Reutilização de Software 2012/2013

Trabalho Prático n.º 1

Data Entrega: 26/10/2012

Implemente a classe Vector<T>, para representação de arrays com dimensão ilimitada, com o seguinte interface:

```
#pragma once
#include<string.h>
#include<cassert>
#include<iostream>
using namespace std;

template<class T>
class Vector {
      template<class U>
      friend ostream& operator<<(ostream&, const Vector<U>&);
private:
      T* data;
      unsigned len;
public:
      Vector(unsigned = 10);
      Vector(const Vector<T>&);
      virtual ~Vector(void);
      Vector<T>& operator =(const Vector<T>&);
      bool operator==(const Vector<T>&);
      T& operator[](unsigned);
};
```

Implemente a classe AssociativeArray<KeyType, ValueType> para gerir tabelas associativas usando: a) herança da classe Vector<T>; b) composição da classe Vector<T>. Os objectos da classe AssociativeArray deverão ter a funcionalidade descrita na seguinte função:

```
void testAssociativeArray() {
      AssociativeArray<String, int> table;
      table["abc"] = 15;
      table["jkl"] = 12;
      table["xyz"] = 85;
      assert(table["jkl"], 12);
}
```

Poderá usar a classe Pair<P, Q> definida por:

```
template<class P, class Q>
class Pair {
      P p;
      Q q;
public:
      Pair(const P& _p = P(), const Q& _q = Q()): p(_p), q(_q) {}
      P& objectP() {return p;}
      Q& objectQ() {return q;}

};
```

Nota: A entrega deverá ser efectuada na plataforma inforestudante.