Problema 10

Varianta care compileaza : da simplu 0p

-valori corecte: 0.3p

-valori apropriate (calcul matematic/semne greu vizibile) 0.2p

-valori indepartate 0.1p

-justificare (variabile si functii apelate) 0.2p

Varianta care nu compileaza: nu simplu/justificare gresita 0p

justificare 0.4p

modificare 0.1p

Varianta 1: NU COMPILEAZA linia 17 da eroare ca a nu e definit

#include <iostream>

using namespace std;

class A

{ int i;

public: A(int x=8):i(x) {}

virtual int f(A a)

{ return i+a.i; } };

class B: public A

{ int j;

public: B(int x=-2):j(x) {}

int f(B b) {return j+b.j; } };

int main()

{ int i=20;

A \*o;

if (i%4) {A a; o=new A(i);}

else {B b; o=new B(i);}

cout<<a->f(\*o);

return 0; }

justificare :

- % ramura corecta (else) 0.1

-a e declarat in scopul then 0.3

-a nu e pointer ci obiect 0.3

-modificare corecta 0.1

Varianta 2: NU COMPILEAZA -b nu e definit

#include <iostream>

using namespace std;

class A

{ int i;

public: A(int x=8):i(x) {}

virtual int f(A a)

{ return i+a.i; } };

class B: public A

{ int j;

public: B(int x=-2):j(x) {}

int f(B b) {return j+b.j; } };

int main()

{ int i=20;

A \*o;

if (i%4) {A a; o=new A(i);}

else {B b; o=new B(i);}

cout<<b->f(\*o);

return 0; }

- % ramura corecta (else) 0.1

-b este creat doar in interiorul scopului else 0.3

-b nu este pointer 0.3

-modificare corecta 0.1

Varianta 3: COMPILEAZA si afiseaza valoarea 16

#include <iostream>

using namespace std;

class A

{ int i;

public: A(int x=8):i(x) {}

virtual int f(A a)

{ return i+a.i; } };

class B: public A

{ int j;

public: B(int x=-2):j(x) {}

int f(B b) {return j+b.j; } };

int main()

{ int i=20;

A \*o;

if (i%4) {A a; o=new A(i);}

else {B b; o=new B(i);}

cout<<o->f(\*o);

return 0; }

- % ramura corecta (else) 0.1p

- pt obiectul o -constructor cu param x=20 0.1p

-care apeleaza constructorul pt A cu valoarea implicita x=8 (sau subinteles din calcul)0.1p

-apel f din clasa A -pt ca \*o e la compilare vazut ca obiect din clasa A 0.1p

- rezultat 8+8=16 0.1p

Varianta 4: COMPILEAZA si afiseaza si afiseaza valoarea -16

#include <iostream>

using namespace std;

class A

{ int i;

public: A(int x=-8):i(x) {}

virtual int f(A a)

{ return i+a.i; } };

class B: public A

{ int j;

public: B(int x=2):j(x) {}

int f(B b) {return j+b.j; } };

int main()

{ int i=20;

A \*o;

if (i%4) {A a; o=new A(i);}

else {B b; o=new B(i);}

cout<<o->f(\*o);

return 0; }

- % ramura corecta (else) 0.1

- pt obiectul o -constructor cu param x=20 0.1p

-care apeleaza constructorul pt A cu valoarea implicita x=-8 (sau subinteles din calcul)0.1p

-apel f din clasa A -pt ca \*o e la compilare vazut ca obiect din clasa A 0.1p

- rezultat -8-8=-16 0.1p

Varianta 5: COMPILEAZA si afiseaza valoarea 60

#include <iostream>

using namespace std;

class A

{ int i;

public: A(int x=-8):i(x) {}

virtual int f(A a)

{ return i+a.i; } };

class B: public A

{ int j;

public: B(int x=2):j(x) {}

int f(B b) {return j+b.j; } };

int main()

{ int i=30;

A \*o;

if (i%4) {A a; o=new A(i);}

else {B b; o=new B(i);}

cout<<o->f(\*o);

return 0; }

- % ramura corecta (then) 0.1

- pt obiectul o -constructor cu param x=30 0.1p

-apel f din clasa A si rezultat 30+30=60 0.3p