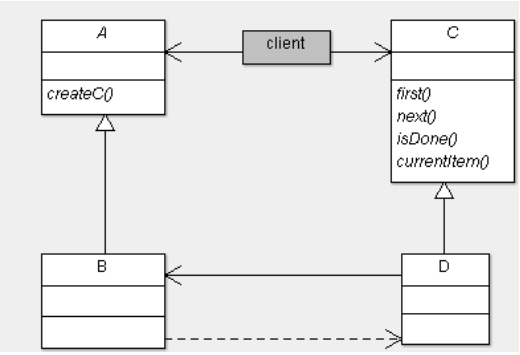
Universitatea "Al.I.Cuza" Iasi
Facultatea de Informatică

Nume:

An: I II III **Semian:** A B **Grupa:** 1 2 3 4 5 6

POO – Test scris 1 B - Barem
17.06.08

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 explice termenul „încapsulare" în contextul programării orientate-obiect. Se va preciza cum se implementează în C++.
Răspuns.

incapsulare = (combinare +) ascundere (1p)
avantaje (protejare date, fiabilitate, portabilitatea, managementul exceptiilor) (1p)
implementarea (nivele de acces in C++) (1p)

2) Să se explice componenta "Controller" din MVC.
Răspuns.

defineste comportarea aplicatiei (0.75)
mapeaza actiunile utilizator pe actualizare model (0.75)
selecteaza vizualizarea raspunsului (0.75)
unul pentru fiecare functionalitate (0.75)

3) Să se identifice ce șablon de proiectare (design pattern) este reprezentat în Figura 1. Să se descrie succint rolul fiecărui participant.
Răspuns.

Iterator (1p)

A = Aggregate (0.5p)
defineste interfata pentru crearea unui obiect Iterator
B = ConcreteIterator (0.5p)
implementeaza interfata Iterator.
memoreaza pozitia curenta in traversarea agregatului
C = Iterator (0.5p)
defineste interfata de accesare si traversare a componentelor
D = ConcreteAggregate (0.5p)
implementeaza interfata de creare a unui Iterator pentru a intoarce o instanta proprie ConcreteIterator.

4) Să se descrie în C++ clasa D din şablonul din Figura 1 (fără definiții metode).

Răspuns.

relatia de mostenire (0.75p)
implementarea operatii first() ... (0.75p)
relatia de asociere cu B (0.75p)
nivel de acces (0.75p)

5)

```
class Animal
{
...
};
class Dog : public Animal
{
...
};
class Cat : public Animal
{
...
};
int main()
{
  Animal a = ...;
  Dog d = ...;
  Cat c = ...;

  a = d;
  a = Animal(d);
  d = a;
  d = Dog(a);
  c = d;
  d = c;
  return 0;
}
```

La care dintre atribuirile alăturate compilatorul va da avertisment/eroare?
Răspuns + justificare.

d = a downcast (0.75p)
d = Dog(a) downcast (0.75p)
c = d crosscast (0.75p)
c = d crosscast (0.75p)

6) Să se precizeze ce funcții/operatori trebuie adaugate la clasele de la Exercițiul 5 pentru ca programul din main() să fie corect. Se va da prototipul (signatura) complet al fiecărui nou membru.

Răspuns + justificare.
Dog::Dog(Animal) (1p)
Dog::Dog(Cat) (1p)
Cat::Cat(Dog) (1p)

7) (9 puncte) Se consideră clasele FirstFit si BestFit cu următoarele responsabilități:

- ambele clase gestioneaza câte un tablou de dimensiune mare de elemente (pentru simplitate se poate presupune că elementele sunt numere întregi);
- elementele din tablou pot fi "ocupate" sau "libere";
- fiecare clasa include o operatie alloc(k) care întoarce adresa unei zone libere de lungime k;
- alloc(k) din Firstfit intoarce prima zona libera de lungime >= k găsită;
- alloc(k) din Bestfit intoarce zona libera de lungime cea mai mica >= k găsită.

Se cere:

a) Care soluție este mai bună: BestFit moştenește FirstFit sau invers? Există altă soluție mai bună decât moştenirea directă dintre cele doua clase?

Niciuna sau alta solutie acceptabila (1p)

Solutia cea mai buna: considerarea unei clase abstracte *Fit* din care deriveaza FisrFit si BestFit. (2p)

b) Să se descrie in C++ cele doua clase, conform răspunsului dat la a.

3 puncte (1.5 puncte fiecare clasa).

c) Să se descrie implementarea uneia dintre cele două metode alloc().
3 puncte (1 punct signatura, 1 punct algoritmul, 1punct scrierea corecta a metodei)