. Peta på examinerator!