Capitolo 1 Linguaggio Java

1.1 Compilare ed eseguire un programma java che scrive “ciao” sullo schermo

Capitolo 2 Programmazione ad oggetti in Java

2.1 Definire una classe che rappresenta una automobile e una classe Applicazione che la utilizza.

2.2 Definire una classe che rappresenta un televisore una classe Applicazione che la utilizza.

Capitolo 3 Elementi del linguaggio

3.1 Definire una classe che rappresenta un intero

Definire metodi che verificano se:

l’intero è un numero primo

è perfetto (è pari alla somma dei suoi divisori)

è il quadrato di un altro numero intero

definire un metodo che calcola la scomposizione in fattori primi

Capitolo 4 Manipolazione degli oggetti

4.1 Definire una classe che rappresenta un vettore di interi

Si definisca un costruttore della classe che ha un parametro intero n e che istanzia il vettore V con n elementi casuali.

Definire metodi per:

visualizzare il vettore

calcolare la somma degli elementi del vettore

calcolare la media degli elementi del vettore

calcolare il massimo degli elementi del vettore

calcolare il minimo degli elementi del vettore

calcolare la moda degli elementi del vettore (il valore che è presente più volte; si supponga il vettore non ordinato)

calcolare la moda degli elementi del vettore (il valore che è presente più volte; si supponga il vettore ordinato)

dato un elemento restituire la posizione dell’elemento nel vettore (si supponga il vettore non ordinato)

dato un elemento restituire la posizione dell’elemento nel vettore (si supponga il vettore ordinato)

contare gli elementi pari

contare gli elementi maggiori di un valore dato

contare gli elementi minori di un valore dato (si supponga il vettore ordinato)

ordinare il vettore

restituire l’unione ordinata di 2 dati vettori ordinati

restituire l’intersezione ordinata di 2 dati vettori ordinati

4.2 Si definisca una lista concatenata tramite puntatori a partire dalle classi:
Elemento, che contiene i campi: *valore* (di tipo intero) e *successivo* (di classe *Elemento*);
Lista, che contiene il campo *testa* (di classe *Elemento*), che rappresenta il riferimento al primo elemento della lista.

visualizzare la lista
calcolare la somma degli elementi della lista
calcolare la media degli elementi della lista
calcolare il massimo degli elementi della lista
calcolare il minimo degli elementi della lista
calcolare la moda degli elementi della lista (il valore che è presente più volte; si supponga la lista non ordinato)
calcolare la moda degli elementi della lista (il valore che è presente più volte; si supponga la lista ordinato)
dato un elemento restituire la posizione dell'elemento nella lista (si supponga la lista non ordinata)
dato un elemento restituire la posizione dell'elemento nella lista (si supponga la lista ordinato)
contare gli elementi pari
contare gli elementi maggiori di un valore dato
contare gli elementi minori di un valore dato (si supponga la lista ordinata)
ordinare la lista
restituire l'unione ordinata di 2 dati liste ordinate
restituire l'intersezione ordinata di 2 dati liste ordinate

Dare anche una versione ricorsiva dei metodi: in questo caso, usando il sovraccarico, definire una funzione senza parametri (quella che richiama l'utente), che richiama quella ricorsiva.

Polimorfismo

4.3 Si definisca una gerarchia di classi per rappresentare i giocatori di una squadra di calcio

si definisca la classe Giocatore che contiene:
- il campo nome (di classe String)
- il campo eta (un intero)
- il metodo costruttore (che ha nome ed eta come parametri)

si definisca la classe Portiere sottoclasse di Giocatore che contiene:
- il campo rigoriParati (un intero),
- il metodo costruttore (che ha nome, eta e rigoriParati come parametri)
- un metodo che restituisce il valore del giocatore (1000000 per rigoriParati – 1000 per eta).

si definisca la classe Attaccante sottoclasse di Giocatore che contiene:
- il campo golSegnati (un intero),
- il metodo costruttore (che ha nome, eta e golSegnati come parametri)
- un metodo che restituisce il valore del giocatore (5000000 per golSegnati – 5000 per eta).

Si definisca una classe Squadra che contiene come campo un array V in cui è possibile inserire giocatori sia di classe Portiere che Attaccante..

Si definisca inoltre un costruttore della classe Squadra che ha due parametri interi m e n e che istanzia il vettore V con m+n elementi inserendo m Portieri ed n attaccanti. Si definiscano i campi degli oggetti in modo casuale.
Si definisca infine un metodo della classe Squadra che restituisce il nome del giocatore con il valore più alto.

Se necessario si modifichi la gerarchia di classi definita sopra.

4.4 Si definisca una gerarchia di classi per rappresentare i clienti di una agenzia di viaggi:

si definisca la classe Cliente che contiene:
- il campo nome (di classe String)
- il campo permanenza (un intero)
- il metodo costruttore (che ha nome e permanenza come parametri)

si definisca la classe Marittimo sottoclasse di Cliente che contiene:
- il campo coefficiente (un intero),
- il metodo costruttore (che ha nome, permanenza e coefficiente come parametri)
- un metodo che restituisce il costo del soggiorno (permanenza moltiplicata per coefficiente).

si definisca la classe Montanaro sottoclasse di Cliente che contiene:
- il campo coefficiente (un double),
- il metodo costruttore (che ha nome, permanenza e coefficiente come parametri)
- un metodo che restituisce il costo del soggiorno (permanenza moltiplicata per coefficiente).

Si definisca una classe Archivio che contiene come campo un array V in cui è possibile inserire clienti sia di classe Marittimo che Montanaro.
Si definisca inoltre un costruttore della classe Archivio che ha due parametri interi m e n e che instanzia il vettore V con m+n elementi inserendo m Marittimi ed n Montanari. Si definiscano i campi degli oggetti in modo casuale.
Si definisca infine un metodo della classe Archivio che, dato il nome di un cliente, restituisce il costo del cliente se questi è presente nel vettore V, il valore -1 altrimenti.
Per questo metodo si supponga che i nomi dei clienti siano tutti diversi.

Se necessario si modifichi la gerarchia di classi definita sopra.

4.5 Si definisca una gerarchia di classi per rappresentare gli animali di un'oasi naturalistica:

si definisca la classe Animale che contiene:
- il campo nome (di classe String)
- il campo peso (un double)
- il metodo costruttore (che ha nome e peso come parametri)

si definisca la classe Felino sottoclasse di Animale che contiene:

- il campo coefficiente (un intero),
- il metodo costruttore (che ha nome, peso e coefficiente come parametri)
- un metodo che restituisce il costo per l'alimentazione (peso moltiplicato per coefficiente).

si definisca la classe Pachiderma sottoclasse di Animale che contiene:

- il campo coefficiente (un double),
- il metodo costruttore (che ha nome, peso e coefficiente come parametri)
- un metodo che restituisce il costo per l'alimentazione (peso moltiplicato per coefficiente).

Si definisca una classe Oasi che contiene come campo un array V in cui è possibile inserire sia felini che pachidermi.
Si definisca inoltre un costruttore della classe Oasi che ha un parametro n e che instanzia il vettore V con n elementi inserendo felini nelle posizioni pari, pachidermi nelle dispari. Si definiscano i campi degli oggetti in modo casuale.
Si definisca infine un metodo della classe Oasi che restituisce il nome dell'animale presente in V il cui costo per l'alimentazione è più elevato.

Se necessario si modifichi la gerarchia di classi definita sopra.

4.6 Si definisca una gerarchia di classi per rappresentare gli attori di una agenzia:
si definisca la classe Attore che contiene:

- il campo nome (di classe String)
- il campo età (un double)
- il metodo costruttore (che ha nome ed età come parametri)

si definisca la classe Comico sottoclasse di Attore che contiene:

- il campo filmfatti (un intero),
- il metodo costruttore (che ha nome, età e filmfatti come parametri)
- un metodo che restituisce il gradimento da parte del pubblico (filmfatti moltiplicato per 0.75).

si definisca la classe Drammatico sottoclasse di Attore che contiene:

- il campo premi (un double),
- il metodo costruttore (che ha nome, età e premi come parametri)
- un metodo che restituisce il gradimento da parte del pubblico (premi moltiplicato per 1.25).

Si definisca una classe Archivio che contiene come campo un array V in cui è possibile inserire sia comici che drammatici.

Si definisca inoltre un costruttore della classe Archivio che ha un parametro n e che instanzia il vettore V con n elementi inserendo un terzo (circa) di comici e due terzi (circa) di drammatici. Si definiscano i campi degli oggetti in modo casuale.

Si definisca infine un metodo della classe archivio che dato un parametro numerico restituisce il numero di attori presenti nell'archivio il cui gradimento è maggiore del parametro.

Se necessario si modifichi la gerarchia di classi definita sopra.

4.7 Si definisca una gerarchia di classi per rappresentare i libri di una biblioteca:
si definisca la classe Libro che contiene:
- il campo titolo (di classe String)
- il campo annopubblicazione (un int)
- il metodo costruttore (che ha titolo ed annopubblicazione come parametri)

si definisca la classe Moderno sottoclasse di Libro che contiene:
- il campo costo (un double),
- il metodo costruttore (che ha titolo, annopubblicazione e costo come parametri)
- un metodo che restituisce il valore del libro (costo moltiplicato per 0.75+ (3000-annopubblicazione)).

si definisca la classe Antico sottoclasse di Libro che contiene:
- il campo stampato (un booleano),
- il metodo costruttore (che ha titolo, annopubblicazione e stampato come parametri)
- un metodo che restituisce il valore del libro ((3000-annopubblicazione) moltiplicato 2.5 + 1000 se il libro non è stampato).

Si definisca una classe Biblioteca che contiene come campo un array V in cui è possibile inserire sia moderni che antichi.
Si definisca inoltre un costruttore della classe Biblioteca che ha un parametro n e che instanzia il vettore V con n elementi inserendo un metà di moderni e metà di antichi. Si definiscano i campi degli oggetti in modo casuale.
Si definisca infine un metodo della classe Biblioteca che restituisce il titolo del libro di valore massimo.
Se necessario si modifichi la gerarchia di classi definita sopra.

4.8 Si definisca una gerarchia di classi per rappresentare le navi di una compagnia di navigazione:
si definisca la classe Nave che contiene:
- il campo nome (di classe String)
- il metodo costruttore (che ha nome come parametro)

si definisca la classe Crociera sottoclasse di Nave che contiene:
- il campo capienza (un intero),
- il metodo costruttore (che ha nome e capienza come parametri)
- un metodo che restituisce il costo di iscrizione al registro (capienza moltiplicato per 100.50).

si definisca la classe Trasporto sottoclasse di Nave che contiene:
- il campo tonnellaggio (un double),
- il metodo costruttore (che ha nome e tonnellaggio come parametri)

- un metodo che restituisce il costo di iscrizione al registro (tonnellaggio moltiplicato per 150).

Si definisca una classe Archivio che contiene come campo un array V in cui è possibile inserire sia navi da Crociera che navi da Trasporto.

Si definisca inoltre un costruttore della classe Archivio che ha un parametro n e che instanzia il vettore V con n elementi inserendo metà navi da Crociera e metà navi da Trasporto. Si definiscano i campi degli oggetti in modo casuale.

Si definisca infine un metodo della classe archivio che dato un parametro numerico verifica se nell'archivio è presente una nave il cui costo di iscrizione al registro sia inferiore al parametro.

Se necessario si modifichi la gerarchia di classi definita sopra.

4.9 Si definisca una gerarchia di classi per rappresentare gli atleti di una società polisportiva:
Gli atleti sono di due tipi: Pallavolista (che contiene i campi nome, età, partite giocate) e Sprinter (che contiene i campi nome, età, vittorie).
Entrambe le classi contengono un metodo "premio" che restituisce l'importo di un premio di fine stagione. Per il Pallavolista il premio è calcolato moltiplicando per 100 il numero di partite giocate, mentre per lo Sprinter il premio è calcolato moltiplicando per 1000 il numero di vittorie.

Si definisca una classe Archivio che contiene come campo un array V in cui è possibile inserire sia Pallavolisti che Sprinter.

Si definisca inoltre un costruttore della classe Archivio che ha un parametro n e che instanzia il vettore V con n elementi inserendo metà Pallavolisti e metà Sprinter. Si definiscano i campi degli oggetti in modo casuale.

Si definisca infine un metodo della classe Archivio che restituisce la media dei premi da assegnare agli atleti presenti nel vettore V.

4.10 Si definisca una gerarchia di classi per rappresentare gli atleti di una società sportiva:
Gli atleti gareggiano in due tipi di sport: di squadra (l'atleta contiene i campi nome, età, partite vinte, numero dei giocatori della squadra) e individuale (l'atleta contiene i campi nome, età, gare vinte).
Entrambe le classi contengono un metodo "premio" che restituisce l'importo di un premio di fine stagione. Per gli atleti degli sport di squadra il premio è calcolato moltiplicando per 1000 il numero di partite vinte e dividendo per il numero dei giocatori della squadra, mentre per gli atleti di gare individuali il premio è calcolato moltiplicando per 500 il numero di gare vinte.

Si definisca una classe Archivio che contiene come campo un array V in cui è possibile inserire atleti sia di sport di squadra che individuali.

Si definisca inoltre un costruttore della classe Archivio che ha un parametro n e che instanzia il vettore V con n elementi inserendo circa 1/3 di atleti di sport individuali e 2/3 di atleti di sport di squadra. Si definiscano i campi degli oggetti in modo casuale.

Si definisca infine un metodo della classe Archivio che restituisce la somma dei premi da assegnare agli atleti presenti nel vettore V.

4.11 Si definisca una gerarchia di classi per rappresentare i giocatori di una squadra di calcio

si definisca la classe Giocatore che contiene:

- il campo nome (di classe String)
- il campo eta (un intero)
- il metodo costruttore (che ha nome ed eta come parametri)

si definisca la classe Portiere sottoclasse di Giocatore che contiene:

- il campo rigoriParati (un intero),
- il metodo costruttore (che ha nome, eta e rigoriParati come parametri)
- un metodo che restituisce il valore del giocatore (1000000 per rigoriParati – 1000 per eta).

si definisca la classe Attaccante sottoclasse di Giocatore che contiene:

- il campo golSegnati (un intero),
- il metodo costruttore (che ha nome, eta e golSegnati come parametri)
- un metodo che restituisce il valore del giocatore (5000000 per golSegnati – 5000 per eta).

Si definisca una classe Squadra che contiene come campo un array V in cui è possibile inserire giocatori sia di classe Portiere che Attaccante..
Si definisca inoltre un costruttore della classe Squadra che ha due parametri interi m e n e che istanzia il vettore V con m+n elementi inserendo m Portieri ed n attaccanti. Si definiscano i campi degli oggetti in modo casuale.
Si definisca infine un metodo della classe Squadra che restituisce il nome del giocatore con il valore più alto.

Se necessario si modifichi la gerarchia di classi definita sopra.

4.12 Si definisca una gerarchia di classi per rappresentare gli Elettrodomestici di casa. Pre ogni elettrodomestico si rappresenti colore e potenza (un double).
Si definiscano le sottoclassi Frigorifero con il campo capienza (un intero) e Televisione con i campi pollici (un intero) e canone (un double)
Entrambe le classi contengono un metodo “costo” che restituisce l’importo del costo di mantenimento.

Per il frigorifero il costo è il prodotto fra potenza e capienza, per i televisori è il canone più il prodotto fra la potenza ed i pollici.
.
Si definisca una classe Casa che contiene come campo un array V in cui è possibile inserire sia frigoriferi che televisori.

Si definisca inoltre un costruttore della classe Casa che ha due parametri n e m che instanzia il vettore V con n frigoriferi ed m televisori. Si definiscano i campi degli oggetti in modo casuale.

Si definisca infine un metodo della classe Casa che restituisce la somma del costo di mantenimento degli elettrodomestici.

Capitolo 5 Applet

Si definisca una semplice applet contenente delle scritte. La si inserisca nella pagina web usando un file jar.

Capitolo 6 Grafica, font e colori

6.1 disegnare una scacchiera sopra un'applet. Il numero di righe e colonne ed i colori vengono letti come parametri dell'applet.

Capitolo 7 Animazioni, immagini e suoni

7.1 Si costruisca un'applet che contiene un'immagine che si muove casualmente sullo schermo.

7.2 Si modifichi l'esempio Neko per fargli percorrere tutto lo schermo.

Capitolo 8 Eventi del mouse e della tastiera

8.1 Si costruisca un'applet che contiene un'immagine che si muove casualmente sullo schermo, seguendo però i movimenti del mouse. Si definisca l'applet come sensore.

8.2 Si costruisca un'applet che contiene un'immagine che si muove casualmente sullo schermo, seguendo però i movimenti del mouse. Si definisca un sensore esterno.

8.3 Si costruisca un'applet che contiene un'immagine che si muove casualmente sullo schermo, seguendo però i click del mouse. Si definisca l'applet come sensore.

8.4 Si costruisca un'applet che contiene un'immagine che si muove casualmente sullo schermo, seguendo però i click del mouse. Si definisca un sensore esterno.

8.5 Si costruisca un'applet che contiene una etichetta. L'applet conta il numero di tasti premuti e lo visualizza sull'etichetta quando si preme invio. Si definisca l'applet come sensore.

8.6 Si costruisca un'applet che contiene una etichetta. L'applet conta il numero di tasti premuti e lo visualizza sull'etichetta quando si preme invio. Si definisca un sensore esterno.

8.7 Si costruisca un'applet che rappresenta lo sportello di un frigo e le scritte magnetiche sopra di esso. Con il mouse si spostano le scritte.

9 Interfaccia AWT

9.1 Si definisca una applicazione grafica costituita da una finestra che contiene (in questo ordine): due campi di testo e un bottone

cliccando sul bottone deve avvenire:

se uno dei due campi di testo è vuoto il loro contenuto viene scambiato

9.2 Si definisca una applicazione grafica costituita da una finestra che contiene (in questo ordine): Un bottone chiamato "Sinistra", una label, un bottone chiamato "Destra"

cliccando sul bottone deve avvenire:

sulla label compare il nome del bottone premuto

9.3 Si definisca una applicazione grafica costituita da una finestra che contiene (in questo ordine): un pannello, due campi di testo e un bottone

cliccando sul bottone deve avvenire:

- si confrontano alfabeticamente le stringhe contenute nei 2 campi di testo e si colora il pannello di rosso se la prima è minore, di giallo se sono uguali di verde se la seconda è minore

Se uno dei campi di testo è vuoto non avverrà nulla

9.4 Si definisca una applicazione grafica costituita da una finestra che contiene un pannello e sotto un
bottone. Inizialmente il pannello deve essere colorato di verde.
cliccando sul bottone deve avvenire:
 se il pannello è verde deve diventare giallo, se è giallo deve diventare rosso, se è rosso deve diventare verde.

9.5 Si definisca una applicazione grafica costituita da una finestra che contiene (in questo ordine): un campo di testo, un bottone e una label.

cliccando sul bottone deve avvenire:

- Nella label compare la scritta: "La stringa è lunga: " seguita dalla lunghezza della stringa contenuta nel campo di testo.

Se il campo di testo è vuoto non avverrà nulla

9.6 Si definisca una applicazione grafica costituita da una finestra che contiene (in questo ordine): Un bottone chiamato "Conta", e una label.

cliccando sul bottone deve avvenire:
sulla label compare il numero di volte che il bottone è stato premuto

9.7 Si definisca una applicazione grafica costituita da una finestra che contiene (in questo ordine): Un bittone chiamato "Sinistra", una label, un bottone chiamato "Destra"

cliccando sul bottone deve avvenire:
sulla label compare il nome del bottone premuto

Capitolo 10 Modificatori

Incapsulazione

Riconsiderare gli esercizi dei capitoli precedenti (in particolare i capitoli 3 e 4) definendo gli opportuni modificatori di accesso per campi e metodi ed introducendo nuovi metodi di accesso se necessario.

11 Package, interfacce e classi interne

11.1 Definire un package per un esercizio sul polimorfismo del capitolo 4.
Si definisca un package contenente la classe principale, un sottopackage per le sottoclassi ed un diverso package per l'applicazione principale. Si definiscano gli opportuni modificatori di accesso.

11.2 Completare l'esempio visto a lezione:
definire una interfaccia che rappresenta i comportamenti del cibo: profumare e nutrire
definire una interfaccia che rappresenta i comportamenti di una sfera: ruotare e rimbalzare
definire una interfaccia che rappresenta i comportamenti di un frutto: profumare e maturare

Definire le classi:

- frutto
- arancia
- banana
- pneumatico

definendo opportunamente i rapporti di ereditarietà e le interfaccie da implementare
Si definisca una applicazione principale che usa oggetti delle classi definite;
nell'applicazione definire un oggetto di tipo interfaccia sfera ed assegnargli tutti i valori possibili.

Capitolo 12 Eccezioni

12.1 Si definisca una applicazione grafica costituita da una finestra che contiene (in questo ordine):

- un bottone
- un campo di testo (finalizzato a contenere numeri interi)

cliccando sul bottone il numero presente nel campo di testo deve essere incrementato di uno.
Nel caso in cui si clicchi sul bottone quando il campi di testo non contiene un numero, nel campo di testo deve apparire il valore 0; si gestisca questa situazione tramite una eccezione.

12.2 Si definisca una applicazione grafica costituita da una finestra che contiene (in questo ordine):

- un bottone, una label (finalizzata a contenere numeri ed inizialmente contenente 0), un campo di testo

cliccando sul bottone la label viene incrementata del valore contenuto nel campo di testo; si gestisca tramite il meccanismo delle Eccezioni il caso in cui il campo di testo non contiene un valore intero: in questo caso la label deve venire azzerata.

Capitolo 13 Multithreading

13.1 si definisca un Thread che stampa a video numeri interi compresi in un intervallo.
Si definisca l'applicazione principale che contiene un metodo costruttore che dato un parametro intero n genera e lancia n Thread però operanti su intervalli diversi.
Inoltre deve essere possibile sospendere e far ripartire i Thread.

Vedi anche esercizi capitolo 15

Capitolo 14 Flussi di input e di output

14.1 Si definisca una lista concatenata tramite puntatori a partire dalle classi:
Elemento, che contiene i campi: *valore* (di tipo intero) e *successivo* (di classe *Elemento*);
Lista, che contiene il campo *testa* (di classe *Elemento*), che rappresenta il riferimento al primo elemento della lista.

Nella classe *Lista* si definisca:

- il costruttore che dato un intero n genera una lista di n elementi casuali.
- il metodo *scriviFile* che, dati come parametro una stringa che rappresenta il nome di un file di caratteri ed un intero N, scriva nel file gli elementi della lista più piccoli di N (un valore per ogni riga) e nell'ultima riga del file la somma dei valori presenti nelle righe precedenti del file.

Si realizzi infine una applicazione che: costruisce un oggetto di classe lista e richiama il metodo scriviFile.

14.2 Si definisca una applicazione grafica costituita da una finestra che contiene (in questo ordine): un campo di testo e un bottone

cliccando sul bottone deve avvenire:

- viene aperto il file (come file di testo) il cui nome è contenuto nel campo di testo.
- Si scrive nel file la frase:

  "Il nome di questo file e' lungo: " seguita dalla lunghezza del nome del file

Se il campo di testo è vuoto non avverrà nulla

14.3 Si definisca una lista concatenata tramite puntatori a partire dalle classi:
Elemento, che contiene i campi: *valore* (di tipo intero) e *successivo* (di classe *Elemento*);
Lista, che contiene il campo *testa* (di classe *Elemento*), che rappresenta il riferimento al primo elemento della lista.
Nella classe *Lista* si definisca:

- il costruttore che dato un intero n genera una lista di n elementi casuali.
- il metodo *scriviFile* che, data come parametro una stringa che rappresenta il nome di un file di caratteri, scriva nel file il contenuto della lista (un valore per ogni riga) e la media degli elementi presenti nella lista.

Si realizzi infine una applicazione che: costruisce un oggetto di classe lista e richiama il metodo scriviFile.

14.4 Si definisca una lista concatenata tramite puntatori a partire dalle classi:
Elemento, che contiene i campi: *valore* (di tipo intero) e *successivo* (di classe *Elemento*);
Lista, che contiene il campo *testa* (di classe *Elemento*), che rappresenta il riferimento al primo elemento della lista.
Nella classe *Lista* si definisca:

- il costruttore che dato un intero n genera una lista di n elementi casuali.
- il metodo *scriviFile* che, data come parametro una stringa che rappresenta il nome di un file di caratteri, scriva nel file gli elementi pari della lista (un valore per ogni riga) e nell'ultima riga del file il numero di elementi pari presenti nella lista.

Si realizzi infine una applicazione che: costruisce un oggetto di classe lista e richiama il metodo scriviFile.

14.5 Si definisca una applicazione che contiene un metodo che dato il nome di un file ed un numero intero *n* restituisce il numero di volte che *n* è presente nel file.
Si supponga che il file sia un file di caratteri contenente un numero intero per ogni riga.
Nel metodo main dell'applicazione si richiami il metodo per cercare quante volte il numero 10 compare nel file "esame.txt" e si trascriva a video il risultato.

Si definisca una lista concatenata tramite puntatori a partire dalle classi:
Elemento, che contiene i campi: *valore* (di tipo intero) e *successivo* (di classe *Elemento*);
Lista, che contiene il campo *testa* (di classe *Elemento*), che rappresenta il riferimento al primo

elemento della lista.

Nella classe *Lista* si definisca:

- il costruttore che dato un intero n genera una lista di n elementi casuali.
- il metodo *scriviFile* che, dati come parametro una stringa che rappresenta il nome di un file di caratteri ed un intero N, scriva nel file gli elementi della lista più piccoli di N (un valore per ogni riga) e nell'ultima riga del file la somma dei valori presenti nelle righe precedenti del file.

Si realizzi infine una applicazione che: costruisce un oggetto di classe lista e richiama il metodo scriviFile.

Capitolo 15 Programmazione di rete

15.1 Si definisca una applicazione client server basata su socket. Il server, utilizzando i thread, deve poter rispondere contemporaneamente a piu' client.

Il server ha un campo n, il cui valore viene assegnato nel costruttore.
Il server manda ai client i valori da 1 a n.
Il server segnala il termine della trasmissione dei valori mandando il messagio "BYE".

Il client deve restituire la media dei valori mandati dal server.

15.2 Si definisca una applicazione client server basata su socket. Il server, utilizzando i thread, deve poter rispondere contemporaneamente a piu' client.

Il server ha un campo nome (una stringa), il cui valore viene assegnato nel costruttore.
Il server apre il file che si chiama nome e che supponiamo contenga un intero per ogni riga.
Il server invia ai client i valori contenuti nel file.

Il server segnala il termine della trasmissione dei valori mandando il messagio "BYE".

Il client deve restituire la somma dei valori mandati dal server.

15.3 Si definisca una applicazione client server basata su socket. Il server, utilizzando i thread, deve poter rispondere contemporaneamente a piu' client.

Il server ha un campo N (un intero), il cui valore viene assegnato nel costruttore.
Il server invia ai client N valori interi positivi minori di 1000 generati casualmente.
Il server segnala il termine della trasmissione dei valori mandando il messagio "BYE".

Il client deve stampare a video i valori ricevuti e restituire il massimo dei valori ricevuti.

15.4 Si definisca una applicazione client server basata su socket. Il server, utilizzando i thread, deve poter rispondere contemporaneamente a piu' client.

Ogni client invia al server un numero generato casualmente.
Il server stampa a video il numero arrivato e la somma di tutti i numeri inviati dai client.

Se necessario, si affrontino le tematiche di sincronizzazione.