import java.util.\*;

/\*\*

\* Creates a cross.

\* Eleni Stylianou ge21708

\* 26/11/2021

\*/

public class Cross

{

private char data[][]; //The table that contains the characters of the cross

public ArrayList<Word> w=new ArrayList<Word>();// an arraylist with the words that we placed in the cross

/\*\*

\* Constructor of Cross

\* noLines - Number of lines

\* noRows - Number of Rows

\* newInitChar - Initial character We fill cross with this character e.g. '\_'

\*/

public Cross(int noRows, int noColumns)

{

data= new char[noRows][noColumns];

for (int i=0;i<noRows;i++){

for (int j=0;j<noColumns;j++){

data[i][j]='\_';

}

}

}

/\*\*

\* Check if word can put in (x,y) position and in Vertical or Horizontal direction if word can put in (x,y) position and in Vertical or Horizontal direction method return true else return false

\* aWord - the word that we want to add in Cross

\* x - the row that we want to add the word

\* y - the column that we want to add the word

\* d - the direction that we want to add the word (values 'H' for Horizontal 'V' for Vertical)

\*/

public boolean isLegal(String aWord,int x,int y,char d)

{

if(d=='V'){

if(data[x].length-x<aWord.length())

return false;

for(int i=y;i<data[x].length;i++){

int j=0;

if(data[x][i]!='\_' && data[x][y]!=aWord.charAt(j))

return false;

j+=1;

}

return true;

}

else{

if(data.length-y<aWord.length())

return false;

for(int i=x;i<data.length;i++){

int j=0;

if(data[i][y]!='\_' && data[x][y]!=aWord.charAt(j))

return false;

j+=1;

}

return true;

}

}

/\*\*

\* Find the first position where the word can be added AFTER a spesific position

\* aWord - the word that we want to add in Croos

\* x - the row that we want to add the word

\* y - the column that we want to add the word

\* d - the direction that we want to add the word (values 'H' for Horizontal 'V' for Vertical)

\*/

public Word findLegal(String aWord,int x,int y,char d)

{

for (int i=x;i<data.length;i++){

for (int j=y;j<data[x].length;j++){

if(isLegal(aWord,i,y,d))

return new Word(aWord,i,j,d);

}

}

return null;

}

/\*\*

\* Find position in all Cross where word can be added We select a random (x,y) position and a random direction and we use findLegal method to find if word can be added in Cross

\* aWord - the word that we want to add in Cross

\*/

public Word findLegalInAllCross(String aWord)

{

char direction='V';

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

for (int j=0;j<data[i].length;j++){

if(isLegal(aWord,i,j,direction))

return new Word(aWord,i,j,direction);

}

}

direction='H';

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

for (int j=0;j<data[i].length;j++){

if(isLegal(aWord,i,j,direction))

return new Word(aWord,i,j,direction);

}

}

return null;

}

/\*\*

\* Place word in a spesific position and direction in Cross Place the word in Cross and add the Word to Solution ArrayList

\* aWord - the word that we want to place in Cross

\* x - the row that we want to place the word

\* y - the column that we want to plce the word

\* d - the direction that we want to place the word (values 'H' for Horizontal 'V' for Vertical)

\*/

public void placeWord(String aWord, int x, int y, char d)

{

if(d=='V'){

int len=x+aWord.length();

for(int i=x;i<len;i++){

data[i][y]=aWord.charAt(i-x);

}

}

else{

int len=y+aWord.length();

for(int i=y;i<len;i++){

data[x][i]=aWord.charAt(i-y);

}

}

w.add(new Word(aWord,x,y,d));

}

/\*\*

\* Fill Cross (empties position) with random characters

\*/

public void FillRemaining()

{

Random rand = new Random();

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

for(int j=0;j<data[0].length;j++){

if(data[i][j]=='\_'){

data[i][j]= (char) (rand.nextInt(26) + 65);

}

}

}

}

/\*\*

\* Fill Cross with a dictionary

\* wordFill Cross with a dictionary

\*/

public void fillCrossFromArrayList(ArrayList wordList)

{

Word list;

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

list=findLegalInAllCross(wordList.get(i).toString());

if(list!=null)

placeWord(list.getWord(),list.getX(),list.getY(),list.getDirection());

}

}

/\*\*

\* Print Cross

\*/

public void PrintCross()

{

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

for (int j=0;j<data[i].length;j++){

System.out.print(data[i][j]+" ");

}

System.out.println();

}

}

/\*\*

\* Print the Solution

\*/

public void PrintSolution()

{

System.out.println("The Solution is:");

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

System.out.println(w.get(i).getWordInfo());

}

}

/\*\*

\* Reset Cross with initial character Fill Cross with initial character

\*/

public void resetCross()

{

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

for (int j=0; j<data[i].length; j++){

data[i][j] ='\_';

}

}

}

/\*\*

\* Returns the cross as a two-dimensional table

\*/

public char[][] getCross()

{

return data;

}

}