UMEÅ UNIVERSITET
Institutionen för datavetenskap

2000-12-16

Thomas Johansson
Kalle Prorok

Exam
Objektorienterad programmering för ingenjörer (TDBB09)
2000-12-19, kl 09.00 - 15.00

Where: Skrivsal 1 och 7, Östra Paviljongerna, Ålidbacken 23
Material allowed: none (except pen/pencil, eraser and a snack)
Course evaluation: Last among the papers is an evaluation form. Bring it home, fill it in and exchange it for your graded exam.

Start every task on a new sheet of paper.
Just use one side of the paper.
Write your name on every sheet.
Sort your solutions in the same order as they are given in the exam.
Note that the questions are not sorted by difficulty level, and remember that even partially solved tasks can reveice points.
Comment all source code so that we can understand its purpose.
Write clearly.

Maximum points is 40, civided among 8 questions.

Grade limits:	5	32p
	4	26p
	3	20p

The course web page will contain information about the on-going grading process. Ask if there is something that is not clear. We will visit the exam room at about **10.00** och **12.00**.

Good luck and Merry Christmas/Happy New Year !

Thomas Johansson(kursansv)
tel 090 - 12 71 16

Kalle Prorok
tel 070-3 33 35 37 (fram till ca 12.00 idag)

Question 1 – Comaprison with C (3p)

C++ has replaced a number of c function; eg.

char *s, strcpy	with	string
printf		<<
scanf		>>
int a[5]; a[i]		vector<int> a(5); a.at(i)
malloc		new

Give at least one cause of problem that you can avoid by using the newer methods (one cause for evevry one of the five cases!)

The string class simplifies the concept of string length (no /0 at the end), allocation, copying and concatenations is simplifyed.

<< is type safe and no matching to format arguments have to be done.

>> avoids the common mistake of forgetting the address-of operator (&) in scanf

<< >> can also be overloaded for user-defined data types.

vector handles size, range checking is easily implmented.

New knows the size of the object it creates, also handles errors due to memory shortage (by using exceptions).

Question 2 – Vektor class with templates and exceptions (8p)

Implement a type paramterized vector class that throws an exception when the index is out of range. use dynamic memory allocation. 'Only' contructor, destructor, assignment operator and indexing operator have to be implemented. Include a small test program. (If you can't remember how it is supposed to be done, solve it your own way and perhaps get a few points.)

```
#include <iostream>
using namespace std;
template <class T> class Vek {
  T *v;  // The allocated vektor
  unsigned int sz; // No of Allocated elements
public:
  Vek(unsigned int size=0) { v = new T[sz=size];}
  ~Vek() { delete [] v; }
  Vek& operator=(const Vek& v2) {
    if (this != &v2) { // Not myself
      delete [] v; v = new T[sz=v2.sz];
      for (int i=0; i<sz;i++) v[i]=v2[i];
    }
    return *this;
```

```
  }
  T& operator[](unsigned int i); // Separate for illustration
};
template <class T> T& Vek<T>::operator[](unsigned int i) {
  if (i<sz) return v[i];
  else throw runtime_error("RangeErr in Vek<T>");
}
//---Testprogram ------------

int main(int argc, char* argv[])
{
   try {
     Vek<double> vd(10);
     vd[4]=42; cout << vd[4] << endl;
     vd[3011]=-273.16; // Generates error
   }
   catch (exception& e){
      cerr << "** Error" << e.what();
   }
   cin.get();  return 0;
}
```

Question 3 – Algorithm Library (STL) (5p)

a) STL consists of four parts, which ? (1p)
b) Give five different container types! (1p)
c) Decribe how to use an iterator! (1p)
d) Is STL object oriented? (1p)
e) What restrictions are put on the elements (ortodox canonic class form)? (1p)

a) Containers, Algorithms, Funktion Objects and Adapters
b) I.e. vector, list, map, multiset, priority_queue (and deque, set, multimap, stack, queue)
c) Works like a pointer, useful in most container types
d) No! (optimized for speed, memory, etc)
e) They must implement: Default Construktor, Copy Constructor, Assignment Opertaor, Destructor, Equality. These are automatically implemented but should be overridden when dynamic memory is used. Set and Map also require Order (<).

Question 4 - Program Development (4p)

Try to give short answers to the following: What are the biggets problems with todays program development methods, why are many projects late and why are corrections at the end of a project so expensive ?

Delays, high costs, doesn't fulfill the requirements, bad structure, no reuse.

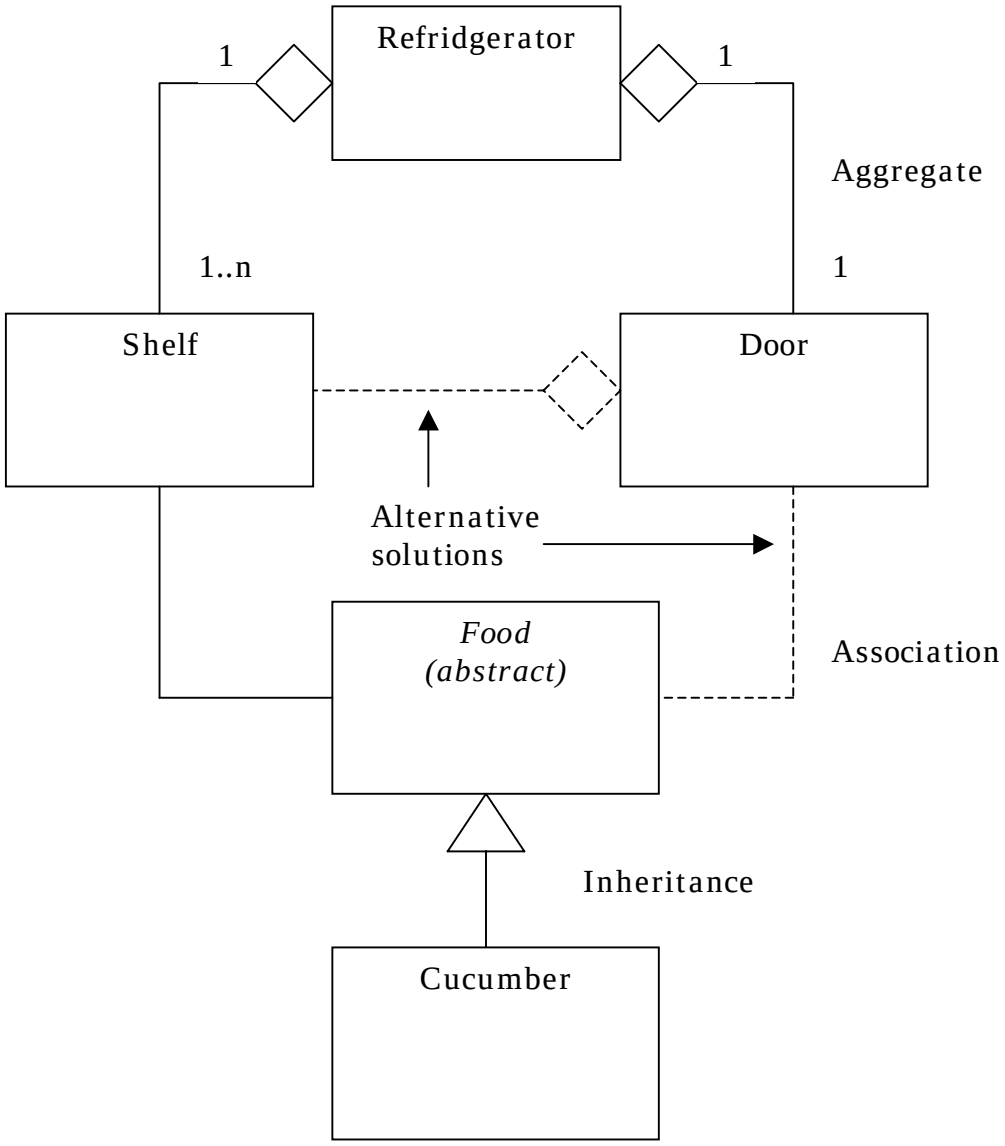
Complex systems, no overwiev, the Big Bang principle.

Because you have to thorw away tested code (it does the wrong thing). Perhaps costly debugging at many customer installations.

Question 5 – Analysis (4p)

When you get home after todays lecture on OOP you find that your ten-year old daughter has converted the refridgerator to a heat pump as an home assignment from school. Since your frozen dinner slowly melts in the shopping bag you decide to restore the fridge to its original state and then buy your daugther her own fridge to experiment with. While you sit on the living-room floor looking at the pieces you realize that this would be an excellent example to train object orientation on.

Draw a class diagram of your refridgerator with contents. Use the UML notation and explain the meaning of all symbols used. The diagram must contain at least one each of aggregate, association, inheritance and abstract base class. There has to be at least five different classes in your diagram. Note: you don't have to write any code !



Question 6 – OO (8p)

a) Name 4 advantages of using Object Orientation (4p)
b) Explain the meaning of *abstract data type* (1p)
c) Explain the meaning of *dynamic binding*(1p)
d) Explain what is meant by *abstract class*, what its purpose is and give one method for making a class abstract in C++ (1p)
e) Name a fundamental difference between object oriented program development and previous methods. (1p)

a) Exemples: Reuse -> better economy, easier to maintain -> better economy, modularisation -> easy to modify parts of the system, mpodel closer to the problem domain -> users can participate in the analysis phase, encapsulation -> less problems with 'global variables"
b) A data type that represents both data and operations that can be performed on the data
c) The metdod actually called at a method call (p->method() depends on the type of what is referred (what p points at) instead of the type of the reference (what p is declared as).
d) An abstract class can not be instanciated, i.e. no objects can be made from that class. Abstract classes are used to collect common behaviour and data from a collection of (sub) classes, and to specify interfaces (what functionality must be present in a sub class). A protected construktor or a pure virtual method is used to make a class abstract.
e) Earlier methods put emphasis on 'how' things were done, OO looks more to 'what' is present in the model (the model is much closer to the problem domain).

Question 7 – C++ (4p)

a) What does the keywords *private*, *protected* och *public* mean if they occur before an attribute in a class definition ? (1p)
b) What is the menaing of *overloaded operators* in C++ ? (1p)
c) What does *signature* mean in C++ ? (1p)
d) Write the first row of code in a class declaration for a class **Car** that inherits from the class **Vehicle** (1p)

a) Private – the attribute can only be reached from within the class, protected – methods in sub classes can reach it, public – everyone can reach it
b) The use of an operator can be extended to include your own data types (classes).
c) The name of a function together with the type, number and order of its arguments.

d) class Car : public Vehicle

UPPGIFT 8 - Java (4p)

a) Give one advantage and one disadvantage with using Java compared to C++ (2p)
b) Why can the following piece of program not be compiled by a Java compiler ? (1p)

```
unsigned int sum(int a, int b)
{
  return a + b;
}
```

c) Is there any way to overload operators like +, -) in Java ? (1p)

a) Exemples: Advantages – typ e safe, large libraries, no pointers that can point to the wrong place, garbage collection prevents memory leaks. Disadvantages – slower than C++. Some people think that templates should be part of Java.
b) There exists no unsigned int data type, all integers are signed.
c) No.

< SLUT PÅ UPPGIFTER >