

AS-0.1102 C/C++-ohjelmointi / Tentti 3.1.2006 / Aki Hiisilä

Vastaa neljään kysymykseen! Tentin arvosteluasteikko on 0 – 20 pistettä. Kaikkien kysymysten painoarvo on sama (5 pistettä/tehtävä). Ohjelmointitehtävät tulee kirjoittaa C++-kielellä hyvää ohjelmointityyliä noudattaen.

Palauta kolme konseptia, siten että ensimmäisellä konseptilla on vastaukset tehtäviin 1 ja 2, toisella konseptilla vastaukset tehtäviin 3 ja 4, sekä kolmannella konseptilla vastaus tehtävään 5.

Tehtävä 1

Toteuta alla listatussa tiedostossa "vector.hh" esitellyt kolme funktiota. Funktio *print* tulostaa vektorin tiedot muodossa "(x, y)". Funktio *getAverageLength* laskee vektoreiden pituuksien keskiarvon väliltä [start, stop[(jotka annetaan parametrina funktiolle). Funktio *sortByLength* järjestää parametrina annetun vektorilistan vektoreiden pituuksien mukaiseen järjestykseen siten, että ensimmäisenä listassa on lyhyin vektori.

```
#ifndef VECTOR_HH
#define VECTOR_HH

#include <vector>

namespace vector {

    struct Vector {

        double x;
        double y;
    };

    void print(Vector const &vector);
    double getAverageLength(std::vector<Vector>::const_iterator start,
                           std::vector<Vector>::const_iterator stop);
    void sortByLength(std::vector<Vector> &vectors);
}

#endif /* VECTOR_HH */
```

Huomautus: Funktioiden parametreille ei tarvitse tehdä järkevyyštarkastuksia.

Tehtävä 2

Mitä alla oleva ohjelma tulostaa?

```
#include <iostream>
#include <vector>
#include <algorithm>

class A {
public:
    A(int i = 0) : v(i) {}

    int operator()(int i)
    {
        this->v += 2;
        i += this->v;
        return i;
    }

    int v;
};

int main() {

    A a1; A a2(2); a1.v=0 a2.v=2
    int arr[4] = {1, 2, 3, 4};
    std::vector<int> vec;

    vec.push_back(10); vec.push_back(20); vec.push_back(30);

    std::cout << "A: " << a1.v << ", " << a2.v << std::endl;

    std::for_each(vec.begin(), vec.end(), a1); a1.v=6 vec = 12, 24, 36
    std::for_each(arr, arr + 3, a2); a2.v=10, arr = {5, 8, 11, 14}

    std::cout << "B: " << a1.v << ", " << a2.v << std::endl;
    std::cout << "C: " << vec[0] << ", " << vec[1] << std::endl;
    std::cout << "D: " << arr[2] << ", " << arr[3] << std::endl;

    std::transform(vec.begin(), vec.end(), arr, A(1)); arr = {15, 27, 39, 14}

    std::cout << "E: " << vec[0] << ", " << vec[1] << std::endl;
    std::cout << "F: " << arr[0] << ", " << arr[1] << std::endl;
    std::cout << "G: " << arr[2] << ", " << arr[3] << std::endl;

    std::transform(arr, arr + 2, std::back_inserter(vec), A());
    vec = 12, 24, 36, 17, 29, 41

    std::cout << "H: " << vec.size() << std::endl;
    std::cout << "I: " << vec[0] << ", " << vec[1] << std::endl;
    std::cout << "J: " << vec[2] << ", " << vec[3] << std::endl;

    return 0;
}
```

Tehtävä 3

C-kielillä on toteutettu kalojen käsittelyä varten abstrakti tietotyyppi (ADT) TFish ja sitä käsitteleviä operaatiofunktioita. Kaloja halutaan kuitenkin käsitellä C++-luokkana seuraavan esimerkissä esitellyllä tavalla:

```
#include "fish.hh"

int main(void)
{
    Fish fish1("Salmon", 1.23);
    Fish fish2(fish1);
    fish1 = fish2;
    fish2 = fish2;
    return 0;
}
```

Määrittele ja toteuta adapteri (adapter, wrapper) luokka C++-kielillä ja siten että yllä oleva koodi toimii. Toteutuksen tulee käyttää hyväkseen alla olevaa C-kielisen toteutuksen funktioita.

```
#ifndef FISH_H
#define FISH_H

typedef struct{
    char *species;
    double weight;
} TFish;

#ifdef __cplusplus
extern "C" {
#endif

TFish *fishConstruct(const char *species, double weight);
TFish *fishCopy(const TFish *f);
void fishDestruct(TFish *f);

#ifdef __cplusplus
}
#endif
#endif /* FISH_H */
```

Tehtävä 4

Mitä alla oleva ohjelma tulostaa?

```
#include <iostream>

class A {
public:
    A(int a = 0){ this->a = a; }
    A(A const &a){ this->a = a.a; std::cout << "C-C " << a.a << std::endl; }
    ~A(){ std::cout << "D " << this->a << std::endl; }
    A &operator=(A const &a1) {
        std::cout << "operator= " << a1.a << std::endl;
        this->a = a1.a;
        return *this;
    }
private:
    int a;
};

int func(int number) {
    A *a1 = new A(3);
    try{
        if(number <= 0){
            A a2 = *a1; throw 0;
        }
        else{
            { A a2(2); a2 = *a1; } throw number;
        }
    } catch(int e){
        std::cout << "func: " << e << std::endl;
        if(e == 0) throw;
    }
    return 0;
}

int main(void)
{
    A *a = new A(1);
    try {
        A a2(13);
        if(func(6) != 0){ throw "rv != 0"; }
        std::cout << "---" << std::endl;
        delete a;
        if(func(-5) != 0){ throw "rv != 0"; }
    } catch(int e){
        std::cout << "main: " << e << std::endl;
    }
    std::cout << "END" << std::endl;
    return 0;
}
```

Tehtävä 5

Alla on toteutettu C++ luokkana geneerinen tietorakenne pino. Luokan funktioiden toteutuksessa (rivit 20-37) on joitakin virheitä. Etsi ja perustele virhekohdat, sekä kerro miten korjaisit ne. Kirjoita myös pieni esimerkkiohjelma jossa käytät luotua pinoa tietotyypeillä `std::string` ja `std::vector<double>`.

```
1: #ifndef MYSTACK_HH
2: #define MYSTACK_HH
3:
4: template<class T>
5: class MyStack
6: {
7: public:
8:     MyStack();
9:     ~MyStack();
10:    void push(T const &item);
11:    T pop();
12:
13: private:
14:    T& operator=(T const &);
15:    MyStack(T const &);
16:    T *array;
17:    unsigned int size;
18: };
19:
20: template <class T> MyStack<T>::MyStack()
21: {
22:     this->array = NULL; size ei alusteta
23: }
24: template <class T> MyStack<T>::~MyStack()
25: {
26:     delete this->array; []
27: }
28: template <class T> void MyStack<T>::push(T const &element)
29: {
30:     this->size++; taulukon kokoa ei kasvateta
31:     this->array[this->size] = element; viittaus muistiin vapautettuaan
32: } muistipaikkaan
33: template <class T> T MyStack<T>::pop()
34: {
35:     return this->array[this->size - 1]; taulukon kokoa ei pienenneta
36: } muistia ei vapauteta
37:
38: #endif /* MYSTACK_HH */
```