

LÖSNINGSFÖRSLAG – TENTA 1999-05-31

Skissat av Kalle Prorok (har ej provkört programexemplen)

- STL är organiserad som en tredimensionell rymd med algoritmer, datastrukturer och datatyper som de tre axlarna. Tack vare templates behövs bara (i princip) en version av varje algoritm resp datastruktur. En iterator fungerar som ett slags pekare med dereferering (*) och inkrementoperator(++). Det finns lite olika varianter på iteratorer; framåt, bakåt, random. läs, skriv.
b) `vector`, `list`, `deque`, `set`, `multiset`, `map`, `multimap`, `stack`, `queue`, `priority_queue` (2 rækker)
- `dynamic_cast` returnerar 0 om en konvertering inte resulterar i (är ett) ett önskat objekt. Används t.ex. då man vill konvertera en basklasspekare till en subclasspekare (för att nå dess medlemmar) då objektet faktiskt är en instans av subclassen. En typsäkrare konvertering.
 - I C/C++ skiljer man inte på pekare till enskilda element eller vektorer av element så därför måste man som programmerare "hjälpa till". Ett alternativ kunde varit att lagra antalsinformation i början av det allokerade minnet men det skulle göra att C++ program tog mer minne än nödvändigt (kanske mer än dubbelt så mycket för enkla element).
 - Genom att lägga till parametrarnas datatyper till funktionsnamnet så får funktionerna olika namn "bakom kulisserna" dvs för kompilatorn och länkaren. Dessa suffix måste skalas av vid kombination med t.ex. C-program så kallad namnmangling. Egenskapen kallas överlagring (eller överladdning), även polymorfism godkänns. Dynamisk bindning är egentligen något annat och har med arv att göra.
- En abstrakt basklass är bra för att fånga gemensamma egenskaper (såsom färg, position etc) och fungerar även som en gränssnittsspecifikation (`draw()`, `move(int,int)` etc) för subclasser. Görs genom att införa en rent virtuell funktion:

```
virtual fkn() = 0;
```

eller genom att lägga konstruktorn `protected`.
 - Model, View, Controller används för att separera kod i ett program; i applikation, hur den presenteras för användaren och interaktionen med användaren.
- Några av skillnaderna kan vara; J saknar pekare, J har skräpsamlare (garbage collector) som gör att delete inte behöver anropas, J är mer portabelt, kan enkelt laddas ner över Internet, J är mer säkert, Enklare att lära. C++ är mer moget (t.ex i standard, utv.miljöer, antal kunniga mm), snabbare exekvering, mer vanligt vid kommersiell applikationsutveckling, har templates, operatoröverlagring mm. Multi-paradigm språk.
- `public`, `friend`, get-funktioner, `protected`(via arv),
 - Get gör att man skyddar detaljerna i implementationen som sedan lättare kan ändras.
 - Bryter mot inkapslingsprincipen.

6. Ett diagram (med symbolförklaring) så att vi förstår att du förstått vad som menas med aggregat, arv, association och abstrakt basklass samt en kortfattad beskrivning i ord eller (pseudo)kod om vad som händer godståget, ett slags dynamisk beskrivning av hur objekten samarbetar. Mer detaljerat svar kommer senare.

7. `class RangeErr { public: RangeErr(){} }; //Exception class`

```
template <class T> class V
{
    T *v;
    int size;
public:
    T& operator[](int i) { if (0 <= i && i < size)
                          return v[i];
                          else throw RangeErr();
                        }
}
```

Not. Det är OK att använda STLs `vector<>` klass (fast det är svårare ☺).

8. `class Tid`

```
{
    long s; // 16-bit int är bara 9 resp 18h(unsigned).
public:
    Tid(long s0 = 0) { s = s0; }
    Tid(int h, int m, int s2){s = 3600L * h + 60L * m + s2;}
    Tid operator+(const Tid& t) const
        { Tid t2; t2.s = s + t.s; return t2;}
    friend ostream& operator<<(ostream& os , const Tid& t);
};
```

```
ostream& operator<<(ostream& os , const Tid& t)
```

```
{
    long h = t.s / 3600L;
    long m = t.s / 60L - h * 60L;
    long s2 = t.s - m * 60L - h * 3600L;
    if (h<10) os << '0'; os << h << ':';
    if (m<10) os << '0'; os << m << ':';
    if (s2<10) os << '0'; os << s2;
    return os;
}
```

Not1. Alternativ (bättre!) lösning vore om klassen tillhandahåller `getH()`, `getM`, `getS()` så slapp utskriftsoperatören vara friend.

Not2. Wrap-around över 24 timmar behöver ej hanteras i denna uppgift.

Not3. `operator+` kan även implementeras som friend men det rekommenderas ej.

Not4. Nåt om talområdet bör finnas med.

Not5. Manipulatorer är ok men svårt. `printf("%02i",h)` är tveksamt.

<< NU ÄR DET SLUT PÅ SVAR <<