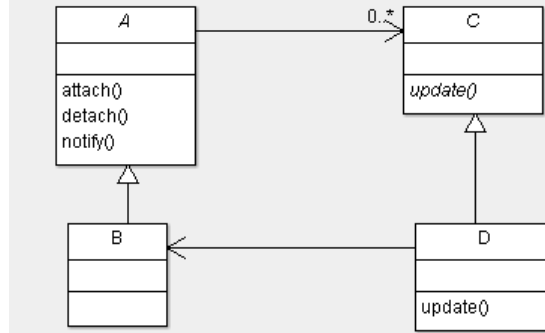


POO – Test scris A BAREM CORECTARE

07.06.2011

Observații:

1. Nu este permisă consultarea bibliografiei. 2. Toate întrebările sunt obligatorii. 3. Dacă nu este precizat altfel, fiecare întrebare este notată cu 3 puncte. Repartiția punctelor la întrebările grilă este: 1 punct alegerea corectă a variantei, 2 puncte justificarea. Alegerea corectă se punctează numai dacă justificarea este total sau parțial corectă. 4. Nu este permisă utilizarea de foi suplimentare.



Figură 1

- 1) Să se identifice ce șablon de proiectare este reprezentat în Fig. 1 și, în acest context, să se explice cum este aplicat principiul (conceptul) POO referitor la abstractizare. Se va preciza și cum este realizată aceasta abstractizare în C++.

Răspuns.

1. Identificare șablon (observer: A subiect și C observator) 1p
2. Clasele A și C sunt clase abstracte 0.5p
3. Metoda update() este abstractă; ea va fi implementată de clasele derivate 0.5p
4. Clasa A declară constructorii protejați 0.5p
5. Clasa C declară update() virtuală pură 0.5

- 2) Să se explice relațiile dintre clasele din diagrama reprezentată în Fig. 1.

Răspuns.

1. Relația între A și C - asociere 1p
2. Relația dintre B și D - asociere 1p
3. Relația dintre D - C și B - A 1p

- 3) Să se descrie în C++ relațiile dintre clasele din diagrama reprezentată în Fig. 1.

Răspuns.

1. Cod C++ relația între A și C - asociere 1p
2. Cod C++ relația dintre B și D - asociere 1p
3. Cod C++ relația dintre D - C și B - A 1p

4)

```
#include <iostream>
using namespace std;
class C{
public:
    C(int a = 0) : i(a) {cout << "C() ";}
    C(const C& x) {
        i = x.i;
        cout << "C(C&) ";
    }
    ~C() {cout << "~C() ";}
    operator int() const{
        cout << "int(C) "; return i;
    }
private: int i;
};
```

Să se precizeze ce va afișa programul alăturat.

- a) C() C(C&) C() ~C() int(C) int(C) 10 0 int(C) int(C) 10 ~C() ~C()
- b) C() C(C&) C() int(C) int(C) 10 0 int(C) int(C) 10 ~C() ~C() ~C()
- c) C() C(C) C() ~C() int(C) int(C) 10 0 int(C) int(C) 10 ~C() ~C()
- d) C() C(C&) C() int(C) int(C) 10 0 int(C) int(C) 10 ~C() ~C()

Justificare.

- Constructor implicit 0.5p
- Constructor copiere 0.5p
- Atribuire 0.5p
- Conversie 0.5p
- Raspuns corect a 1p

<pre>int main(){ C a, b = a; a = 10; int k = b; cout << a <<' ' << k << ' '; cout << a + b << ' '; return 0; }</pre>	
<p>5)</p> <pre>#include <iostream> #include <math.h> using namespace std; template <class T=int> class punct { public: punct (T xx, T yy){x=xx; y=yy;} T module () { return T(sqrt(x*x + y*y));} private:T x, y; }; template <> class punct <unsigned> { public: punct (int xx, int yy){x=xx; y=yy;} int module () { return x%y; } private: int x, y; }; int main () { punct <unsigned> u (10,3); punct <float> w(10,3); cout << u.module() << ' '; cout << int(w.module()); return 0; }:</pre>	<p>Ce este afisat după executia programului alăturat?</p> <p>a) 10 10 b) 1 1 c) 1 10 d) nimic deoarece programul are erori</p> <p>Justificare.</p> <ul style="list-style-type: none"> • Clasa parametrizata 0.5p • Specializare pentru unsigned 0.5p • Suprascriere module() 0.5p • Conversie de la float la int 0.5p • Varianta corectă c 1p
<p>6) Explicați următorul program și precizați rezultatul execuției?</p> <pre>#include <iostream> #include <map> using namespace std; int f(int x) { return (x - 1) * (x - 1); } int main(){ map<int, char> mp; map<int, char>::iterator it, it1, it2; char c = 'A'; for (int i= 0; i < 3; ++i) mp.insert(make_pair(f(i), c++)); it1 = mp.begin(); it2 = mp.end(); for (it = it1; it != it2; ++it) cout << "(" << it->first << ", " << it->second << "); "; return 0; }</pre>	<p>Răspuns.</p> <ul style="list-style-type: none"> • mp map cu cheia int și val char 0.5p • crearea lui mp 0.5p • utilizare iterator 0.5p • afișare 0.5p • rezultat corect 1p <p>(0,B); (1,A);</p>
<p>7)</p> <pre>#include <iostream> using namespace std; class A { public: virtual int f() = 0; }; class UNU:public A{ public: int f(){cout << b <<' '; return 1;} protected: static int b;</pre>	<p>Ce va afișa programul alăturat?</p> <p>a) 0 2 1 2 b) 1 1 2 2 c) 1 1 2 1 d) 1 2 2 2</p> <ul style="list-style-type: none"> • ierarhia de clasa 0.5p • funcția f virtuală 0.5p • a și b instanțe ale clasei DOI 0.5p

<pre> }; class DOI:public UNU { public: DOI(){b++;} int f() { cout << b << ' '; return 2; } }; int UNU::b = 0; int main(){ UNU* b = new DOI; cout << b->f() << " "; A* a = new DOI; cout << a->f() << " "; delete a; delete b; return 0; } </pre>	<ul style="list-style-type: none"> • este apelată DOI::f() 0.5p • rezultat corect d) 1p
---	---

8) (9 puncte) Un *text* este format din *cuvinte*. Fiecare cuvânt este un şir de caractere si are o *lungime*. *Lungimea* unui text este dată de suma lungimilor cuvintelor care îl compun la care se adaugă numărul de cuvinte (care corespunde spaţiilor şi *caracterului de terminare*). Un text este dinamic, în sensul că se poate adăuga sau şterge un cuvânt (de) la început.

- Să se descrie complet în C++ clasa *Cuvant* corespunzătoare cuvintelor. Se va utiliza tipul *string* din STL.
- Să se descrie complet în C++ clasa *Text* corespunzătoare textelor. Se va utiliza un container din STL pentru a reprezenta relaţia dintre clasa *Text* şi *Cuvant*.
- Să se descrie o secvenţă de program C++ care să creeze textul „Stiu raspunsul la aceasta intrebare.”

Răspuns .

- clasa *Cuvant*:
 - constructor 0.5
 - metoda *lungime()* 1p
 - atributul *sir* 0.5p
- clasa *Text*:
 - constructor 0.5
 - metoda *lungime()* 1p
 - adăugare 1p
 - ştergere 1p
 - atribut *cuvinte* 0.5p
- utilizarea claselor pt. crearea textului 2p
- stil de scriere 1p