

Cognome e Nome _____

Matricola _____

Appello del 30 maggio 2017

Esercizio 1 (9 punti)

Cosa stampa il seguente programma Java?

```
class A {
    private int n;
    public A (int n){
        this.n=n;
    }
    public int getN(){
        return n;
    }
    void metodo (A a, int n){
        System.out.println ("metodo 1; " + n);
    }
    void metodo (B a, int n){
        System.out.println ("metodo 2; " + n);
    }
}

class B extends A{
    public B(int n){
        super (2*n);
    }
    void metodo (A a, int n){
        System.out.println ("metodo 3; " + n);
    }
}

public class MainClass {
    public static void main(String[] args) {
        A a = new A(5);
        B b = new B(4);
        A ab = new B(3);
        a.metodo(ab, ab.getN());
        ab.metodo(b, a.getN());
        ab.metodo(ab, b.getN());
    }
}
```



Esercizio 2 (10 punti)

Un array di **char** rappresenta in modo naturale una stringa. Ad esempio, l'array `{'c', 'a', 'r'}` rappresenta la stringa "car".

Scrivere un metodo

static boolean prefisso (char[] s1, char[] s2)

che, presi come parametro due array di char, restituisce true se e solo se la stringa rappresentata da **s1** è un prefisso di quella rappresentata da **s2**. Se **s1** o **s2** vagono **null**, viene restituito *false*.

Ad esempio se **s1**={ 'c', 'a' }, e **s2**={ 'c', 'a', 's', 'a' }, viene restituito true.

Non è possibile usare nessun metodo/operatore della classe **String**.

Esercizio 3 (15 punti)

Si progetti una classe **Cliente** con

- una variabile di istanza **nome** (tipo String, final, private)
- una variabile di istanza **cognome** (tipo String, final, private).

La classe deve avere:

- un costruttore che crea un Cliente dati **nome** e **cognome**;
- metodi pubblici accessori **getNome()**, **getCognome()**;
- un metodo pubblico **toString()** che restituisce la descrizione del cliente concatenando il nome e il cognome (separati da uno spazio)

Si progetti una classe **ListaDiAttesa** con

- una variabile di istanza **lista** di tipo ArrayList<Cliente> , private

La classe deve avere:

- un costruttore che crea una lista di attesa vuota (creando l'arrayList **lista** vuoto)
- un metodo **public void aggiungi(Cliente c)** che aggiunge alla fine della lista di attesa il cliente **c**
- un metodo **public Cliente estrai()** che elimina dalla lista il primo cliente in attesa e lo restituisce.
- un metodo **public int inAttesa()** che restituisce il numero dei clienti presenti nella lista d'attesa.
- un metodo pubblico **toString()** che restituisce la descrizione della lista di attesa elencando uno dopo l'altro i clienti in attesa preceduti dal numero d'ordine.

Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **90** minuti.
- Scrivere **subito** nome, cognome, matricola su **OGNI FOGLIO (compreso questo)**.
- Le risposte al primo esercizio devono essere date direttamente nel riquadro di questo foglio.
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- **Non** è possibile consultare appunti, libri, dispense o qualsiasi altro materiale.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc...) deve essere **completamente disattivato** e **depositato in vista sulla cattedra**
- Mettere in vista sul banco un valido documento di identità.