

AS-0.1102 C/C++-ohjelmointi / Tentti 18.12.2006 / 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 "fish.hh" esiteltyt kolme funktiota. Funktio **print** tulostaa kalan tiedot muodossa "laji paino". Funktio **getAverageWeight** laskee parametrina annetun kalalistan kalojen painojen keskiarvon. Funktio **sortByWeight** järjestää parametrina annetun kalalistan kalojen painojen mukaiseen järjestykseen siten, että ensimmäisenä listassa on kevyin kala.

```
#ifndef FISH_HH
#define FISH_HH

#include <string>
#include <list>

namespace fish {

    struct Fish {
        std::string species;
        double weight;
    };

    void print(Fish const &fish);
    double getAverageWeight(std::list<Fish> const &fishes);
    void sortByWeight(std::list<Fish> &fishes);
}

#endif /* FISH_HH */
```

Huomautus: Tehtävässä voidaan olettaa, että funktiolle `getAverageWeight` parametrina annettavassakalalistassa on aina vähintään yksi alkio.

Tehtävä 2

Mitä alla oleva ohjelma tulostaa?

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

class Point {
public:
    Point(int x = 0, int y = 0) : x(x), y(y) {}

    double func(Point &p) {
        this->x = 7; p.y = 7; return 1.0;
    }
    int func(Point *p) {
        this->y = 9; p->x = 9; return 2;
    }
    void print(char c) const {
        std::cout << c << ": (" << this->x << ", "
            << this->y << ")" << std::endl;
    }
private:
    int x; int y;
};

void func(std::vector<Point> points, Point &p) {
    for (std::vector<Point>::size_type i=0; i<points.size(); i++) {
        points[i].func(p);
    }
}

int main()
{
    Point p1; Point p2(1, 2);
    std::vector<Point> points;
    points.push_back(p1); points.push_back(p2);

    p1.print('A'); p2.print('a');
    double d = p1.func(&p2);
    p1.print('B'); p2.print('b');
    p1.func(*(points.begin()));
    p1.print('C'); p2.print('c');
    func(points, p1);
    p1.print('D'); p2.print('d');
    (*(points.begin() + 1)).print('E');
    (*(points.end() - 1)).print('e');
    return 0;
}
```

Tehtävä 3

C-kielellä on toteutettu matriisien käsittelyä varten abstrakti tietotyyppi (ADT) TMatrix ja sitä käsitteleviä operaatiofunktioita. Matriiseja halutaan kuitenkin käsitellä C++-luokkana seuraavan esimerkissä esitellyllä tavalla:

```
#include "matrix.hh"

int main(void)
{
    Matrix matrix1(5, 5);
    Matrix matrix2(matrix1);
    matrix1 = matrix2;
    matrix2 = matrix2;
    return 0;
}
```

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

```
#ifndef MATRIX_H
#define MATRIX_H

typedef struct {
    int rows;
    int cols;
    double **cells;
} TMatrix;

#ifdef __cplusplus
extern "C" {
#endif

TMatrix *matrixConstruct(int rows, int cols);
TMatrix *matrixCopy(const TMatrix *m);
void matrixDestruct(TMatrix *m);

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

Tehtävä 4

Mitä alla oleva ohjelma tulostaa?

```
#include <iostream>

class A {
public:
    A(int a = 0){ this->a = a; }
    A(A const ka){
        this->a = a.a;
        std::cout << "C-C " << a.a << std::endl;
    }
    ~A(){ std::cout << "D " << this->a << std::endl; }
    A koperator=(A const ka1) {
        std::cout << "operator=" << a1.a << std::endl;
        this->a = a1.a;
        return *this;
    }
private:
    int a;
};

int func(int number) {
    A a1;
    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;
    }
    return 0;
}

int main(void) {
    A a(100);
    try {
        if(func(2) != 0){ throw "rv != 0"; }
        std::cout << "----" << std::endl;
        if(func(-5) != 0){ throw "rv != 0"; }
    } catch(const char *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 perustelee virhekohtat, sekä kerro miten korjaisit ne. Kirjoita myös pieni esimerkkiohjelma jossa käytät luotua pinoa tietotyypeillä **double** ja **std::string**.

```
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->size = 0;
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++;
31:     this->array[this->size] = element;
32: }
33: template <class T> T MyStack<T>::pop ()
34: {
35:     this->size--;
36:     return this->array[this->size - 2];
37: }
38:
39: #endif /* MYSTACK_HH */
```