

Apr 14, 04 9:37

subsets.cpp

Page 1/2

```
#include <iostream>
#include <string>
using namespace std;

#include "tvector.h"

class Subsets
{
public:
    Subsets();
    virtual ~Subsets(){}
    virtual void process(const tvector<string>& vec);
    virtual void generate() = 0;

protected:
    tvector<string> myElements;
};

Subsets::Subsets()
{
    myElements.push_back("apple");
    myElements.push_back("banana");
    myElements.push_back("cherry");
    /**
     myElements.push_back("date");
     myElements.push_back("eggplant");
     myElements.push_back("fig");
     */
}

void Subsets::process(const tvector<string>& vec)
{
    cout << vec.size() << " elts=";
    for(int k=0; k < vec.size(); k++) {
        cout << vec[k] << " ";
    }
    cout << endl;
}

class RecursiveSubsets : public Subsets
{
public:
    void generate(){
        tvector<string> subs;
        doGenerate(subs,0);
    }

private:
    void doGenerate(tvector<string>& subs, int index){
        if (index >= myElements.size()){
            process(subs);
        }
        else {
            doGenerate(subs, index+1);
            subs.push_back(myElements[index]);
            doGenerate(subs, index+1);
            subs.pop_back();
        }
    }
};

class BitSubsets : public Subsets
{
public:
}
```

Wednesday April 14, 2004

Apr 14, 04 9:37

subsets.cpp

Page 2/2

```
void generate(){
    int size = 1 << myElements.size();
    for(int k=0; k < size; k++){
        doProcess(k);
    }
}

private:

void doProcess(int mask){
    tvector<string> subs;
    int size = myElements.size();
    for(int k=0; k < size; k++){

        if ( (mask & 1) == 1){
            subs.push_back(myElements[k]);
        }
        mask = mask >> 1;
    }
    process(subs);
}

int main()
{
    Subsets * sub1 = new RecursiveSubsets();
    sub1->generate();
    cout << "\nbits\n" << endl;
    Subsets * sub2 = new BitSubsets();
    sub2->generate();
}
```

subsets.cpp

1/1