

*Università degli Studi di Trieste*

Corsi di ingegneria industriale e navale

**Esercitazioni di Fondamenti di Informatica**

Giacomo Strangolino

mailto/chat: [delleceste@gmail.com](mailto:delleceste@gmail.com)

**Materiale didattico anche su:**

**<http://www.giacomos.it>**

**(<http://www.giacomos.it/didattica/units/2011/>)**

# Lezione 9

## Esercizi sui file

- Si scriva una classe che legga da file un dizionario (*SMScompress.dat*) in cui ogni riga sia formata da:  
parola\_intera <spazio> sua\_abbreviazione
  - Successivamente, dato un file in ingresso che contiene il testo di un SMS, produca un file di uscita con l'SMS compresso secondo le abbreviazioni date nel dizionario.
- Si scriva un modulo *estrai\_capoverso.py* con una funzione *estrai()* che, ricevuti in ingresso un parametro con un nome di file di input, una lista di stringhe e un nome di file di output, cerchi le occorrenze di ogni parola chiave della lista all'interno di ogni capoverso. Si supponga che il capoverso sia delimitato da un “punto e a capo”: “. \n” . La funzione scrive sul file di output tutti i capoversi trovati, separandoli con una riga di trattini ed evidenziando le parole chiave mediante due '\*' Il capoverso individuato, per essere valido, deve contenere *tutte* le parole chiave passate come secondo parametro.

# Lezione 9

## *Homework*

Si progetti un gioco per la battaglia navale.

- Vi saranno  $N$  navi, rappresentate ciascuna dall'oggetto *Nave*, con una posizione della prua, una dimensione e un orientamento (orizzontale o verticale) sulla scacchiera.

La classe *Nave* dovrà avere dei metodi che dicano se è *colpita*( $x, y$ ) o addirittura *affondata*()

- Un oggetto *Scacchiera* conterrà le  $N$  navi in una lista e dovrà fornire il metodo *colpisci*( $x, y$ ) che chiede ad ogni nave se ( $x, y$ ) è una posizione che la danneggia o la affonda nel caso estremo.

La *Scacchiera* avrà altresì il metodo *disegna*() che ad ogni mossa eseguita dal giocatore deve disegnare la battaglia navale sotto forma di matrice contenente opportuni simboli adatti a descrivere la situazione corrente...

# Lezione 9

## *Homework*

### Suggerimenti

- Si usi la funzione *randint(a,b)* del modulo random per generare un numero casuale tra 0 e N (ne serviranno 3, uno per la posizione x della prua, uno per la y e uno (0 oppure 1) per l'orientazione (orizzontale o verticale) di ogni nave).
- Se durante la generazione delle navi due navi si scontrano, bisogna rigenerare l'ultima nave finché non ci sono più intersezioni...
- La funzione che usa la classe Scacchiera dovrà eseguire un ciclo infinito finché o tutte le navi sono affondate o l'utente inserisce una stringa particolare, tipo “esci”, per terminare anticipatamente il gioco.

# Lezione 9

## *Homework*

- Si scriva una funzione *find(filename, keyword)* che trovi nel file *filename* la parola chiave *keyword* e indichi il numero di linea in cui essa compare e quante volte.
- Si scriva una funzione che, dato il nome di un file, lo legga carattere per carattere e indichi la linea e la posizione all'interno della linea in cui trova un carattere non ASCII standard.
- Riscrivere il progetto *Università* in modo che carichi studenti ed esami da file, nel modo indicato in un *homework* di una lezione precedente e inoltre si aggiunga un metodo *salva()* alla classe *Universita* che salvi su file gli studenti e i loro esami. In pratica i file salvati dal metodo *salva()* sono gli stessi due che vengono dati in ingresso per il caricamento di studenti ed esami, e quindi il formato in cui l'informazione viene salvata deve essere compatibile.