

Instuderingsfrågor

Klas Arvidsson

2021-03-10

1 Koncept

1.1 Klasser

- På vilka sätt skiljer sig en medlemsfunktion från en fri(vanlig) funktion?
- Vad är likheterna mellan en medlemsfunktion och en fri(vanlig) funktion?
- På vilka sätt skiljer sig en datamedlem från en vanlig variabel?
- Vilka likheter finns mellan en datamedlem och en vanlig variabel?
- På vilka sätt skiljer sig en konstruktor från en medlemsfunktion?
- Vilka likheter finns mellan en konstruktor och en medlemsfunktion?
- Vilken uppgift har en konstruktor (tekniskt)?
- Vilka 6 medlemsfunktioner räknar vi till "speciella medlemsfunktioner" som automatiskt används i olika skeden av ett objekts livstid?
- Vilken uppgift har en destruktör (tekniskt)?
- Vad är det för skillnad på klass och instans?
- Vad är slicing?
- Vad betyder "pure virtual"?
- Vad är dynamisk bindning?
- Vad är statisk bindning?

1.2 Minneshantering

- Hur hanteras variabler på programmets stack?
- Hur hanteras variabler på programmets heap?
- Vad är dynamiskt minne?
- Vad är en anonym variabel?
- Vad är en pekare (pekarvariabel)?
- Vad är en djup kopia?
- Vad är en minnesläcka?

1.3 Objektorientering

- Vad är ett aggregat? (koncept, i kod)
- Vad innebär aggregation? (koncept, i kod, UML)
- Vad innebär komposition? (koncept, i kod, UML)
- Vad innebär arv(specialisering)? (koncept, i kod, UML)
- Vad innebär association? (koncept, i kod, UML)
- Vad innebär inkapsling?
- Vad är abstraktion?
- Finns det kopplingar mellan abstraktion och inkapsling?
- Är det skillnad på instans och objekt?
- Vilka fördelar ger en väl utförd inkapsling?
- Vad behövs utöver skydd med public, protected, private för att få god inkapsling?

2 Praktik

2.1 Programmering

- Varför är det viktigare att förstå problemet än att få koden att fungera?
- Hur använder du klasser för att kapsla in och automatisera en del av minneshanteringen?
- Hur gör du för att undvika att objekt kopieras i onödan?
- Vilka kompileringsflaggor/varningar ska du slå på när du kompilerar?
- Med en lång lista kompileringsfel i terminalen, vilket ska du åtgärda först?
- Hur ofta ska du kompilera för att inte få en lång lista kompileringsfel?
- Varför är det viktigt att åtgärda alla varningar kompilatorn ger?
- Vilket program kan användas för att detektera minnesläckor?
- I vilka situationer behövs anonyma variabler?
- Behövs några särskilda tester för att garanterat hitta alla minnesläckor?
- Vad har du gjort för fel när det blir "undefined reference to XYZ" i slutet av kompileringen?
- Var ska du leta efter problem när kompilatorn pratar om "vtable"?

2.2 Klasser

- Hur ritas du upp en klass (som "lådorna" ser ut i minnet)?
- Hur ritas du upp en klass (som UML klassdiagram)?
- Hur avbryter du en konstruktor utan att det skapa ett objekt?
- Vilka speciella medlemsfunktioner skapas automatiskt om du inte gör det själv?
- I vilken situation kommer de automatiskt skapade speciella medlemsfunktionerna vara otillräckliga?
- Hur blir en klass "abstrakt"?
- Vilken begränsning finns med en abstrakt klass?
- Hur uppstår slicing?
- Hur skapas(allokeras) en anonym variabel?
- Hur tas en anonym variabel bort(avallokeras)?
- När anropas destruktorer?
- Hur uppstår en minnesläcka?
- Hur ska `dynamic_cast` användas?
- När blir det ett problem att en "virtual" destruktor saknas?
- I vilka situationer behöver "this" användas?
- Vilka fördelar kan en konstruktor ge den som skriver en klass?
- Vilka fördelar kan en konstruktor ge den som använder en klass?
- Vilka fördelar kan en destruktor ge den som skriver en klass?
- Vilka fördelar kan en destruktor ge den som använder en klass?
- Vilka fördelar kan "private" ge den som skriver en klass?
- Vilka fördelar kan "private" ge den som använder en klass?
- När är det lämpligt att ha "const" efter en medlemsfunktion?
- Varför är det viktigt att använda "const" efter en medlemsfunktion när det är lämpligt?
- När är det rimligt att göra en klass abstrakt?

2.3 Syntax

- Vad betyder "virtual" framför en medlemsfunktion?
- Vad betyder " = 0" efter en medlemsfunktion?
- Vad betyder " = delete" efter en medlemsfunktion?
- Vad betyder " = default" efter en medlemsfunktion?
- Vad betyder "override" efter en medlemsfunktion?
- Kan "override" användas efter vilken medlemsfunktion som helst?
- Kan "virtual" användas framför vilken medlemsfunktion som helst?
- Vad betyder "const" efter en medlemsfunktion?
- Vad gör avrefereringsoperatorn(*)?
- Vad gör piloperatorn(->)?
- Vad betyder public, protected, private? (tekniskt)
- Vad betyder "this"?
- Hur känner du igen en konstruktor i koden?
- Hur känner du igen en destruktör i koden?
- Vad betyder "&&" i samband med en parameterdeklaration?
- Vad är en lämplig ordning för att veta var anonyma variabler ska avallokeras?