

dt047g Programmeringsmetodik

Laboration - Arv

Martin Kjellqvist*

inheritance.tex 500 2023-08-10 09:57:47Z martin

1 Introduktion

Arv.

Det här är en omskrivning av en labb ursprungligen författad av Alexander Kozynchenko.

https://en.cppreference.com/w/cpp/language/derived_class

Läs igenom texten på cppreference.

Lägg märke till hur de flera gånger pekar på svårigheter med arv. Bland annat

- Virtuella basklasser, alla basklasser måste ange att de ingår i ett virtuellt arv.
- Subklasser med virtuella basklasser har ansvaret att skapa de virtuella basklasserna om de är ett löv i klass-hierarkin.
- *References and pointers to a derived object are expected to be usable by any code that expects references or pointers to any of its public bases (see LSP) or, in DbC terms, a derived class should maintain class invariants of its public bases, should not strengthen any precondition or weaken any postcondition of a member function it overrides.*
- Exemplet med `ColorMenu` ser vid en första anblick rimlig ut men blir snabbt en komplicerad röra på vad som borde vara ett enkelt problem.

*E-post: martin.kjellqvist@miun.se.

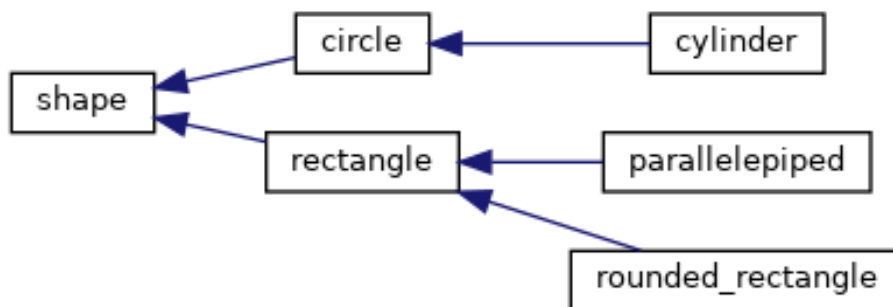
- Protected inheritance är väldigt ovanligt.
- Private inheritance blir ofta krångligare än vad man tänkt sig. Namnkollisioner hopar sig snabbt.

Labben består av två delar.

- Skapa en klasshierarki med Shape som basklass. I arvshierarkin ingår Shape, Circle, Rectangle, Square, Parallelepiped, Cylinder.
- Skapa en main.cpp som testar och undersöker beteendet hos objekt du skapar.

2 Hierarki

Skapa en hierarki som ser ut som:



Figur 1: Arvsdiagram. shape är bas.

De publika medlemmarna i shape ska vara.

```

class shape{
public:
    shape(std::string color);
    virtual double get_area() = 0;
    std::string get_color();
}
  
```

Glöm inte klassen `square`, stoppa in den i arvshierarkin där du tycker det är lämpligt.

Du har alltså till slut 7 klasser som ingår i arvshierarkin. Det naturliga är att använda din IDE och välja *New class* alternativet i din miljö så det inte blir en så fasligt massa klickande och tryckande.

3 Testning och utvärdering

I `main.cpp` skriver du

```
int main(){
// Här skapar du minst ett objekt av varje typ i
// hierarkin.
// Placera dem i en lämplig container.

// Anropa en funktion get_data(container)
}
```

och funktionen `get_data`

```
void get_data(container c){
// iterera över samtliga objekt i containern.
// För varje objekt:
// - Skriv ut objektets färg och skriv ut arean.
// Till sist skriver du ut den totala arean av
// samtliga objekt.
}
```

4 Examination

Det normala är att redovisar i labbsal. Se sidan tips labb 2 för vad som förväntas under denna redovisning.

Om du inte har möjlighet att redovisa kommunicerar du detta med din handledare så löser vi det.