

Modulo di Laboratorio di Programmazione (CdL Informatica)

a.a. 2018-2019

Prova Totale - 21 Gennaio 2019

NOME

COGNOME

MATRICOLA

Nota Bene: Il testo del compito deve essere riconsegnato insieme al proprio elaborato scritto. Le soluzioni degli esercizi **non** devono contenere alcun metodo **main** né alcuna istruzione di stampa né istruzioni **switch**, **break**, **continue**, **try-catch**, etc. La soluzione per l'Esercizio 3 **non** deve contenere alcuna istruzione di ciclo (**for**, **while**, etc.). La non osservanza di tali requisiti comporterà l'annullamento della prova scritta dello studente.

Esercizio 1. Si consideri il seguente frammento di codice in Java:

```
String[] [] a = b;
int i = 0, j = 0, k = 0; boolean p = false; String s = "";
while (i < a.length && !p) {
    for (j = 0; j < a[i].length-1; j++) {
        if (a[i][j].length() <= a[i][j+1].length()) {
            p = true;
            s = s + a[i][j];
        }
    }
    if (p) k = i;
    i++;
}
```

Determinare il valore finale delle variabili **i**, **j**, **k**, **p** ed **s** nei seguenti casi:

- 1) $b = \{\{"jkr", "tw"\}, \{"fgx", "b", ""\}, \{"vx", "pdf", "sc", "qz"\}, \{"r", "yy"\}\}$;
- 2) $b = \{\{"bd"\}, \{"sd", "bb", "m", "fs", "gf"\}, \{"hf", "kj"\}\}$.

Esercizio 2. Scrivere un metodo *statico* iterativo che, dati un array bidimensionale di stringhe a ed un array monodimensionale di interi b , restituisce un array monodimensionale di stringhe c tale che $c[i]$ è la *prima* stringa nella riga $a[i]$ (letta da sx a dx) la cui lunghezza è maggiore di $b[i]$. Se in $a[i]$ non esiste alcuna stringa che soddisfa tale condizione, allora $c[i]$ è la stringa vuota. Esempio: se $a = \{\{"boh", "argh", "oops"\}, \{"", "uffa", "ok", "yes"\}, \{"no", "???"\}\}$ e $b = \{3, 2, 4\}$, il metodo restituisce l'array $c = \{"argh", "uffa", ""\}$.

N.B. i) Occorre passare alla riga successiva non appena si trova la prima stringa in $a[i]$ che soddisfa la condizione richiesta. ii) Il metodo deve sollevare un'opportuna eccezione se le lunghezze di a e b non sono uguali.

Esercizio 3. Scrivere una versione ricorsiva del metodo definito per risolvere l'Es.2.

Esercizio 4. Un *tragitto fatto in treno* può essere caratterizzato da nome della stazione di partenza, nome della stazione di arrivo, numero del treno, tipologia del treno, distanza percorsa (espressa in numero di km), ora di partenza e ora di arrivo. Scrivere una classe **TragittoT** con un opportuno costruttore ed i metodi che restituiscono i valori delle variabili istanza. Definire inoltre un metodo che modifica la tipologia, un metodo che, dati due tragitti in treno, verifica se hanno la stessa stazione di partenza e la stessa stazione di arrivo ed un

metodo che, dati due tragitti in treno, restituisce quello con la distanza maggiore. Definire infine un metodo che restituisce una stringa che descrive un oggetto della classe `TragittoT`.

Esercizio 5. Un *tragitto in treno con prenotazione* è un tragitto in treno caratterizzato anche dal numero di livelli di servizio offerti e dalla presenza o meno di servizio ristoro a bordo. Scrivere una classe `TragittoTPren` con un opportuno costruttore ed i metodi che restituiscono i valori delle variabili istanza. Definire inoltre un metodo che modifica il numero di livelli di servizio ed un metodo che, dati due tragitti in treno con prenotazione, verifica se hanno lo stesso numero di livelli di servizio e collegano le stesse stazioni (utilizzare il metodo definito nella classe dell'Es.4). Definire infine un metodo che restituisce una stringa che descrive un oggetto della classe `TragittoTPren`.

Esercizio 6. Un *orario delle partenze di tragitti in treno* può essere caratterizzato tramite nome della stazione, anno di validità e l'elenco dei tragitti in treno che partono dalla stazione nell'orario. Scrivere una classe `OrarioP`, il cui costruttore imposta il numero massimo di tragitti in treno di un orario. Oltre ai metodi che restituiscono i valori delle variabili istanza, definire i seguenti metodi:

- un metodo che, dato un tragitto in treno t , aggiunge t nell'elenco dei tragitti in treno di un orario a patto che la stazione di partenza di t coincida con quella dell'orario (il metodo deve sollevare opportune eccezioni *distinte* in caso di nomi di stazione diversi e di elenco pieno);
- un metodo che, dato un intero k , restituisce l'elenco dei tragitti in treno che partono dopo l'ora k ;
- un metodo che, data una stringa s , restituisce il numero dei tragitti in treno con prenotazione che hanno s come stazione di arrivo.

Esercizio 7. Siano date le seguenti classi:

```
class A {
    private String str;

    public A (String s) {
        this.str = s;
    }
    public String getS() {
        return this.str;
    }
    public String m (A a) {
        if (a.str.charAt(0) ==
            this.str.charAt(0))
            return this.str + a.str;
        else
            return a.str + this.str;
    }
}

class B extends A {
    private int val;

    public B (String s, int v) {
        super(s);
        this.val = v;
    }
    public int getV() {
        return this.val;
    }
    public String m (A a) {
        if (a.getS().length() <= this.val)
            return this.val + super.m(a);
        else
            return super.m(a) + this.val;
    }
}
```

Dato il seguente frammento di codice:

```
A x = new A ("esami");
A y = new B ("laurea",7);
B z = new B ("linguaggio",5);
String s = x.m(z), t = y.m(x), u = z.m(y);
```

determinare il valore delle variabili s , t ed u .