

1. Osztály, osztálysablon készítése

Implementáljon egy olyan `Array` osztálysablon, hogy a következő forráskód érvényes legyen:

```
Array<double> a1( 10 );
Array<double> a2=a1;
double d[]={1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0};
Array<double> a3( &d[2], &d[5]);
for( int i=0; i<10; ++i ) a1[ i ] = i;
```

2. Beágyazott iterátor készítése

Egészítse ki az előző feladatban készített forráskódot úgy, hogy a következő kód érvényes legyen:

```
Array<int> a( 10 );
Array<int> :: iterator it;
for(it = a.begin(); it != a.end(); ++it ) cout<< *it <<endl;
```

3. Nyilvános származtatás

Mit ír ki a következő program? Magyarázza az eredményt!

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

class A {
public:
    virtual void g() { cout << "A"; }
    void f() { g(); }
};

class B: public A {
public:
    void g() { cout << "B"; }
};

main() {
    A a; B b; A* pa = &a;
    b.f(); pa->f(); pa = &b; pa->f();
    return 0;
}
```

4. Privát származtatás

A `vector<>` osztálysablon felhasználásával készítsen egy verem osztályt.

- Használjon privát származtatást az osztály megvalósítására.
- Készítsen kódrészletet, amelyben a saját veremosztálya segítségével megfordít egy karakterláncot.

5. Operátorok túlterhelése

Készítsen egy komplex számok ábrázolására szolgáló osztályt csak a következő műveletekkel:

- implicit konstruktor
- inserter operátor
- extractor operátor

Készítsen kódrészletet, amely deklarál egy 12 elemű komplex számokból álló tömböt, beolvassa az elemeit az `in.txt` szövegállományból és kiírja a komplex számokat egy `out.txt` nevű szövegállományba.

1. C++ alapismeretek

```
void print( double * x, int n){
    for(int i=0; i<n; ++i )
        cout<<x[ i ]<<"\t";
    cout<<endl;
}
main(){
    double x[]={1, 2, 3};
    print(x, 3);
    max(x,3)=-1;
    print(x, 3);
    max(x,3)=-1;
    print(x, 3);
    return 0;
}
```

Implementálja a `max` függvényt úgy, hogy a megadott program az alábbi kimenetet állítsa elő!

Kimenet:

1 2 3

1 2 -1

1-1 -1

2. Osztálysablon készítése

```
template <class T>
class matrix
{
public:
    matrix( int m, int n );
    matrix( const matrix &other);
    ~matrix();
    matrix operator=( const matrix &other);

    int rows() const { return m; }
    int cols() const { return n; }
    T& at( int i, int j) throw(indexError);
    T at( int i, int j) const
        throw(indexError);

private:
    int m;
    int n;
    T *v;
};
```

- Implementálja a `matrix` osztály műveleteit!
- Miért van szükség a második `at` függvényre? Adjon egy példát, amely e nélkül nem működne!

3. Operátorok túlterhelése

Készítsen egy `inserter` és egy `extractor` műveletet, a `matrix` osztályhoz. Ezen műveletek segítségével olvasson be a standard bemenetről egy 3x4 méretű matrixot, majd írassa ki a matrixot egy szöveges állományba. Csak az operátorfüggvényeket használhatja az olvasásra, illetve az írásra!

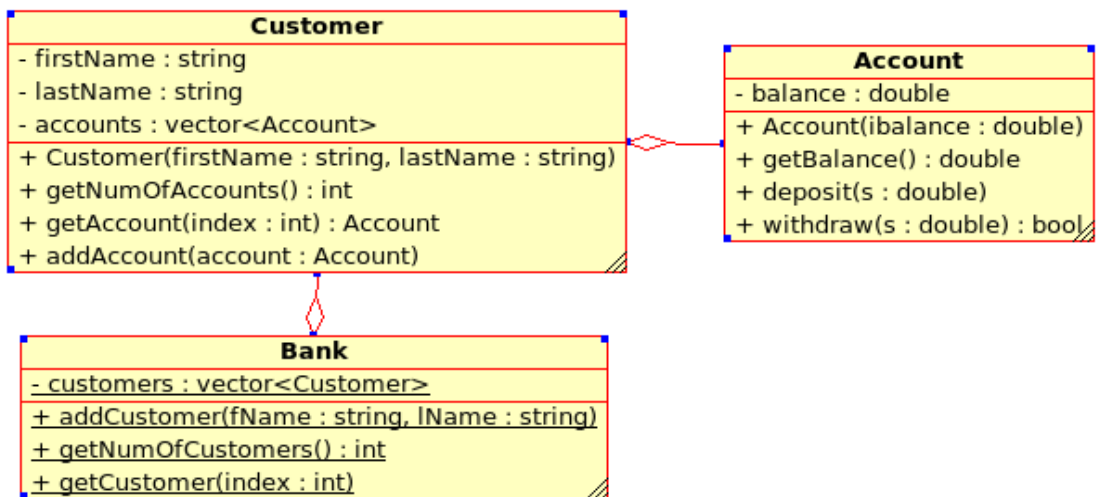
4. Nyilvános származtatás

Származtasson a `vector<T>` osztálysablonból egy `verem` (`stack`) osztálysablont. Az így kapott osztálysablont használja fel egy karakterlánc megfordítására. Milyen veszélyei vannak a nyilvános származtatásnak?

1. Osztályok, objektumok, statikus tagok **4 pont**

Implementáljuk az alábbi diagramon megadott osztályokat (az aláhúzás statikus tagokat jelent). Készítsünk egy main függvényt, amelyben:

- a bankhoz hozzáadunk két ügyfelet
- az első ügyfélnek egy számlát nyitunk és a másodiknak pedig kettőt.
- a számlák kezdőegyenlegét állítsa elő véletlenszerűen
- számítsa ki a bank összes ügyfeleihez tartozó számláinak összegyenlegét



2. Operátorok túlterhelése **2 pont**

Készítsen egy-egy inserter (kiíró) műveletet, az Account és a Customer osztályokhoz. Írassa ki a bank ügyfeleit, felhasználva az inserter operátort, egy előzőleg megnyitott szöveges állományba.

3. Nyilvános származtatás **2 pont**

Származtasson a vector<T> osztálysablonból egy verem (stack) osztálysablon. Az így kapott osztálysablon használja fel egy karakterlánc megfordítására. Milyen veszélyei vannak a nyilvános származtatásnak?

4. Polimorfizmus, származtatás **1 pont**

Mit ír ki a következő program? Magyarázza az eredményt!

```

#include <iostream>
using namespace std;
class A {
public:
    virtual void g() { cout << "A"; }
    void f() { g(); }
};
class B: public A {
public:
    void g() { cout << "B"; }
};
main() {
    A a; B b; A* pa = &a;
    b.f(); pa->f(); pa = &b; pa->f();
    return 0;
}
    
```

1. Osztály, osztálysablon készítése

```
template <class T>
class matrix
{
public:
    matrix( int m, int n );
    matrix( const matrix &other);
    ~matrix();
    matrix operator=( const matrix &other);

    int rows() const { return m; }
    int cols() const { return n; }
    T& at( int i, int j) throw(indexError);
    T at( int i, int j) const
        throw(indexError);

private:
    int m;
    int n;
    T *v;
};
```

- Implementálja a `matrix` osztály műveleteit!
- Miért van szükség a második `at` függvényre? Adjon egy példát, amely e nélkül nem működne!

2. Beágyazott iterátor készítése

Egészítse ki az előző feladatban készített forráskódot úgy, hogy a következő kód érvényes legyen:

```
matrix<int> a( 10, 5 );
matrix<int> :: iterator it;
for(it = a.begin(); it != a.end(); ++it ) cout<< *it <<endl;
```

3. Tartalmazási kapcsolat osztályok között

A `vector<T>` osztálysablon felhasználásával készítsen egy verem osztályt.

- Használjon tartalmazási kapcsolatot az osztály megvalósítására.
- Készítsen kódrészletet, amelyben a saját veremosztálya segítségével megfordít egy karakterláncot.

4. Operátorok túlterhelése

Készítsen egy *raciónalis* számok ábrázolására szolgáló osztályt, csak a következő műveletekkel:

- implicit konstruktor
- inserter operátor
- extractor operátor

Készítsen kódrészletet, amely deklarál egy 12 elemű racionális számokból álló tömböt, beolvassa az elemeit az `in.txt` szövegállományból és kiírja a komplex számokat egy `out.txt` nevű szövegállományba. *Hibakezelést nem kell végezni!*

5. C++ alapismeretek

- Mit jelent a `const` módosító függvényre nézve? Hát függvényparaméterre nézve? Adjon példát mindkettőre!
- Mi a különbség a privát és a nyilvános származtatás között? Adjon egy-egy „értelmes” példát!