

Esame Fondamenti di Informatica – 1° appello: 01-02-2023

La documentazione delle API di jdk può essere consultata. Se in dubbio, chiedete alla docente o ai sorveglianti se è ammesso l'utilizzo di una particolare classe e/o metodo

NB: verranno corretti solo i file .java che si trovano nella cartella home e che sono **compilabili** in tale cartella. Se usate Eclipse o altro assicuratevi di copiare i file e renderli compilabili (togliendo istruzioni di definizioni di package o altro inserite da Eclipse e che non rendono i sorgenti compilabili nella cartella home).

Importante: nella prima riga di ciascun file realizzato scrivere sempre in un commento il proprio nome, cognome, numero di matricola.

Esercizio 1 prova di programmazione [11 punti]

Implementare una classe eseguibile il cui nome deve essere *CognomeMatricolaMaxRun.java*, dove Cognome e Matricola **devono essere il proprio nome e la propria matricola**.

Progettare una classe eseguibile che legga da standard input una stringa e che riporti in uscita la più lunga *run* in essa presente. Una run è una sequenza di caratteri consecutivi uguali. Ad esempio:

input: ciaooocaaaaasa

output: aaaaa

Se sono presenti più run con la stessa lunghezza massima, riportare in uscita la prima.

Ad esempio:

input: ciaooooocaaaaasa

output: ooooo

Per individuare la run realizzare un metodo statico che riceva come parametro la stringa da analizzare e restituisca in uscita un valore intero corrispondente all'indice in cui inizia la run.

Riassumendo, la classe eseguibile da realizzare deve:

- leggere da standard input una stringa
- invocare il metodo statico che individua l'indice di partenza della run più lunga
- stampare in uscita la run più lunga.

Esercizio 2 prova di programmazione [11 punti]

Sia data un'implementazione *ArraySet* di un ADT Set implementata come array riempito in parte ridimensionabile (file *ArraySet.java*, *Set.java*, *Container.java*).

Implementare una sua sottoclasse, chiamata **OrganizationsArraySet**, il cui codice deve essere scritto in un file **OrganizationsArraySet.java**. Nella sottoclasse si dovrà:

- sovrascrivere il metodo **add** in modo che OrganizationsArraySet possa contenere solo stringhe (in caso negativo lanciare l'eccezione **IllegalArgumentException**).
- implementare un metodo con firma:
public String[] sortOrganizations()
che restituisca un array di stringhe contenente i nomi delle organizzazioni presenti nella struttura dati in ordine crescente. E' possibile utilizzare qualsiasi algoritmo di ordinamento visto a lezione. Il contenuto dell'array v utilizzato dalla struttura dati non deve essere modificato.
- implementare un metodo con firma:
public String[] StringArraySet acronyms()
che restituisca l'elenco degli acronimi delle stringhe presenti nel set nello stesso ordine in cui sono ordinate le stringhe dal metodo precedente (NB: si assuma che tutte le iniziali delle parole presenti nel nome dell'organizzazione concorrano a formare l'acronimo).

E' fornito un tester eseguibile **ProfSetTester.class** che può aiutarvi nel debugging.

Esercizio 3 prova di programmazione [11 punti]

Implementare una classe eseguibile CognomeOrganizationsTester (dove "Cognome" va sostituito con i propri dati) :

- Creare un oggetto OrganizationsArraySet
- Leggere dal file org.txt i nomi delle organizzazioni e inserirli nella struttura dati. Il file contiene le parole che formano il nome completo di un'organizzazione sulla stessa riga.
- Stampare il contenuto del set in ordine
- Stampare gli acronimi che iniziano con 'E' oppure con 'I'
- Gestire tutte le possibile eccezioni

IMPORTANTE: nella prima riga di tutte le classi realizzate inserite anche un commento con cognome, nome, numero di matricola.

Esempio di output:

```
MacBook-Pro-di-Cinzia:esame01022023 cinzia$ java PizziOrganizationTester
Agenzia Spaziale Italiana
Associazione Nazionale Comuni Italiani
Automobil Club Italia
Consiglio Nazionale Ricerche
European Commission
European Space Agency
Food Agriculture Organization
International Energy Agency
Istituto Nazionale Geofisica Vulcanologia
Ministero Istruzione Universita' Ricerca
North Atlantic Treaty Organisation
Organizzazione Nazioni Unite
Servizio Bibliotecario Nazionale
EC
ESA
IEA
INGV
```