

Cognome e Nome _____

Matricola _____

Appello del 4 luglio 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(1);
        B b = new B(2);
        A ab = new B(3);
        ab.metodo(a, ab.getN());
        ab.metodo(b, b.getN());
        ab.metodo(ab, a.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 palindromo (char[] s)

che, presi come parametro un array di char, restituisce true se e solo se la stringa rappresentata da **s** è palindroma, ovvero può essere letta indifferentemente da sinistra verso destra o da destra verso sinistra.

Se **s** vale *null*, viene restituito *false*.

Ad esempio se **s**={`'a', 't', 'e', 'l', 'e', 't', 'a'`}, viene restituito true.

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

Esercizio 3 (15 punti)

Si progetti una classe **Ciente** con

- una variabile di istanza **name** (tipo String, final, private)
- una variabile di istanza **surname** (tipo String, final, private).

La classe deve avere:

- un costruttore che crea un Cliente dati **nome** e **cognome**;
- metodi pubblici accessori **getName()**, **getSurname()**;
- metodo pubblico **boolean equals (Cliente c)** che restituisce true se e solo se **this** e **c** hanno name e surname uguali (secondo l'equals tra stringhe).

Si progetti una classe **ListaDiAttesa** con

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

La classe deve avere:

- un costruttore che crea una lista di attesa vuota (creando l'arrayList **list** vuoto)
- un metodo **public void add (Cliente c)** che aggiunge alla fine della lista di attesa il cliente **c** (se **c** è già presente in lista, non viene aggiunto nuovamente; per effettuare il controllo, si sfrutti il metodo *equals* scritto nella classe Cliente)
- un metodo **public Cliente extract()** che elimina dalla lista il primo cliente in attesa e lo restituisce.
- un metodo **public boolean isWaiting(Cliente c)** che restituisce true se e solo se il cliente **c** è presente nella lista di attesa (si sfrutti il metodo *equals* scritto nella classe Cliente)

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.