**import** java.io.\*;

**import** java.util.Random;

/\*\*

\* Applicazione per la creazione di parole chiave.

\* **@author** Abramo Gerardo Giordano.

\* **@version** 1.3 (16 Aprile 2019).

\*/

**public** **class** **GeneratoreParoleChiave** {

**static** **String** **parolaChiave** = **new** String();

**static** **Random** **caso** = **new** Random();

**static** **char** **elencoCaratteri**[];

**static** **boolean** **condizioneCicloDoWhile**;

**static** **boolean** **trovataCifraNumerica**;

**static** **boolean** **trovataLetteraMinuscola**;

**static** **boolean** **trovataLetteraMaiuscola**;

**static** **boolean** **trovatoCarattereSpeciale**;

**public** **static** **void** main(**String** args[]) **throws** **IOException** {

**String** stringaInput = **new** String();

**String** percorsoNomeFile = **new** String("ParoleChiave.txt");

**InputStreamReader** lettoreFlussoIngresso = **new** InputStreamReader(**System**.***in***);

**BufferedReader** tastiera = **new** BufferedReader(lettoreFlussoIngresso);

**FileWriter** flussoScritturafileTXTParoleChiave = **new** FileWriter(percorsoNomeFile, **true**);

**BufferedWriter** fileTXTParoleChiave = **new** BufferedWriter(flussoScritturafileTXTParoleChiave);

**String** elencoParoleChiave[];

**final** **int** numeroMinimoParoleChiave = 1; // Numero minimo di parole chiave che il programma può generare.

**final** **int** numeroMassimoParoleChiave = 1000; // Numero massimo di parole chiave che il programma puo generare.

**final** **int** lunghezzaMinimaParoleChiave = 10; // Lunghezza minima (in caratteri) di ogni parola chiave che il programma può generare.

**final** **int** lunghezzaMassimaParoleChiave = 1000; // Lunghezza massima (in caratteri) di ogni parola chiave che il programma può generare.

**int** numeroParoleChiave = 0;

**int** lunghezzaParoleChiave = 0;

**char** sceltaUtenteChar = '\_';

**boolean** condizioneCicloDoWhile = **false**; // Variabile utilizzata esclusivamente nei cicli DO... WHILE per la verifica dei dati d'ingresso.

**boolean** condizioneNuovaElaborazioneProgramma = **false**; // Variabile utilizzata per l'esecuzione di una nuova elaborazione del programma senza riavviarlo.

**boolean** condizioneUscitaProgramma = **false**; // Variabile utilizzata per terminare l'esecuzione del programma.

**boolean** condizioneRiavvioProgramma = **false**; // Variabile utilizzata per il riavvio del programma.

*stampa*("\n\n\n\n");

// Punto di inizio per il riavvio del programma.

**do** {

condizioneRiavvioProgramma = **false**;

*stampa*("\n\tProgramma per la creazione di PAROLE CHIAVE.");

*stampa*("\n\tScrivere \"EE\" oppure \"ESCI\" oppure \"EXIT\" (in minuscolo o maiuscolo) + INVIO, per uscire dal programma.");

*stampa*("\n\tScrivere \"RR\" oppure \"RIAVVIA\" oppure \"RESTART\" (in minuscolo o maiuscolo) + INVIO, per riavviare il programma.");

*stampa*("\n");

// Punto di inizio per l'esecuzione di una nuova elaborazione del programma senza riavviarlo.

**do** {

condizioneNuovaElaborazioneProgramma = **false**;

condizioneUscitaProgramma = **false**;

**if** (condizioneUscitaProgramma==**false**) {

*stampa*("\tChe tipo di parola chiave vuoi creare?");

*stampa*("\n\tA) Numerica.");

*stampa*("\n\tB) Alfabetica.");

*stampa*("\n\tC) Alfabetica minuscola.");

*stampa*("\n\tD) Alfabetica maiuscola.");

*stampa*("\n\tE) Alfanumerica");

*stampa*("\n\tF) Alfanumerica minuscola.");

*stampa*("\n\tG) Alfanumerica maiuscola.");

*stampa*("\n\tH) Alfanumerica con caratteri speciali.");

*stampa*("\n\tI) Alfanumerica minuscola con caratteri speciali.");

*stampa*("\n\tL) Alfanumerica maiuscola con caratteri speciali.");

*stampa*("\n");

// Inizio del ciclo DO... WHILE per la verifica del valore inserito dall'utente per il tipo di parola chiave da generare.

**do** {

condizioneCicloDoWhile = **true**;

*stampa*("\tScegli un'opzione: ");

stringaInput = tastiera.readLine();

**if** (*esci*(stringaInput)==**true**) {

condizioneCicloDoWhile = **false**;

condizioneUscitaProgramma = **true**;

*stampa*("\tTERMINE DELL\'ESECUZIONE DEL PROGRAMMA.\n");

} **else** **if** (*riavvia*(stringaInput)==**true**) {

condizioneCicloDoWhile = **false**;

condizioneUscitaProgramma = **true**;

condizioneRiavvioProgramma = **true**;

*stampa*("\tRIAVVIO DEL PROGRAMMA IN CORSO ...\n");

} **else** **if** (*eliminaSpazi*(stringaInput).length()==0)

*stampa*("\tATTENZIONE: non hai scelto alcuna opzione.\n");

**else** {

stringaInput = *eliminaSpazi*(stringaInput);

**if** (stringaInput.length()>1)

*stampa*("\tATTENZIONE: hai scelto un\'opzione non valida (l\'opzione è formata da una sola lettera).\n");

**else** {

stringaInput = stringaInput.toUpperCase();

sceltaUtenteChar = stringaInput.charAt(0);

**if** ((sceltaUtenteChar!='A')&&(sceltaUtenteChar!='B')&&(sceltaUtenteChar!='C')&&(sceltaUtenteChar!='D')&&(sceltaUtenteChar!='E')&&(sceltaUtenteChar!='F')&&(sceltaUtenteChar!='G')&&(sceltaUtenteChar!='H')&&(sceltaUtenteChar!='I')&&(sceltaUtenteChar!='L'))

*stampa*("\tATTENZIONE: hai scelto un\'opzione non in elenco (le opzioni disponibili sono A, B, C, D, E, F, G, H, I e L).\n");

**else**

condizioneCicloDoWhile = **false**;

}

}

} **while** (condizioneCicloDoWhile==**true**);

}

**if** (condizioneUscitaProgramma==**false**) {

// Inizio del ciclo DO... WHILE per la verifica del valore inserito dall'utente relativo al numero di parole chiave da generare.

**do** {

condizioneCicloDoWhile = **true**;

*stampa*("\tQuante sono le parole chiave che vuoi generare? ");

stringaInput = tastiera.readLine();

**if** (*esci*(stringaInput)==**true**) {

condizioneCicloDoWhile = **false**;

condizioneUscitaProgramma = **true**;

*stampa*("\tTERMINE DELL\'ESECUZIONE DEL PROGRAMMA.\n");

} **else** **if** (*riavvia*(stringaInput)==**true**) {

condizioneCicloDoWhile = **false**;

condizioneUscitaProgramma = **true**;

condizioneRiavvioProgramma = **true**;

*stampa*("\tRIAVVIO DEL PROGRAMMA IN CORSO ...\n");

} **else** **if** (*eliminaSpazi*(stringaInput).length()==0)

*stampa*("\tATTENZIONE: non hai inserito alcun valore.\n");

**else** **if** (*trovaNumeroIntero*(stringaInput)==**false**)

*stampa*("\tATTENZIONE: non hai inserito un numero in un formato valido (devi inserire un numero INTERO POSITIVO o senza segno).\n");

**else** {

**try** {

stringaInput = *eliminaSpazi*(stringaInput);

numeroParoleChiave = **Integer**.*parseInt*(stringaInput);

**if** (numeroParoleChiave==0)

*stampa*("\tATTENZIONE: hai inserito un numero di parole chiave pari a 0.\n");

**else** **if** (numeroParoleChiave<0)

*stampa*("\tATTENZIONE: hai inserito un numero di parole chiave minore di 0.\n");

**else** **if** (numeroParoleChiave<numeroMinimoParoleChiave)

*stampa*("\tATTENZIONE: il numero minimo di parole chiave deve essere maggiore o uguale a " + numeroMinimoParoleChiave + ".\n");

**else** **if** (numeroParoleChiave>numeroMassimoParoleChiave)

*stampa*("\tATTENZIONE: questo programma può generare un numero massimo di " + numeroMassimoParoleChiave + " parole chiave.\n");

**else**

condizioneCicloDoWhile = **false**;

} **catch** (**NumberFormatException** eccezzione1) {

*stampa*("\tATTENZIONE: il numero inserito (in modulo) è troppo grande.\n");

}

}

} **while** (condizioneCicloDoWhile==**true**);

}

**if** (condizioneUscitaProgramma==**false**) {

// Inizio del ciclo DO... WHILE per la verifica del valore inserito dall'utente relativo alla lunghezza dele parole chiave.

**do** {

condizioneCicloDoWhile = **true**;

*stampa*("\tQuanti caratteri deve contenere ogni parola chiave? ");

stringaInput = tastiera.readLine();

**if** (*esci*(stringaInput)==**true**) {

condizioneCicloDoWhile = **false**;

condizioneUscitaProgramma = **true**;

*stampa*("\tTERMINE DELL\'ESECUZIONE DEL PROGRAMMA.");

} **else** **if** (*riavvia*(stringaInput)==**true**) {

condizioneCicloDoWhile = **false**;

condizioneUscitaProgramma = **true**;

condizioneRiavvioProgramma = **true**;

*stampa*("\tRIAVVIO DEL PROGRAMMA IN CORSO PROGRAMMA ...\n");

} **else** **if** (*eliminaSpazi*(stringaInput).length()==0)

*stampa*("\tATTENZIONE: non hai inserito alcun valore.\n");

**else** **if** (*trovaNumeroIntero*(stringaInput)==**false**)

*stampa*("\tATTENZIONE: non hai inserito un numero in un formato valido (devi inserire un numero INTERO POSITIVO o senza segno).\n");

**else** {

**try** {

stringaInput = *eliminaSpazi*(stringaInput);

lunghezzaParoleChiave = **Integer**.*parseInt*(stringaInput);

**if** (lunghezzaParoleChiave==0)

*stampa*("\tATTENZIONE: hai inserito un numero di caratteri pari a 0.\n");

**else** **if** (lunghezzaParoleChiave<0)

*stampa*("\tATTENZIONE: hai inserito un numero di caratteri minore di 0.\n");

**else** **if** (lunghezzaParoleChiave<lunghezzaMinimaParoleChiave)

*stampa*("\tATTENZIONE: la lunghezza minima delle parole chiave deve essere maggiore o uguale a " + lunghezzaMinimaParoleChiave + " caratteri.\n");

**else** **if** (lunghezzaParoleChiave>lunghezzaMassimaParoleChiave)

*stampa*("\tATTENZIONE: questo programma può generare parole chiave lunghe al massimo " + lunghezzaMassimaParoleChiave + " caratteri.\n");

**else**

condizioneCicloDoWhile = **false**;

} **catch** (**NumberFormatException** eccezzione1) {

*stampa*("\tATTENZIONE: il numero inserito (in modulo) è troppo grande.\n");

}

}

} **while** (condizioneCicloDoWhile==**true**);

}

**if** (condizioneUscitaProgramma==**false**) {

**switch** (sceltaUtenteChar) {

**case** 'A': fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

elencoParoleChiave = **new** **String**[numeroParoleChiave];

**for** (**int** i = 0; i<numeroParoleChiave; i++) {

elencoParoleChiave[i] = *creaParolaChiaveNumerica*(lunghezzaParoleChiave);

**for** (**int** j = 0; j<=i; j++) {

**if** (j==i)

**continue**;

**else** **if** (elencoParoleChiave[i].equals(elencoParoleChiave[j])) {

i--;

**break**;

}

}

}

**for** (**int** i = 0; i<elencoParoleChiave.length; i++) {

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.write("Chiave numerica " + elencoParoleChiave[i]);

*stampa*("\n\t" + elencoParoleChiave[i]);

}

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

*stampa*("\n\n");

**break**;

**case** 'B': fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

elencoParoleChiave = **new** **String**[numeroParoleChiave];

**for** (**int** i = 0; i<numeroParoleChiave; i++) {

elencoParoleChiave[i] = *creaParolaChiaveAlfabetica*(lunghezzaParoleChiave);

**for** (**int** j = 0; j<=i; j++) {

**if** (j==i)

**continue**;

**else** **if** (elencoParoleChiave[i].equals(elencoParoleChiave[j])) {

i--;

**break**;

}

}

}

**for** (**int** i = 0; i<elencoParoleChiave.length; i++) {

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.write("Chiave alfabetica " + elencoParoleChiave[i]);

*stampa*("\n\t" + elencoParoleChiave[i]);

}

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

*stampa*("\n\n");

**break**;

**case** 'C': fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

elencoParoleChiave = **new** **String**[numeroParoleChiave];

**for** (**int** i = 0; i<numeroParoleChiave; i++) {

elencoParoleChiave[i] = *creaParolaChiaveAlfabeticaMinuscola*(lunghezzaParoleChiave);

**for** (**int** j = 0; j<=i; j++) {

**if** (j==i)

**continue**;

**else** **if** (elencoParoleChiave[i].equals(elencoParoleChiave[j])) {

i--;

**break**;

}

}

}

**for** (**int** i = 0; i<elencoParoleChiave.length; i++) {

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.write("Chiave alfabetica minuscola " + elencoParoleChiave[i]);

*stampa*("\n\t" + elencoParoleChiave[i]);

}

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

*stampa*("\n\n");

**break**;

**case** 'D': fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

elencoParoleChiave = **new** **String**[numeroParoleChiave];

**for** (**int** i = 0; i<numeroParoleChiave; i++) {

elencoParoleChiave[i] = *creaParolaChiaveAlfabeticaMaiuscola*(lunghezzaParoleChiave);

**for** (**int** j = 0; j<=i; j++) {

**if** (j==i)

**continue**;

**else** **if** (elencoParoleChiave[i].equals(elencoParoleChiave[j])) {

i--;

**break**;

}

}

}

**for** (**int** i = 0; i<elencoParoleChiave.length; i++) {

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.write("Chiave alfabetica maiuscola " + elencoParoleChiave[i]);

*stampa*("\n\t" + elencoParoleChiave[i]);

}

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

*stampa*("\n\n");

**break**;

**case** 'E': fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

elencoParoleChiave = **new** **String**[numeroParoleChiave];

**for** (**int** i = 0; i<numeroParoleChiave; i++) {

elencoParoleChiave[i] = *creaParolaChiaveAlfanumerica*(lunghezzaParoleChiave);

**for** (**int** j = 0; j<=i; j++) {

**if** (j==i)

**continue**;

**else** **if** (elencoParoleChiave[i].equals(elencoParoleChiave[j])) {

i--;

**break**;

}

}

}

**for** (**int** i = 0; i<elencoParoleChiave.length; i++) {

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.write("Chiave alfanumerica " + elencoParoleChiave[i]);

*stampa*("\n\t" + elencoParoleChiave[i]);

}

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

*stampa*("\n\n");

**break**;

**case** 'F': fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

elencoParoleChiave = **new** **String**[numeroParoleChiave];

**for** (**int** i = 0; i<numeroParoleChiave; i++) {

elencoParoleChiave[i] = *creaParolaChiaveAlfanumericaMinuscola*(lunghezzaParoleChiave);

**for** (**int** j = 0; j<=i; j++) {

**if** (j==i)

**continue**;

**else** **if** (elencoParoleChiave[i].equals(elencoParoleChiave[j])) {

i--;

**break**;

}

}

}

**for** (**int** i = 0; i<elencoParoleChiave.length; i++) {

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.write("Chiave alfanumerica minuscola " + elencoParoleChiave[i]);

*stampa*("\n\t" + elencoParoleChiave[i]);

}

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

*stampa*("\n\n");

**break**;

**case** 'G': fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

elencoParoleChiave = **new** **String**[numeroParoleChiave];

**for** (**int** i = 0; i<numeroParoleChiave; i++) {

elencoParoleChiave[i] = *creaParolaChiaveAlfanumericaMaiuscola*(lunghezzaParoleChiave);

**for** (**int** j = 0; j<=i; j++) {

**if** (j==i)

**continue**;

**else** **if** (elencoParoleChiave[i].equals(elencoParoleChiave[j])) {

i--;

**break**;

}

}

}

**for** (**int** i = 0; i<elencoParoleChiave.length; i++) {

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.write("Chiave alfanumerica maiuscola " + elencoParoleChiave[i]);

*stampa*("\n\t" + elencoParoleChiave[i]);

}

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

*stampa*("\n\n");

**break**;

**case** 'H': fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

elencoParoleChiave = **new** **String**[numeroParoleChiave];

**for** (**int** i = 0; i<numeroParoleChiave; i++) {

elencoParoleChiave[i] = *creaParolaChiaveAlfanumericaCaratteriSpeciali*(lunghezzaParoleChiave);

**for** (**int** j = 0; j<=i; j++) {

**if** (j==i)

**continue**;

**else** **if** (elencoParoleChiave[i].equals(elencoParoleChiave[j])) {

i--;

**break**;

}

}

}

**for** (**int** i = 0; i<elencoParoleChiave.length; i++) {

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.write("Chiave alfanumerica con caratteri speciali " + elencoParoleChiave[i]);

*stampa*("\n\t" + elencoParoleChiave[i]);

}

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

*stampa*("\n\n");

**break**;

**case** 'I': fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

elencoParoleChiave = **new** **String**[numeroParoleChiave];

**for** (**int** i = 0; i<numeroParoleChiave; i++) {

elencoParoleChiave[i] = *creaParolaChiaveAlfanumericaMinuscolaCaratteriSpeciali*(lunghezzaParoleChiave);

**for** (**int** j = 0; j<=i; j++) {

**if** (j==i)

**continue**;

**else** **if** (elencoParoleChiave[i].equals(elencoParoleChiave[j])) {

i--;

**break**;

}

}

}

**for** (**int** i = 0; i<elencoParoleChiave.length; i++) {

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.write("Chiave alfanumerica minuscola con caratteri speciali " + elencoParoleChiave[i]);

*stampa*("\n\t" + elencoParoleChiave[i]);

}

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

*stampa*("\n\n");

**break**;

**case** 'L': fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

elencoParoleChiave = **new** **String**[numeroParoleChiave];

**for** (**int** i = 0; i<numeroParoleChiave; i++) {

elencoParoleChiave[i] = *creaParolaChiaveAlfanumericaMaiuscolaCaratteriSpeciali*(lunghezzaParoleChiave);

**for** (**int** j = 0; j<=i; j++) {

**if** (j==i)

**continue**;

**else** **if** (elencoParoleChiave[i].equals(elencoParoleChiave[j])) {

i--;

**break**;

}

}

}

**for** (**int** i = 0; i<elencoParoleChiave.length; i++) {

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.write("Chiave alfanumerica maiuscola con caratteri speciali " + elencoParoleChiave[i]);

*stampa*("\n\t" + elencoParoleChiave[i]);

}

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

fileTXTParoleChiave.newLine();

*stampa*("\n\n");

**break**;

}

}

**if** (condizioneUscitaProgramma==**false**) {

// Inizio del ciclo DO... WHILE per la verifica dell'opzione scelta dall'utente relativa alla richiesta del programma di eseguire una nuova elaborazione.

**do** {

condizioneCicloDoWhile = **true**;

*stampa*("\tVuoi creare altre parole chiave (S) o (N)? ");

stringaInput = tastiera.readLine();

**if** (*esci*(stringaInput)==**true**) {

condizioneCicloDoWhile = **false**;

// condizioneUscitaProgramma = true;

*stampa*("\tTERMINE DELL\'ESECUZIONE DEL PROGRAMMA.\n");

} **else** **if** (*riavvia*(stringaInput)==**true**) {

condizioneCicloDoWhile = **false**;

// condizioneUscitaProgramma = true;

condizioneRiavvioProgramma = **true**;

*stampa*("\tRIAVVIO DEL PROGRAMMA IN CORSO ...\n");

} **else** **if** (*eliminaSpazi*(stringaInput).length()==0)

*stampa*("\tATTENZIONE: non hai scelto alcuna opzione.\n");

**else** {

stringaInput = *eliminaSpazi*(stringaInput);

**if** (stringaInput.length()>1)

*stampa*("\tATTENZIONE: hai scelto un\'opzione non valida (l\'opzione è formata da una sola lettera).\n");

**else** {

stringaInput = stringaInput.toUpperCase();

sceltaUtenteChar = stringaInput.charAt(0);

**if** ((sceltaUtenteChar!='S')&&(sceltaUtenteChar!='N'))

*stampa*("\tATTENZIONE: hai scelto un\'opzione non in elenco (le opzioni disponibili sono S e N).\n");

**else** **if** (sceltaUtenteChar=='S') {

condizioneCicloDoWhile = **false**;

condizioneNuovaElaborazioneProgramma = **true**;

}

**else**

condizioneCicloDoWhile = **false**;

}

}

} **while** (condizioneCicloDoWhile==**true**);

}

} **while** (condizioneNuovaElaborazioneProgramma==**true**);

} **while** (condizioneRiavvioProgramma==**true**);

fileTXTParoleChiave.close();

*stampa*("\n\n\n\n");

}

// Metodo pubblico statico per la creazione di una parola chiave numerica.

**public** **static** **String** creaParolaChiaveNumerica(**int** lunghezzaParolaChiave) {

**elencoCaratteri** = **new** **char**[10];

**parolaChiave** = "";

**for** (**int** i = 0; i<10; i++)

**elencoCaratteri**[i] = (**char**) (48+i);

**for** (**int** i = 0; i<lunghezzaParolaChiave; i++)

**parolaChiave** = **parolaChiave** + **elencoCaratteri**[**caso**.nextInt(10)];

**return** **parolaChiave**;

}

// Metodo pubblico statico per la creazione di una parola chiave alfabetica.

**public** **static** **String** creaParolaChiaveAlfabetica(**int** lunghezzaParolaChiave) {

**elencoCaratteri** = **new** **char**[52];

**parolaChiave** = "";

**for** (**int** i = 0; i<26; i++) {

**elencoCaratteri**[i] = (**char**) (65+i);

**elencoCaratteri**[i+26] = (**char**) (97+i);

}

**for** (**int** i = 0; i<lunghezzaParolaChiave; i++)

**parolaChiave** = **parolaChiave** + **elencoCaratteri**[**caso**.nextInt(52)];

**return** **parolaChiave**;

}

// Metodo pubblico statico per la creazione di una parola chiave alfabetica minuscola.

**public** **static** **String** creaParolaChiaveAlfabeticaMinuscola(**int** lunghezzaParolaChiave) {

**elencoCaratteri** = **new** **char**[26];

**parolaChiave** = "";

**for** (**int** i = 0; i<26; i++) {

**elencoCaratteri**[i] = (**char**) (97+i);

}

**for** (**int** i = 0; i<lunghezzaParolaChiave; i++)

**parolaChiave** = **parolaChiave** + **elencoCaratteri**[**caso**.nextInt(26)];

**return** **parolaChiave**;

}

// Metodo pubblico statico per la creazione di una parola chiave alfabetica maiuscola.

**public** **static** **String** creaParolaChiaveAlfabeticaMaiuscola(**int** lunghezzaParolaChiave) {

**elencoCaratteri** = **new** **char**[26];

**parolaChiave** = "";

**for** (**int** i = 0; i<26; i++)

**elencoCaratteri**[i] = (**char**) (65+i);

**for** (**int** i = 0; i<lunghezzaParolaChiave; i++)

**parolaChiave** = **parolaChiave** + **elencoCaratteri**[**caso**.nextInt(26)];

**return** **parolaChiave**;

}

// Metodo pubblico statico per la creazione di una parola chiave alfanumerica.

**public** **static** **String** creaParolaChiaveAlfanumerica(**int** lunghezzaParolaChiave) {

**elencoCaratteri** = **new** **char**[62];

**for** (**int** i = 0; i<26; i++) {

**elencoCaratteri**[i] = (**char**) (65+i);

**elencoCaratteri**[i+26] = (**char**) (97+i);

}

**for** (**int** i = 0; i<10; i++)

**elencoCaratteri**[i+52] = (**char**) (48+i);

**do** {

**parolaChiave** = "";

**condizioneCicloDoWhile** = **true**;

**trovataCifraNumerica** = **false**;

**trovataLetteraMinuscola** = **false**;

**trovataLetteraMaiuscola** = **false**;

**for** (**int** i = 0; i<lunghezzaParolaChiave; i++) {

**parolaChiave** = **parolaChiave** + **elencoCaratteri**[**caso**.nextInt(62)];

**if** ((**condizioneCicloDoWhile**==**true**)&&(lunghezzaParolaChiave>=3)) {

**if** ((**trovataCifraNumerica**==**false**)&&((**parolaChiave**.charAt(i)=='0')||(**parolaChiave**.charAt(i)=='1')||(**parolaChiave**.charAt(i)=='2')||(**parolaChiave**.charAt(i)=='3')||(**parolaChiave**.charAt(i)=='4')||(**parolaChiave**.charAt(i)=='5')||(**parolaChiave**.charAt(i)=='6')||(**parolaChiave**.charAt(i)=='7')||(**parolaChiave**.charAt(i)=='8')||(**parolaChiave**.charAt(i)=='9')))

**trovataCifraNumerica** = **true**;

**if** ((**trovataLetteraMinuscola**==**false**)&&((**parolaChiave**.charAt(i)=='a')||(**parolaChiave**.charAt(i)=='b')||(**parolaChiave**.charAt(i)=='c')||(**parolaChiave**.charAt(i)=='d')||(**parolaChiave**.charAt(i)=='e')||(**parolaChiave**.charAt(i)=='f')||(**parolaChiave**.charAt(i)=='g')||(**parolaChiave**.charAt(i)=='h')||(**parolaChiave**.charAt(i)=='i')||(**parolaChiave**.charAt(i)=='j')||(**parolaChiave**.charAt(i)=='k')||(**parolaChiave**.charAt(i)=='l')||(**parolaChiave**.charAt(i)=='m')||(**parolaChiave**.charAt(i)=='n')||(**parolaChiave**.charAt(i)=='o')||(**parolaChiave**.charAt(i)=='p')||(**parolaChiave**.charAt(i)=='q')||(**parolaChiave**.charAt(i)=='r')||(**parolaChiave**.charAt(i)=='s')||(**parolaChiave**.charAt(i)=='t')||(**parolaChiave**.charAt(i)=='u')||(**parolaChiave**.charAt(i)=='v')||(**parolaChiave**.charAt(i)=='w')||(**parolaChiave**.charAt(i)=='x')||(**parolaChiave**.charAt(i)=='y')||(**parolaChiave**.charAt(i)=='z')))

**trovataLetteraMinuscola** = **true**;

**if** ((**trovataLetteraMaiuscola**==**false**)&&((**parolaChiave**.charAt(i)=='A')||(**parolaChiave**.charAt(i)=='B')||(**parolaChiave**.charAt(i)=='C')||(**parolaChiave**.charAt(i)=='D')||(**parolaChiave**.charAt(i)=='E')||(**parolaChiave**.charAt(i)=='F')||(**parolaChiave**.charAt(i)=='G')||(**parolaChiave**.charAt(i)=='H')||(**parolaChiave**.charAt(i)=='I')||(**parolaChiave**.charAt(i)=='J')||(**parolaChiave**.charAt(i)=='K')||(**parolaChiave**.charAt(i)=='L')||(**parolaChiave**.charAt(i)=='M')||(**parolaChiave**.charAt(i)=='N')||(**parolaChiave**.charAt(i)=='O')||(**parolaChiave**.charAt(i)=='P')||(**parolaChiave**.charAt(i)=='Q')||(**parolaChiave**.charAt(i)=='R')||(**parolaChiave**.charAt(i)=='S')||(**parolaChiave**.charAt(i)=='T')||(**parolaChiave**.charAt(i)=='U')||(**parolaChiave**.charAt(i)=='V')||(**parolaChiave**.charAt(i)=='W')||(**parolaChiave**.charAt(i)=='X')||(**parolaChiave**.charAt(i)=='Y')||(**parolaChiave**.charAt(i)=='Z')))

**trovataLetteraMaiuscola** = **true**;

**if** ((**trovataCifraNumerica**==**true**)&&(**trovataLetteraMinuscola**==**true**)&&(**trovataLetteraMaiuscola**==**true**))

**condizioneCicloDoWhile** = **false**;

}

}

} **while** (**condizioneCicloDoWhile**==**true**);

**return** **parolaChiave**;

}

// Metodo pubblico statico per la creazione di una parola chiave alfanumerica minuscola.

**public** **static** **String** creaParolaChiaveAlfanumericaMinuscola(**int** lunghezzaParolaChiave) {

**elencoCaratteri** = **new** **char**[36];

**for** (**int** i = 0; i<26; i++) {

**elencoCaratteri**[i] = (**char**) (97+i);

}

**for** (**int** i = 0; i<10; i++)

**elencoCaratteri**[i+26] = (**char**) (48+i);

**do** {

**parolaChiave** = "";

**condizioneCicloDoWhile** = **true**;

**trovataCifraNumerica** = **false**;

**trovataLetteraMinuscola** = **false**;

**for** (**int** i = 0; i<lunghezzaParolaChiave; i++) {

**parolaChiave** = **parolaChiave** + **elencoCaratteri**[**caso**.nextInt(36)];

**if** ((**condizioneCicloDoWhile**==**true**)&&(lunghezzaParolaChiave>=2)) {

**if** ((**trovataCifraNumerica**==**false**)&&((**parolaChiave**.charAt(i)=='0')||(**parolaChiave**.charAt(i)=='1')||(**parolaChiave**.charAt(i)=='2')||(**parolaChiave**.charAt(i)=='3')||(**parolaChiave**.charAt(i)=='4')||(**parolaChiave**.charAt(i)=='5')||(**parolaChiave**.charAt(i)=='6')||(**parolaChiave**.charAt(i)=='7')||(**parolaChiave**.charAt(i)=='8')||(**parolaChiave**.charAt(i)=='9')))

**trovataCifraNumerica** = **true**;

**if** ((**trovataLetteraMinuscola**==**false**)&&((**parolaChiave**.charAt(i)=='a')||(**parolaChiave**.charAt(i)=='b')||(**parolaChiave**.charAt(i)=='c')||(**parolaChiave**.charAt(i)=='d')||(**parolaChiave**.charAt(i)=='e')||(**parolaChiave**.charAt(i)=='f')||(**parolaChiave**.charAt(i)=='g')||(**parolaChiave**.charAt(i)=='h')||(**parolaChiave**.charAt(i)=='i')||(**parolaChiave**.charAt(i)=='j')||(**parolaChiave**.charAt(i)=='k')||(**parolaChiave**.charAt(i)=='l')||(**parolaChiave**.charAt(i)=='m')||(**parolaChiave**.charAt(i)=='n')||(**parolaChiave**.charAt(i)=='o')||(**parolaChiave**.charAt(i)=='p')||(**parolaChiave**.charAt(i)=='q')||(**parolaChiave**.charAt(i)=='r')||(**parolaChiave**.charAt(i)=='s')||(**parolaChiave**.charAt(i)=='t')||(**parolaChiave**.charAt(i)=='u')||(**parolaChiave**.charAt(i)=='v')||(**parolaChiave**.charAt(i)=='w')||(**parolaChiave**.charAt(i)=='x')||(**parolaChiave**.charAt(i)=='y')||(**parolaChiave**.charAt(i)=='z')))

**trovataLetteraMinuscola** = **true**;

**if** ((**trovataCifraNumerica**==**true**)&&(**trovataLetteraMinuscola**==**true**))

**condizioneCicloDoWhile** = **false**;

}

}

} **while** (**condizioneCicloDoWhile**==**true**);

**return** **parolaChiave**;

}

// Metodo pubblico statico per la creazione di una parola chiave alfanumerica maiuscola.

**public** **static** **String** creaParolaChiaveAlfanumericaMaiuscola(**int** lunghezzaParolaChiave) {

**elencoCaratteri** = **new** **char**[36];

**for** (**int** i = 0; i<26; i++) {

**elencoCaratteri**[i] = (**char**) (65+i);

}

**for** (**int** i = 0; i<10; i++)

**elencoCaratteri**[i+26] = (**char**) (48+i);

**do** {

**parolaChiave** = "";

**condizioneCicloDoWhile** = **true**;

**trovataCifraNumerica** = **false**;

**trovataLetteraMaiuscola** = **false**;

**for** (**int** i = 0; i<lunghezzaParolaChiave; i++) {

**parolaChiave** = **parolaChiave** + **elencoCaratteri**[**caso**.nextInt(36)];

**if** ((**condizioneCicloDoWhile**==**true**)&&(lunghezzaParolaChiave>=2)) {

**if** ((**trovataCifraNumerica**==**false**)&&((**parolaChiave**.charAt(i)=='0')||(**parolaChiave**.charAt(i)=='1')||(**parolaChiave**.charAt(i)=='2')||(**parolaChiave**.charAt(i)=='3')||(**parolaChiave**.charAt(i)=='4')||(**parolaChiave**.charAt(i)=='5')||(**parolaChiave**.charAt(i)=='6')||(**parolaChiave**.charAt(i)=='7')||(**parolaChiave**.charAt(i)=='8')||(**parolaChiave**.charAt(i)=='9')))

**trovataCifraNumerica** = **true**;

**if** ((**trovataLetteraMaiuscola**==**false**)&&((**parolaChiave**.charAt(i)=='A')||(**parolaChiave**.charAt(i)=='B')||(**parolaChiave**.charAt(i)=='C')||(**parolaChiave**.charAt(i)=='D')||(**parolaChiave**.charAt(i)=='E')||(**parolaChiave**.charAt(i)=='F')||(**parolaChiave**.charAt(i)=='G')||(**parolaChiave**.charAt(i)=='H')||(**parolaChiave**.charAt(i)=='I')||(**parolaChiave**.charAt(i)=='J')||(**parolaChiave**.charAt(i)=='K')||(**parolaChiave**.charAt(i)=='L')||(**parolaChiave**.charAt(i)=='M')||(**parolaChiave**.charAt(i)=='N')||(**parolaChiave**.charAt(i)=='O')||(**parolaChiave**.charAt(i)=='P')||(**parolaChiave**.charAt(i)=='Q')||(**parolaChiave**.charAt(i)=='R')||(**parolaChiave**.charAt(i)=='S')||(**parolaChiave**.charAt(i)=='T')||(**parolaChiave**.charAt(i)=='U')||(**parolaChiave**.charAt(i)=='V')||(**parolaChiave**.charAt(i)=='W')||(**parolaChiave**.charAt(i)=='X')||(**parolaChiave**.charAt(i)=='Y')||(**parolaChiave**.charAt(i)=='Z')))

**trovataLetteraMaiuscola** = **true**;

**if** ((**trovataCifraNumerica**==**true**)&&(**trovataLetteraMaiuscola**==**true**))

**condizioneCicloDoWhile** = **false**;

}

}

} **while** (**condizioneCicloDoWhile**==**true**);

**return** **parolaChiave**;

}

// Metodo pubblico statico per la creazione di una parola chiave alfanumerica con caratteri speciali.

**public** **static** **String** creaParolaChiaveAlfanumericaCaratteriSpeciali(**int** lunghezzaParolaChiave) {

**elencoCaratteri** = **new** **char**[86];

**for** (**int** i = 0; i<26; i++) {

**elencoCaratteri**[i] = (**char**) (65+i);

**elencoCaratteri**[i+26] = (**char**) (97+i);

}

**for** (**int** i = 0; i<10; i++)

**elencoCaratteri**[i+52] = (**char**) (48+i);

**elencoCaratteri**[62] = '(';

**elencoCaratteri**[63] = ')';

**elencoCaratteri**[64] = '.';

**elencoCaratteri**[65] = ',';

**elencoCaratteri**[66] = ':';

**elencoCaratteri**[67] = ';';

**elencoCaratteri**[68] = '\_';

**elencoCaratteri**[69] = '=';

**elencoCaratteri**[70] = '€';

**elencoCaratteri**[71] = '£';

**elencoCaratteri**[72] = '$';

**elencoCaratteri**[73] = '%';

**elencoCaratteri**[74] = '&';

**elencoCaratteri**[75] = '?';

**elencoCaratteri**[76] = '\*';

**elencoCaratteri**[77] = '/';

**elencoCaratteri**[78] = '+';

**elencoCaratteri**[79] = '-';

**elencoCaratteri**[80] = '[';

**elencoCaratteri**[81] = ']';

**elencoCaratteri**[82] = '{';

**elencoCaratteri**[83] = '}';

**elencoCaratteri**[84] = '!';

**elencoCaratteri**[85] = '^';

**do** {

**parolaChiave** = "";

**condizioneCicloDoWhile** = **true**;

**trovataCifraNumerica** = **false**;

**trovataLetteraMinuscola** = **false**;

**trovataLetteraMaiuscola** = **false**;

**trovatoCarattereSpeciale** = **false**;

**for** (**int** i = 0; i<lunghezzaParolaChiave; i++) {

**parolaChiave** = **parolaChiave** + **elencoCaratteri**[**caso**.nextInt(86)];

**if** ((**condizioneCicloDoWhile**==**true**)&&(lunghezzaParolaChiave>=4)) {

**if** ((**trovataCifraNumerica**==**false**)&&((**parolaChiave**.charAt(i)=='0')||(**parolaChiave**.charAt(i)=='1')||(**parolaChiave**.charAt(i)=='2')||(**parolaChiave**.charAt(i)=='3')||(**parolaChiave**.charAt(i)=='4')||(**parolaChiave**.charAt(i)=='5')||(**parolaChiave**.charAt(i)=='6')||(**parolaChiave**.charAt(i)=='7')||(**parolaChiave**.charAt(i)=='8')||(**parolaChiave**.charAt(i)=='9')))

**trovataCifraNumerica** = **true**;

**if** ((**trovataLetteraMinuscola**==**false**)&&((**parolaChiave**.charAt(i)=='a')||(**parolaChiave**.charAt(i)=='b')||(**parolaChiave**.charAt(i)=='c')||(**parolaChiave**.charAt(i)=='d')||(**parolaChiave**.charAt(i)=='e')||(**parolaChiave**.charAt(i)=='f')||(**parolaChiave**.charAt(i)=='g')||(**parolaChiave**.charAt(i)=='h')||(**parolaChiave**.charAt(i)=='i')||(**parolaChiave**.charAt(i)=='j')||(**parolaChiave**.charAt(i)=='k')||(**parolaChiave**.charAt(i)=='l')||(**parolaChiave**.charAt(i)=='m')||(**parolaChiave**.charAt(i)=='n')||(**parolaChiave**.charAt(i)=='o')||(**parolaChiave**.charAt(i)=='p')||(**parolaChiave**.charAt(i)=='q')||(**parolaChiave**.charAt(i)=='r')||(**parolaChiave**.charAt(i)=='s')||(**parolaChiave**.charAt(i)=='t')||(**parolaChiave**.charAt(i)=='u')||(**parolaChiave**.charAt(i)=='v')||(**parolaChiave**.charAt(i)=='w')||(**parolaChiave**.charAt(i)=='x')||(**parolaChiave**.charAt(i)=='y')||(**parolaChiave**.charAt(i)=='z')))

**trovataLetteraMinuscola** = **true**;

**if** ((**trovataLetteraMaiuscola**==**false**)&&((**parolaChiave**.charAt(i)=='A')||(**parolaChiave**.charAt(i)=='B')||(**parolaChiave**.charAt(i)=='C')||(**parolaChiave**.charAt(i)=='D')||(**parolaChiave**.charAt(i)=='E')||(**parolaChiave**.charAt(i)=='F')||(**parolaChiave**.charAt(i)=='G')||(**parolaChiave**.charAt(i)=='H')||(**parolaChiave**.charAt(i)=='I')||(**parolaChiave**.charAt(i)=='J')||(**parolaChiave**.charAt(i)=='K')||(**parolaChiave**.charAt(i)=='L')||(**parolaChiave**.charAt(i)=='M')||(**parolaChiave**.charAt(i)=='N')||(**parolaChiave**.charAt(i)=='O')||(**parolaChiave**.charAt(i)=='P')||(**parolaChiave**.charAt(i)=='Q')||(**parolaChiave**.charAt(i)=='R')||(**parolaChiave**.charAt(i)=='S')||(**parolaChiave**.charAt(i)=='T')||(**parolaChiave**.charAt(i)=='U')||(**parolaChiave**.charAt(i)=='V')||(**parolaChiave**.charAt(i)=='W')||(**parolaChiave**.charAt(i)=='X')||(**parolaChiave**.charAt(i)=='Y')||(**parolaChiave**.charAt(i)=='Z')))

**trovataLetteraMaiuscola** = **true**;

**if** ((**trovatoCarattereSpeciale**==**false**)&&((**parolaChiave**.charAt(i)=='(')||(**parolaChiave**.charAt(i)==')')||(**parolaChiave**.charAt(i)=='.')||(**parolaChiave**.charAt(i)==',')||(**parolaChiave**.charAt(i)==':')||(**parolaChiave**.charAt(i)==';')||(**parolaChiave**.charAt(i)=='\_')||(**parolaChiave**.charAt(i)=='=')||(**parolaChiave**.charAt(i)=='€')||(**parolaChiave**.charAt(i)=='£')||(**parolaChiave**.charAt(i)=='$')||(**parolaChiave**.charAt(i)=='%')||(**parolaChiave**.charAt(i)=='&')||(**parolaChiave**.charAt(i)=='?')||(**parolaChiave**.charAt(i)=='\*')||(**parolaChiave**.charAt(i)=='/')||(**parolaChiave**.charAt(i)=='+')||(**parolaChiave**.charAt(i)=='-')||(**parolaChiave**.charAt(i)=='[')||(**parolaChiave**.charAt(i)==']')||(**parolaChiave**.charAt(i)=='{')||(**parolaChiave**.charAt(i)=='}')||(**parolaChiave**.charAt(i)=='!')||(**parolaChiave**.charAt(i)=='^')))

**trovatoCarattereSpeciale** = **true**;

**if** ((**trovataCifraNumerica**==**true**)&&(**trovataLetteraMinuscola**==**true**)&&(**trovataLetteraMaiuscola**==**true**)&&(**trovatoCarattereSpeciale**==**true**))

**condizioneCicloDoWhile** = **false**;

}

}

} **while** (**condizioneCicloDoWhile**==**true**);

**return** **parolaChiave**;

}

// Metodo pubblico statico per la creazione di una parola chiave alfanumerica minuscola con caratteri speciali.

**public** **static** **String** creaParolaChiaveAlfanumericaMinuscolaCaratteriSpeciali(**int** lunghezzaParolaChiave) {

**elencoCaratteri** = **new** **char**[60];

**for** (**int** i = 0; i<26; i++) {

**elencoCaratteri**[i] = (**char**) (97+i);

}

**for** (**int** i = 0; i<10; i++)

**elencoCaratteri**[i+26] = (**char**) (48+i);

**elencoCaratteri**[36] = '(';

**elencoCaratteri**[37] = ')';

**elencoCaratteri**[38] = '.';

**elencoCaratteri**[39] = ',';

**elencoCaratteri**[40] = ':';

**elencoCaratteri**[41] = ';';

**elencoCaratteri**[42] = '\_';

**elencoCaratteri**[43] = '=';

**elencoCaratteri**[44] = '€';

**elencoCaratteri**[45] = '£';

**elencoCaratteri**[46] = '$';

**elencoCaratteri**[47] = '%';

**elencoCaratteri**[48] = '&';

**elencoCaratteri**[49] = '?';

**elencoCaratteri**[50] = '\*';

**elencoCaratteri**[51] = '/';

**elencoCaratteri**[52] = '+';

**elencoCaratteri**[53] = '-';

**elencoCaratteri**[54] = '[';

**elencoCaratteri**[55] = ']';

**elencoCaratteri**[56] = '{';

**elencoCaratteri**[57] = '}';

**elencoCaratteri**[58] = '!';

**elencoCaratteri**[59] = '^';

**do** {

**parolaChiave** = "";

**condizioneCicloDoWhile** = **true**;

**trovataCifraNumerica** = **false**;

**trovataLetteraMinuscola** = **false**;

**trovatoCarattereSpeciale** = **false**;

**for** (**int** i = 0; i<lunghezzaParolaChiave; i++) {

**parolaChiave** = **parolaChiave** + **elencoCaratteri**[**caso**.nextInt(60)];

**if** ((**condizioneCicloDoWhile**==**true**)&&(lunghezzaParolaChiave>=3)) {

**if** ((**trovataCifraNumerica**==**false**)&&((**parolaChiave**.charAt(i)=='0')||(**parolaChiave**.charAt(i)=='1')||(**parolaChiave**.charAt(i)=='2')||(**parolaChiave**.charAt(i)=='3')||(**parolaChiave**.charAt(i)=='4')||(**parolaChiave**.charAt(i)=='5')||(**parolaChiave**.charAt(i)=='6')||(**parolaChiave**.charAt(i)=='7')||(**parolaChiave**.charAt(i)=='8')||(**parolaChiave**.charAt(i)=='9')))

**trovataCifraNumerica** = **true**;

**if** ((**trovataLetteraMinuscola**==**false**)&&((**parolaChiave**.charAt(i)=='a')||(**parolaChiave**.charAt(i)=='b')||(**parolaChiave**.charAt(i)=='c')||(**parolaChiave**.charAt(i)=='d')||(**parolaChiave**.charAt(i)=='e')||(**parolaChiave**.charAt(i)=='f')||(**parolaChiave**.charAt(i)=='g')||(**parolaChiave**.charAt(i)=='h')||(**parolaChiave**.charAt(i)=='i')||(**parolaChiave**.charAt(i)=='j')||(**parolaChiave**.charAt(i)=='k')||(**parolaChiave**.charAt(i)=='l')||(**parolaChiave**.charAt(i)=='m')||(**parolaChiave**.charAt(i)=='n')||(**parolaChiave**.charAt(i)=='o')||(**parolaChiave**.charAt(i)=='p')||(**parolaChiave**.charAt(i)=='q')||(**parolaChiave**.charAt(i)=='r')||(**parolaChiave**.charAt(i)=='s')||(**parolaChiave**.charAt(i)=='t')||(**parolaChiave**.charAt(i)=='u')||(**parolaChiave**.charAt(i)=='v')||(**parolaChiave**.charAt(i)=='w')||(**parolaChiave**.charAt(i)=='x')||(**parolaChiave**.charAt(i)=='y')||(**parolaChiave**.charAt(i)=='z')))

**trovataLetteraMinuscola** = **true**;

**if** ((**trovatoCarattereSpeciale**==**false**)&&((**parolaChiave**.charAt(i)=='(')||(**parolaChiave**.charAt(i)==')')||(**parolaChiave**.charAt(i)=='.')||(**parolaChiave**.charAt(i)==',')||(**parolaChiave**.charAt(i)==':')||(**parolaChiave**.charAt(i)==';')||(**parolaChiave**.charAt(i)=='\_')||(**parolaChiave**.charAt(i)=='=')||(**parolaChiave**.charAt(i)=='€')||(**parolaChiave**.charAt(i)=='£')||(**parolaChiave**.charAt(i)=='$')||(**parolaChiave**.charAt(i)=='%')||(**parolaChiave**.charAt(i)=='&')||(**parolaChiave**.charAt(i)=='?')||(**parolaChiave**.charAt(i)=='\*')||(**parolaChiave**.charAt(i)=='/')||(**parolaChiave**.charAt(i)=='+')||(**parolaChiave**.charAt(i)=='-')||(**parolaChiave**.charAt(i)=='[')||(**parolaChiave**.charAt(i)==']')||(**parolaChiave**.charAt(i)=='{')||(**parolaChiave**.charAt(i)=='}')||(**parolaChiave**.charAt(i)=='!')||(**parolaChiave**.charAt(i)=='^')))

**trovatoCarattereSpeciale** = **true**;

**if** ((**trovataCifraNumerica**==**true**)&&(**trovataLetteraMinuscola**==**true**)&&(**trovatoCarattereSpeciale**==**true**))

**condizioneCicloDoWhile** = **false**;

}

}

} **while** (**condizioneCicloDoWhile**==**true**);

**return** **parolaChiave**;

}

// Metodo pubblico statico per la creazione di una parola chiave alfanumerica maiuscola con caratteri speciali.

**public** **static** **String** creaParolaChiaveAlfanumericaMaiuscolaCaratteriSpeciali(**int** lunghezzaParolaChiave) {

**elencoCaratteri** = **new** **char**[60];

**for** (**int** i = 0; i<26; i++) {

**elencoCaratteri**[i] = (**char**) (65+i);

}

**for** (**int** i = 0; i<10; i++)

**elencoCaratteri**[i+26] = (**char**) (48+i);

**elencoCaratteri**[36] = '(';

**elencoCaratteri**[37] = ')';

**elencoCaratteri**[38] = '.';

**elencoCaratteri**[39] = ',';

**elencoCaratteri**[40] = ':';

**elencoCaratteri**[41] = ';';

**elencoCaratteri**[42] = '\_';

**elencoCaratteri**[43] = '=';

**elencoCaratteri**[44] = '€';

**elencoCaratteri**[45] = '£';

**elencoCaratteri**[46] = '$';

**elencoCaratteri**[47] = '%';

**elencoCaratteri**[48] = '&';

**elencoCaratteri**[49] = '?';

**elencoCaratteri**[50] = '\*';

**elencoCaratteri**[51] = '/';

**elencoCaratteri**[52] = '+';

**elencoCaratteri**[53] = '-';

**elencoCaratteri**[54] = '[';

**elencoCaratteri**[55] = ']';

**elencoCaratteri**[56] = '{';

**elencoCaratteri**[57] = '}';

**elencoCaratteri**[58] = '!';

**elencoCaratteri**[59] = '^';

**do** {

**parolaChiave** = "";

**condizioneCicloDoWhile** = **true**;

**trovataCifraNumerica** = **false**;

**trovataLetteraMaiuscola** = **false**;

**trovatoCarattereSpeciale** = **false**;

**for** (**int** i = 0; i<lunghezzaParolaChiave; i++) {

**parolaChiave** = **parolaChiave** + **elencoCaratteri**[**caso**.nextInt(60)];

**if** ((**condizioneCicloDoWhile**==**true**)&&(lunghezzaParolaChiave>=3)) {

**if** ((**trovataCifraNumerica**==**false**)&&((**parolaChiave**.charAt(i)=='0')||(**parolaChiave**.charAt(i)=='1')||(**parolaChiave**.charAt(i)=='2')||(**parolaChiave**.charAt(i)=='3')||(**parolaChiave**.charAt(i)=='4')||(**parolaChiave**.charAt(i)=='5')||(**parolaChiave**.charAt(i)=='6')||(**parolaChiave**.charAt(i)=='7')||(**parolaChiave**.charAt(i)=='8')||(**parolaChiave**.charAt(i)=='9')))

**trovataCifraNumerica** = **true**;

**if** ((**trovataLetteraMaiuscola**==**false**)&&((**parolaChiave**.charAt(i)=='A')||(**parolaChiave**.charAt(i)=='B')||(**parolaChiave**.charAt(i)=='C')||(**parolaChiave**.charAt(i)=='D')||(**parolaChiave**.charAt(i)=='E')||(**parolaChiave**.charAt(i)=='F')||(**parolaChiave**.charAt(i)=='G')||(**parolaChiave**.charAt(i)=='H')||(**parolaChiave**.charAt(i)=='I')||(**parolaChiave**.charAt(i)=='J')||(**parolaChiave**.charAt(i)=='K')||(**parolaChiave**.charAt(i)=='L')||(**parolaChiave**.charAt(i)=='M')||(**parolaChiave**.charAt(i)=='N')||(**parolaChiave**.charAt(i)=='O')||(**parolaChiave**.charAt(i)=='P')||(**parolaChiave**.charAt(i)=='Q')||(**parolaChiave**.charAt(i)=='R')||(**parolaChiave**.charAt(i)=='S')||(**parolaChiave**.charAt(i)=='T')||(**parolaChiave**.charAt(i)=='U')||(**parolaChiave**.charAt(i)=='V')||(**parolaChiave**.charAt(i)=='W')||(**parolaChiave**.charAt(i)=='X')||(**parolaChiave**.charAt(i)=='Y')||(**parolaChiave**.charAt(i)=='Z')))

**trovataLetteraMaiuscola** = **true**;

**if** ((**trovatoCarattereSpeciale**==**false**)&&((**parolaChiave**.charAt(i)=='(')||(**parolaChiave**.charAt(i)==')')||(**parolaChiave**.charAt(i)=='.')||(**parolaChiave**.charAt(i)==',')||(**parolaChiave**.charAt(i)==':')||(**parolaChiave**.charAt(i)==';')||(**parolaChiave**.charAt(i)=='\_')||(**parolaChiave**.charAt(i)=='=')||(**parolaChiave**.charAt(i)=='€')||(**parolaChiave**.charAt(i)=='£')||(**parolaChiave**.charAt(i)=='$')||(**parolaChiave**.charAt(i)=='%')||(**parolaChiave**.charAt(i)=='&')||(**parolaChiave**.charAt(i)=='?')||(**parolaChiave**.charAt(i)=='\*')||(**parolaChiave**.charAt(i)=='/')||(**parolaChiave**.charAt(i)=='+')||(**parolaChiave**.charAt(i)=='-')||(**parolaChiave**.charAt(i)=='[')||(**parolaChiave**.charAt(i)==']')||(**parolaChiave**.charAt(i)=='{')||(**parolaChiave**.charAt(i)=='}')||(**parolaChiave**.charAt(i)=='!')||(**parolaChiave**.charAt(i)=='^')))

**trovatoCarattereSpeciale** = **true**;

**if** ((**trovataCifraNumerica**==**true**)&&(**trovataLetteraMaiuscola**==**true**)&&(**trovatoCarattereSpeciale**==**true**))

**condizioneCicloDoWhile** = **false**;

}

}

} **while** (**condizioneCicloDoWhile**==**true**);

**return** **parolaChiave**;

}

// Metodo privato statico per la ricerca delle parole "EXIT" o "ESCI" o delle lettere "EE", scritte in maiuscolo o minuscolo, in una stringa di testo dopo sono presenti degli spazi.

**private** **static** **boolean** esci(**String** stringaInput) {

**int** j = 0;

**boolean** condizioneUscitaProgramma = **false**;

**boolean** condizioneCicloDoWhile = **false**;

**boolean** trovatoPrimoCarattereDiversoDaSpaziatura = **false**;

**boolean** trovatoSpaziaturaDopoCarattere = **false**;

**do** {

condizioneCicloDoWhile = **true**;

// Questa condizione si verifica quando l'utente inserisce una stringa vuota, e termina subito l'esecuzione del metodo restituendo per la variabile booleana "condizioneUscita" un valore pari a FALSE.

**if** (stringaInput.length()==0)

condizioneCicloDoWhile = **false**;

**else**

**for** (**int** i = 0; i< stringaInput.length(); i++) {

// Queste condizioni, eliminano gli eventuali spazi che (nella stringa) precedono la parola ricercata. Appena viene trovato un carattere diverso da quello di spaziatura, viene assegnata alla variabile "trovatoPrimoCarattereDiversoDaspaziatura" un valore booleano pari a TRUE, causando il salto di queste istruzioni di selezione nelle successive iterazioni del ciclo FOR.

**if** ((trovatoPrimoCarattereDiversoDaSpaziatura==**false**)&&(stringaInput.charAt(i)!=' ')) {

trovatoPrimoCarattereDiversoDaSpaziatura = **true**;

**break**;

}

**if** ((trovatoPrimoCarattereDiversoDaSpaziatura==**false**)&&(stringaInput.charAt(i)==' ')) {

stringaInput = stringaInput.substring(0, i) + stringaInput.substring((i+1), stringaInput.length());

**break**;

}

// Questa condizione si verifica quando (dopo aver eliminato gli eventuali caratteri di spaziatura che precedono la parola da ricercare) compare una sequenza di più caratteri consecutivi che formano una parola. Vengono contate le lettere della parola fino a quando non viene trovato (eventualmente) un altro carattere di spaziatura.

**if** ((trovatoPrimoCarattereDiversoDaSpaziatura==**true** )&&(trovatoSpaziaturaDopoCarattere==**false**)&&(stringaInput.charAt(i)!=' '))

j++;

**else**

trovatoSpaziaturaDopoCarattere = **true**;

// Queste condizioni (dopo aver spostato la prima parola nella prima parte della stringa) verificano che non vi siano caratteri diversi da quello di spaziatura (che vengolo eliminati) dopo la parola.

**if** ((trovatoSpaziaturaDopoCarattere==**true**)&&(i==j)&&(stringaInput.charAt(i)==' ')) {

stringaInput = stringaInput.substring(0, i) + stringaInput.substring((i+1), stringaInput.length());

**break**;

} **else** **if** ((trovatoSpaziaturaDopoCarattere==**true**)&&(i==j)&&(stringaInput.charAt(i)!=' ')) {

condizioneCicloDoWhile = **false**;

**break**;

}

// Questa condizione si verifica quando (dopo aver isolato nella stringa solo la prima parola o lettera trovata e aver eliminato gli eventuali spazi) l'indice "i" del ciclo FOR giunge all'ultima lettera della parola (posizione dell'ultima lettera -1). Il metodo verifica che la parola o la lettera rimanente nella stringa corrisponda alle parole ricercate "EXIT" o "E" o "ESCI" e in caso affermativo assegna alla variabile booleana "condizioneUscita" un valore pari a TRUE.

**if** (i==stringaInput.length()-1) {

stringaInput = stringaInput.toUpperCase();

**if** ((stringaInput.equalsIgnoreCase("EE"))||stringaInput.equalsIgnoreCase("EXIT")||(stringaInput.equalsIgnoreCase("ESCI"))) {

condizioneUscitaProgramma = **true**;

condizioneCicloDoWhile = **false**;

} **else**

condizioneCicloDoWhile = **false**;

}

}

} **while** (condizioneCicloDoWhile==**true**);

**return** condizioneUscitaProgramma;

}

// Metodo privato statico per la ricerca delle parole "RESTART" o "RIAVVIA" o delle lettere "RR", scritte in maiuscolo o minuscolo, in una stringa di testo dopo sono presenti degli spazi.

**private** **static** **boolean** riavvia(**String** stringaInput) {

**int** j = 0;

**boolean** condizioneRiavvioProgramma = **false**;

**boolean** condizioneCicloDoWhile = **false**;

**boolean** trovatoPrimoCarattereDiversoDaSpaziatura = **false**;

**boolean** trovatoSpaziaturaDopoCarattere = **false**;

**do** {

condizioneCicloDoWhile = **true**;

// Questa condizione si verifica quando l'utente inserisce una stringa vuota, e termina subito l'esecuzione del metodo restituendo per la variabile booleana "condizioneUscita" un valore pari a FALSE.

**if** (stringaInput.length()==0)

condizioneCicloDoWhile = **false**;

**else**

**for** (**int** i = 0; i< stringaInput.length(); i++) {

// Queste condizioni, eliminano gli eventuali spazi che (nella stringa) precedono la parola ricercata. Appena viene trovato un carattere diverso da quello di spaziatura, viene assegnata alla variabile "trovatoPrimoCarattereDiversoDaspaziatura" un valore booleano pari a TRUE, causando il salto di queste istruzioni di selezione nelle successive iterazioni del ciclo FOR.

**if** ((trovatoPrimoCarattereDiversoDaSpaziatura==**false**)&&(stringaInput.charAt(i)!=' ')) {

trovatoPrimoCarattereDiversoDaSpaziatura = **true**;

**break**;

}

**if** ((trovatoPrimoCarattereDiversoDaSpaziatura==**false**)&&(stringaInput.charAt(i)==' ')) {

stringaInput = stringaInput.substring(0, i) + stringaInput.substring((i+1), stringaInput.length());

**break**;

}

// Questa condizione si verifica quando (dopo aver eliminato gli eventuali caratteri di spaziatura che precedono la parola da ricercare) compare una sequenza di più caratteri consecutivi che formano una parola. Vengono contate le lettere della parola fino a quando non viene trovato (eventualmente) un altro carattere di spaziatura.

**if** ((trovatoPrimoCarattereDiversoDaSpaziatura==**true** )&&(trovatoSpaziaturaDopoCarattere==**false**)&&(stringaInput.charAt(i)!=' '))

j++;

**else**

trovatoSpaziaturaDopoCarattere = **true**;

// Queste condizioni (dopo aver spostato la prima parola nella prima parte della stringa) verificano che non vi siano caratteri diversi da quello di spaziatura (che vengolo eliminati) dopo la parola.

**if** ((trovatoSpaziaturaDopoCarattere==**true**)&&(i==j)&&(stringaInput.charAt(i)==' ')) {

stringaInput = stringaInput.substring(0, i) + stringaInput.substring((i+1), stringaInput.length());

**break**;

} **else** **if** ((trovatoSpaziaturaDopoCarattere==**true**)&&(i==j)&&(stringaInput.charAt(i)!=' ')) {

condizioneCicloDoWhile = **false**;

**break**;

}

// Questa condizione si verifica quando (dopo aver isolato nella stringa solo la prima parola o lettera trovata e aver eliminato gli eventuali spazi) l'indice "i" del ciclo FOR giunge all'ultima lettera della parola (posizione dell'ultima lettera -1). Il metodo verifica che la parola o la lettera rimanente nella stringa corrisponda alle parole ricercate "RESTART" o "R" o "RIAVVIA" e in caso affermativo assegna alla variabile booleana "condizioneUscita" un valore pari a TRUE.

**if** (i==stringaInput.length()-1) {

stringaInput = stringaInput.toUpperCase();

**if** ((stringaInput.equalsIgnoreCase("RR"))||stringaInput.equalsIgnoreCase("RESTART")||(stringaInput.equalsIgnoreCase("RIAVVIA"))) {

condizioneRiavvioProgramma = **true**;

condizioneCicloDoWhile = **false**;

} **else**

condizioneCicloDoWhile = **false**;

}

}

} **while** (condizioneCicloDoWhile==**true**);

**return** condizioneRiavvioProgramma;

}

// Metodo privato statico per la stampa a video.

**private** **static** **void** stampa(**String** stringaInput) {

**System**.***out***.print(stringaInput);

}

// Metodo privato statico per l'eliminazione degli spazi in una stringa di testo.

**private** **static** **String** eliminaSpazi(**String** stringaInput) {

**boolean** trovatoCarattereDiSpaziatura = **false**;

**do** {

trovatoCarattereDiSpaziatura = **false**;

**for** (**int** i = 0; i<stringaInput.length(); i++) {

**if** (stringaInput.charAt(i)==' ') {

stringaInput = stringaInput.substring(0, i) + stringaInput.substring((i+1), stringaInput.length());

trovatoCarattereDiSpaziatura = **true**;

**break**;

}

}

} **while** (trovatoCarattereDiSpaziatura==**true**);

**return** stringaInput;

}

// Metodo privato statico per la ricerca di un solo numero INTERO in una stringa di testo dove sono presenti degli spazi.

**private** **static** **boolean** trovaNumeroIntero (**String** stringaInput) {

**int** j = 0;

**boolean** trovatoUnNumeroIntero = **false**;

**boolean** trovatoUnCarattereNonConsentito = **false**;

**boolean** trovatoSpazioA = **false**;

**boolean** trovatoSpazioB = **false**;

**boolean** trovataUnaCifra = **false**;

**boolean** trovatoUnPrimoCarattereConsentito = **false**;

**boolean** trovatoUnoSpazioDopoUnCarattereValido = **false**;

**do** {

trovatoSpazioA = **false**;

trovatoSpazioB = **false**;

**for** (**int** i = 0; i<stringaInput.length(); i++) {

// Questa condizione verifica che non vengano mai inseriti caratteri non validi (diversi dalla cifre da 0 a 9, dal segno negativo (-) o positivo (+) e dai caratteri di spaziatura ( )).

**if** ((trovatoUnoSpazioDopoUnCarattereValido==**false**)&&(trovatoUnPrimoCarattereConsentito==**false**)&&((stringaInput.charAt(i)!=' ')&&(stringaInput.charAt(i)!='-')&&(stringaInput.charAt(i)!='+')&&(stringaInput.charAt(i)!='0')&&(stringaInput.charAt(i)!='1')&&(stringaInput.charAt(i)!='2')&&(stringaInput.charAt(i)!='3')&&(stringaInput.charAt(i)!='4')&&(stringaInput.charAt(i)!='5')&&(stringaInput.charAt(i)!='6')&&(stringaInput.charAt(i)!='7')&&(stringaInput.charAt(i)!='8')&&(stringaInput.charAt(i)!='9'))) {

trovatoUnCarattereNonConsentito = **true**;

trovatoUnNumeroIntero = **false**;

**break**;

// Questa condizione verifica che non vengano inseriti altri segni negativi (-) o positivi (+) dopo il primo carattere valido. In caso contrario, il programma restituisce un valore booleano FALSE per la variabile "trovatoUnNumeroIntero" e termina il ciclo iterativo FOR.

} **else** **if** ((trovatoUnoSpazioDopoUnCarattereValido==**false**)&&(trovatoUnPrimoCarattereConsentito==**true**)&&((stringaInput.charAt(i)!=' ')&&(stringaInput.charAt(i)!='0')&&(stringaInput.charAt(i)!='1')&&(stringaInput.charAt(i)!='2')&&(stringaInput.charAt(i)!='3')&&(stringaInput.charAt(i)!='4')&&(stringaInput.charAt(i)!='5')&&(stringaInput.charAt(i)!='6')&&(stringaInput.charAt(i)!='7')&&(stringaInput.charAt(i)!='8')&&(stringaInput.charAt(i)!='9'))) {

trovatoUnCarattereNonConsentito = **true**;

trovatoUnNumeroIntero = **false**;

**break**;

// Questa condizione si verifica eventualmente solo una volta, appena viene trovato come primo carattere valido un segno negativo (-) o positivo (+).

} **else** **if** ((trovatoUnoSpazioDopoUnCarattereValido==**false**)&&(trovatoUnPrimoCarattereConsentito==**false**)&&((stringaInput.charAt(i)=='-')||(stringaInput.charAt(i)=='+'))) {

trovatoUnPrimoCarattereConsentito = **true**;

// Questa condizione si verifica eventualmente solo una volta, appena viene trovato come carattere valido una cifra da 0 a 9. Questo ramo della selezione serve per verificare che nella stringa vi sia almeno un cifra.

} **else** **if** ((trovatoUnoSpazioDopoUnCarattereValido==**false**)&&(trovataUnaCifra==**false**)&&((stringaInput.charAt(i)=='0')||(stringaInput.charAt(i)=='1')||(stringaInput.charAt(i)=='2')||(stringaInput.charAt(i)=='3')||(stringaInput.charAt(i)=='4')||(stringaInput.charAt(i)=='5')||(stringaInput.charAt(i)=='6')||(stringaInput.charAt(i)=='7')||(stringaInput.charAt(i)=='8')||(stringaInput.charAt(i)=='9'))) {

trovataUnaCifra = **true**;

trovatoUnNumeroIntero = **true**;

trovatoUnPrimoCarattereConsentito = **true**;

// Questa condizione si verifica eventualmente tutte le volte che viene trovato un carattere di spaziatura prima di un carattere valido e lo elimina.

} **else** **if** ((trovatoUnPrimoCarattereConsentito==**false**)&&(stringaInput.charAt(i)==' ')) {

stringaInput = stringaInput.substring(0, i) + stringaInput.substring(i+1, stringaInput.length());

trovatoSpazioA = **true**;

**break**;

// Questa condizione si verifica eventualmente solo una volta, appena viene trovato un carattere di spaziatura dopo un carattere valido e lo elimina. Viene settata la variabile "j" = i e la variabile "trovatoUnoSpazioDopoUnCarattereValido" = TRUE per bypassare l'elaborazione dei primi 4 rami della struttura di selezione semplice del metodo.

} **else** **if** ((stringaInput.charAt(i)==' ')&&(j==0)) {

stringaInput = stringaInput.substring(0, i) + stringaInput.substring(i+1, stringaInput.length());

trovatoSpazioB = **true**;

trovatoUnoSpazioDopoUnCarattereValido = **true**;

j = i;

**break**;

// Questa condizione si verifica eventualmente tutte le volte che viene trovato un secondo carattere di spaziatura dopo un carattere valido ed elimina tutti gli spazi dopo il numero intero.

} **else** **if** ((stringaInput.charAt(i)==' ')&&(i==j)) {

stringaInput = stringaInput.substring(0, i) + stringaInput.substring(i+1, stringaInput.length());

trovatoSpazioB = **true**;

**break**;

// Questa condizione si verifica quando viene trovato un carattere diverso da quello di spaziatura dopo il numero intero. Il programma restituisce un valore booleano FALSE per la variabile "trovatoUnNumeroDecimale" e interrompe il ciclo iterativo FOR.

} **else** **if** ((stringaInput.charAt(i)!=' ')&&(i==j)) {

trovatoUnCarattereNonConsentito = **true**;

trovatoUnNumeroIntero = **false**;

**break**;

// Questa condizione si verifica se viene trovata almeno una cifra nella stringa di testo. In caso contrario il programma restituisce un valore booleano FALSE per la variabile "trovatoUnNumeroDecimale" e interrompe il ciclo iterativo FOR.

} **else** **if** (trovataUnaCifra!=**true**) {

trovatoUnNumeroIntero = **false**;

}

}

} **while** ((trovatoUnCarattereNonConsentito==**false**)&&((trovatoSpazioA==**true**)||(trovatoSpazioB==**true**)));

**return** trovatoUnNumeroIntero;

}

}