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

**import** java.util.ArrayList;

/\*\*

\* Questo programma contiene un algoritmo per l'ordinamento (dalla A alla Z) di una serie di oggetti di una classe in un array dinamico, in riferimento alla variabile di istanza "campoOrdinamento" incapsulata in ogni oggetto.

\* La classe, di cui si creano gli oggetti, deve avere sempre la variabile di istanza di tipo stringa "campoOrdinamento", e dei metodi pubblici per la lettura e la scrittura di tale variabile.

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

\* **@version** 1.0 (16 Aprile 2019).

\*/

**class** **StringaDiTesto** {

**private** **String** campoOrdinamento = **new** String();

**public** **void** campoOrdinamento(**String** stringaInput) {**this**.campoOrdinamento = stringaInput;}

**public** **String** campoOrdinamento() {**return** **this**.campoOrdinamento;}

}

**public** **class** **OrdinamentoOggettiArrayDinamico** {

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

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

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

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

**ArrayList**<StringaDiTesto> arrayDinamicoOggetti = **new** ArrayList<StringaDiTesto>();

**final** **int** lunghezzaCampoStringa = 80; // Costante utilizzata per definire la lunghezza massima (in caratteri) di una stringa.

**char** sceltaUtente = '\_';

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

**boolean** condizioneIterazioneMenuProgramma = **false**; // Variabile utilizzata per il controllo dell'iterazione del menu principale del programma.

**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 L'ORDINAMENTO (dalla A alla Z) DI UNA SERIE DI OGGETTI CONTENUTI IN UN ARRAY DINAMICO (ArrayList),");

*stampa*("\n\tIN RIFERIMENTO A UNA VARIABILE D'ISTANZA (di tipo stringa) INCAPSULATA IN OGNI OGGETTO.");

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

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

*stampa*("\n");

// Punto di inizio per l'iterazione del menu principale del programma.

**do** {

condizioneIterazioneMenuProgramma = **false**;

condizioneUscitaProgramma = **false**;

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

*stampa*("\tScegli un'operazione da svolgere:");

*stampa*("\n\tA) Inserisci una stringa di testo.");

*stampa*("\n\tB) Stampa a video lo stato interno di tutti gli oggetti contenuti nell'array dinamico.");

*stampa*("\n\tC) Ordina gli oggetti dell'array dinamico, dalla A alla Z, tenendo come riferimento le stringhe di testo inserite.");

*stampa*("\n");

// Inizio del ciclo DO... WHILE per la verifica dell'opzione inserita dall'utente relativa alla scelta dell'operazione da eseguire.

**do** {

condizioneCicloDoWhile = **true**;

*stampa*("\tQuale operazione scegli? ");

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.\n");

**else** {

stringaInput = stringaInput.toUpperCase();

sceltaUtente = stringaInput.charAt(0);

**if** ((sceltaUtente!='A')&&(sceltaUtente!='B')&&(sceltaUtente!='C'))

*stampa*("\tATTENZIONE: hai scelto un'opzione non in elenco.\n");

**else**

condizioneCicloDoWhile = **false**;

}

}

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

}

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

costruttoSwitch: // Impostazione di un'etichetta per la gestione di un eventuale salto del costrutto SWITCH... CASE tramite il comando BREAK.

**switch** (sceltaUtente) {

**case** 'A': {

// Inizio del ciclo DO... WHILE per la verifica della stringa inserita dall'utente.

**do** {

condizioneCicloDoWhile = **true**;

*stampa*("\tInserisci una stringa di testo: ");

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 DE PROGRAMMA IN CORSO ...\n");

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

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

**else** **if** (*eliminaSpaziInEccesso*(stringaInput).length()>lunghezzaCampoStringa)

*stampa*("\tATTENZIONE: la lunghezza massima consentita per una stringa di testo è di " + lunghezzaCampoStringa + " caratteri.\n");

**else** {

condizioneCicloDoWhile = **false**;

stringaInput = *eliminaSpaziInEccesso*(stringaInput);

**StringaDiTesto** oggettoClasseStringaDiTesto = **new** StringaDiTesto();

oggettoClasseStringaDiTesto.campoOrdinamento(stringaInput);

arrayDinamicoOggetti.add(oggettoClasseStringaDiTesto);

*stampa*("\n\tLa stringa è stata memorizzata nella variabile di istanza dell'oggetto, e l'oggetto è stato aggiunto all'array dinamico.\n\n");

}

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

**break**;

}

**case** 'B': {

**if** (arrayDinamicoOggetti.size()!=0) {

*stampa*("\n\tSTAMPA DELLO STATO INTERNO DEGLI OGGETTI CONTENUTI NELL'ARRAY DINAMICO:");

*stampa*("\n\tNumero degli oggetti contenuti nell'array dinamico:\t\t\t\t\t\t\t" + arrayDinamicoOggetti.size());

**for** (**int** i = 0; i<arrayDinamicoOggetti.size(); i++)

*stampa*("\n\tContenuto della variabile di istanza del " + (i+1) + "° oggetto contenuto nell'array dinamico (indice " + i + "):\t\t" + arrayDinamicoOggetti.get(i).campoOrdinamento());

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

} **else**

*stampa*("\n\tATTENZIONE: l'array dinamico non contiene alcun oggetto.\n\n");

**break**;

}

**case** 'C': {

**if** (arrayDinamicoOggetti.size()!=0) {

**String** stringa01 = **new** String();

**String** stringa02 = **new** String();

**int** lunghezzaMassimaStringa = 0;

**int** variazioneLunghezzaStringa = 0;

**boolean** effettuatoScambio = **false**;

{ // Blocco di elaborazione A.

// Questo ciclo FOR cerca la variabile di istanza "campoOrdinamento" di lunghezza maggiore (in caratteri) in ogni singolo oggetto dell'array dinamico "arrayDinamicoOggetti" e ne assegna il valore numerico alla variabile "lunghezzaMassimaStringa".

**for** (**int** i = 0; i<arrayDinamicoOggetti.size(); i++) {

**if** (arrayDinamicoOggetti.get(i).campoOrdinamento().length()>lunghezzaMassimaStringa)

lunghezzaMassimaStringa = arrayDinamicoOggetti.get(i).campoOrdinamento().length();

}

// Questo ciclo FOR aggiunge dei caratteri di spaziatura al termine della stringa contenuta nella variabile di istanza "campoOrdinamento", icapsulata in ogni oggetto dell'array dinamico "arrayDinamicoOggetti", fino a ottenere variabili di istanza (in ogni singlo oggetto) con lo stesso numero di caratteri (matrice rettangolare).

**for** (**int** i = 0; i<arrayDinamicoOggetti.size(); i++)

**if** (arrayDinamicoOggetti.get(i).campoOrdinamento().length()<lunghezzaMassimaStringa) {

variazioneLunghezzaStringa = lunghezzaMassimaStringa - arrayDinamicoOggetti.get(i).campoOrdinamento().length();

stringa01 = arrayDinamicoOggetti.get(i).campoOrdinamento();

**for** (**int** j = 0; j<variazioneLunghezzaStringa; j++)

stringa01 += " ";

arrayDinamicoOggetti.get(i).campoOrdinamento(stringa01);

}

}

{ // Blocco di elaborazione B.

// Questo ciclo DO... WHILE viene eseguito fino a quando viene effettuato un cambio nella posizione degli oggetti all'interno dell'array dinamico "arrayDinamicoOggetti".

**do** {

effettuatoScambio = **false**;

// Questo ciclo FOR ordina ogni oggetto dell'array dinamico "arrayDinamicoOggetti" in senso alfabetico dalla A alla Z, in riferimento alla variabile di istanza "campoOrdinamento".

**for** (**int** i = 0; i<arrayDinamicoOggetti.size()-1; i++) {

stringa01 = arrayDinamicoOggetti.get(i).campoOrdinamento().toLowerCase();

stringa02 = arrayDinamicoOggetti.get(i+1).campoOrdinamento().toLowerCase();

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

**if** ((stringa01.substring(0, j).equalsIgnoreCase(stringa02.substring(0, j)))&&(((**int**)stringa01.charAt(j))>((**int**)stringa02.charAt(j)))) {

effettuatoScambio = **true**;

arrayDinamicoOggetti.add(i, arrayDinamicoOggetti.get(i+1));

arrayDinamicoOggetti.remove(i+2);

}

}

}

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

}

*stampa*("\n\tGli oggetti all'interno dell'array dinamico sono stati ordinati (dalla A alla Z) in riferimento alla variabile di istanza contenuta in ogni oggetto.\n\n");

} **else**

*stampa*("\n\tATTENZIONE: l'array dinamico non contiene alcun oggetto.\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 operazione.

**do** {

condizioneCicloDoWhile = **true**;

*stampa*("\tVuoi eseguire una nuova operazione (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*("\tATTENIONE: hai scelto un'opzione non valida.\n");

**else** {

stringaInput = stringaInput.toUpperCase();

sceltaUtente = stringaInput.charAt(0);

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

*stampa*("\tATTENZIONE: ha scelto un'opzione non in elenco.\n");

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

condizioneCicloDoWhile = **false**;

condizioneIterazioneMenuProgramma = **true**;

} **else**

condizioneCicloDoWhile = **false**;

}

}

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

}

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

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

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

}

**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 l'eliminazione degli spazi ridondanti in una stringa di testo. Elimina gli spazi doppi, e quelli all'inizio e alla fine di una stringa di testo dove è presente almeno un carattere diverso da quello di spaziatura.

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

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

**boolean** trovatoCarattereDiversoDaSpazio = **false**;

**do** {

trovatoCarattereDiSpaziatura = **false**;

**for** (**int** i = 0; i<stringaInput.length()-1; i++)

**if** ((stringaInput.charAt(i)==' ')&&(stringaInput.charAt(i+1)==' ')) {

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

trovatoCarattereDiSpaziatura = **true**;

**break**;

}

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

**do** {

trovatoCarattereDiSpaziatura = **false**;

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

// Questa condizione si verifica quando vengono trovati dei caratteri di spaziatura all'inizio di una stringa di testo e li elimina.

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

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

trovatoCarattereDiSpaziatura = **true**;

**break**;

// Questa condizione si verifica quando viene trovato per la prima volta un carattere diverso da quello di spaziatura.

} **else** **if** ((trovatoCarattereDiversoDaSpazio==**false**)&&(stringaInput.charAt(0)!=' ')) {

trovatoCarattereDiversoDaSpazio = **true**;

// Questa condizione si verifica quando viene trovato un carattere di spaziatura alla fine di una stringa di testo e lo elimina.

} **else** **if** (stringaInput.charAt(stringaInput.length()-1)==' ') {

stringaInput = stringaInput.substring(0, stringaInput.length()-1);

trovatoCarattereDiSpaziatura = **true**;

**break**;

}

}

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

**return** stringaInput;

}

}