

Programmazione ad oggetti – Laurea in Informatica
Appello d’Esame – 21/3/2011

Nome..... Cognome.....

Matricola..... Laurea in.....

Non si possono consultare appunti e libri. Dove previsto scrivere CHIARAMENTE la risposta nell’apposito spazio.

Quesito 1

```
class A {
public:
    virtual void f(int x=0) { cout << "A::f(" << x << ")" << endl;}
};

class B : public A {
public:
    virtual void f(int x=1) { cout << "B::f(" << x << ")" << endl;}
};

int main() {

    A *p=new A;
    p->f();

    p=new B;
    p->f();

    B *q=new B;
    q->f();

    return 0;
}
```

Il precedente programma compila correttamente. (con gli opportuni `include` e `using`). Si scrivano le stampe prodotte dalla sua esecuzione. Se una riga non produce alcuna stampa (oltre a quella già indicata) si scriva **NESSUNA STAMPA**. Se una riga dovesse provocare un errore a run-time si scriva **ERRORE RUN-TIME**, e si lascino vuote le righe successive.

.....

.....

.....

Quesito 2

```
class A {
public:
    A() { cout << "A ";}
    A(const A&) {cout << "Ac ";}
    virtual ~A() {cout << "~A ";}
};

class B : virtual public A {
public:
    B() { cout << "B "; }
    ~B() {cout << "~B "; }
};

class C : virtual public A {
public:
    static A x;
    C() { cout << "C "; }
    C(const C&) {cout << "Cc ";}
};
A C::x;

class D : public B, public C {
public:
    D() { cout << "D "; }
    D& operator=(const D& d) {cout << "D= "; return *this;}
    ~D() {cout << "~D "; }
};

int main() {
    cout << "START" << endl;
    D *b=new D(); cout << "UNO" << endl;
    D d,e; cout << "DUE" << endl;
    e=d;cout << "TRE" << endl;

    return 0;
}
```

Il precedente programma compila correttamente. (con gli opportuni `include` e `using`). Si scrivano le stampe prodotte dalla sua esecuzione. Se una riga non produce alcuna stampa (oltre a quella già indicata) si scriva **NESSUNA STAMPA**. Se una riga dovesse provocare un errore a run-time si scriva **ERRORE RUN-TIME**, e si lascino vuote le righe successive.

.....

.....

.....

.....

.....

Quesito 3

EDS[®] è una compagnia che offre servizi di corriere espresso per i propri clienti. Offre diverse tipologie di trasporto pacchi: PaccoBase con una tariffa proporzionale al peso, PaccoDue che garantisce l'arrivo in due giorni in cambio di una tariffa aggiuntiva fissa, PaccoFast che garantisce l'arrivo il giorno successivo a quello di spedizione, con una tariffa aggiuntiva proporzionale al peso. Vogliamo modellare tale realtà con una opportuna gerarchia di classi.

Si definisca la seguente gerarchia di classi

1. Definire una classe PaccoBase alla base della gerarchia. Ogni oggetto di tale classe è caratterizzato dal mittente e destinatario, dal peso del pacco in Kg e dalla tariffa per chilogrammo applicata ai pacchi di tipo PaccoBase fissata in 10 euro al Kg.
Definire inoltre un metodo `costo` che ritorni il costo della spedizione del PaccoBase di invocazione.
2. Definire una classe PaccoDue derivata da PaccoBase. La tariffa fissa aggiuntiva per i pacchi di tipo PaccoDue è fissata in 2 euro. Il metodo `costo` deve ritornare il costo di spedizione del PaccoDue di invocazione.
3. Definire una classe PaccoFast derivata da PaccoBase. Un PaccoFast può prevedere la spedizione overnight, ovvero a partenza immediata (e notturna). La tariffa fissa aggiuntiva proporzionale al peso per i pacchi di tipo PaccoFast è fissata in 5 euro al Kg. Inoltre se la spedizione è overnight allora è previsto un ulteriore costo fissato in 7 euro. Il metodo `costo` deve ritornare il costo di spedizione del PaccoFast di invocazione.
4. Definire una classe AgenziaEDS i cui oggetti rappresentano un'agenzia EDS. Quindi ogni oggetto AgenziaEDS è caratterizzato dall'insieme dei pacchi in spedizione presso quella agenzia.
 - Definire l'overloading di `operator<<` per un'agenzia in modo tale che scriva in output: (1) il costo totale per la spedizione di tutti i pacchi in spedizione presso una agenzia; (2) per ogni pacco in spedizione presso l'agenzia, scriva in output
 - la tipologia del pacco con l'eventuale specifica overnight nel caso si tratti di un PaccoFast
 - il suo costo totale di spedizione
 - la eventuale tariffa aggiuntiva rispetto al costo di un PaccoBase.
 - Definire un metodo `cerca` che dato un destinatario D ed un peso p ritorna l'insieme di tutti i PacchiFast in spedizione overnight presso l'agenzia di invocazione che sono destinati a D e che hanno un peso superiore a p .
 - Definire un metodo `modifica` che dato un mittente M , modifica tutti i PacchiBase in spedizione presso l'agenzia di invocazione di mittente M in modo tale che essi diventino dei PacchiFast con spedizione overnight.
5. Scrivere un breve programma che illustri tutte le funzionalità sviluppate.

NB: Scrivere la soluzione **chiaramente** nel foglio a quadretti. Per comodità di correzione, definire tutti i metodi inline.