תרגול חזרה

G_ess

T_e Wor_

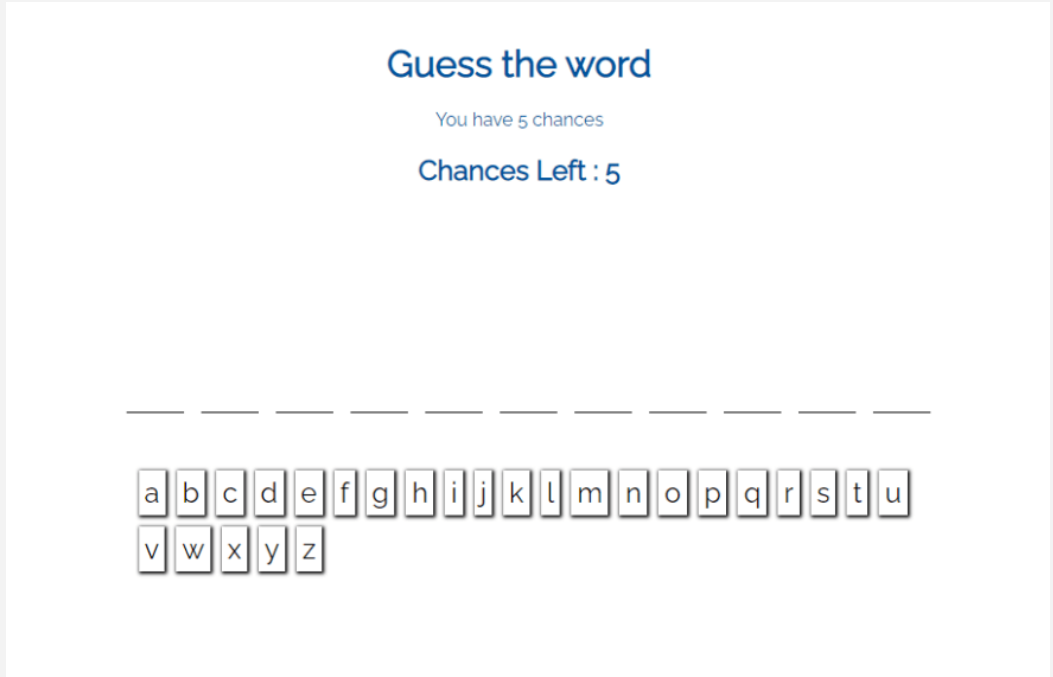
חוקי המשחק:

- מטרת המשחק:
- לנחש את המילה המסתתרת מאחורי התבנית, שבה חלק מהאותיות חסרות.
- התבנית תכיל את האותיות שגיליתם עד כה, ובמקום אלו שטרם גיליתם יופיע התו '_'

- מהלך המשחק:
- בכל פעם מותר לנחש אות אחת באנגלית ואם היא אכן מופיעה במילה היא תופיע במקומה במילה המקורית
- אם ניחשתם 3 אותיות אשר אינן מופיעות במילה – הפסדתם
- אם הצלחתם לנחש את המילה בשלמותה - ניצחתם

חוקי המשחק:

- עזרים:
- אם מתקשים לנחש את המילה ניתן לבקש רמז – הרמז מחזיר שתי אותיות (בסדר אלפביתי) – אחת שבהכרח קיימת במילה ואחת שאינה קיימת במילה. אך בכל מקרה אף אחת מהאותיות לא תהיה אות שכבר ניחשנו.



מימוש:

- נרצה לכתוב תוכנית אשר תקבל מילה ומספר אותיות מוסתרות משורת הפקודה ותבנה לה תבנית תחת ההגבלות הבאות:
  - בהכרח יש אותיות חשופות
  - בהכרח יש אותיות מוסתרות
  - בעבור כל אות – כל המופעים שלה במילה חשופים או מוסתרים
  - התבנית תבחר באופן רנדומלי מבין כל התבניות האפשריות
- התוכנית בכל סיבוב תקבל קלט מהמשתמש:
  - אות קטנה אשר תהיה אות שמנחשים
  - H גדולה, אשר תהיה בקשה לרמז

תכנון הקוד

- נגדיר תחילה את State שנרצה להחזיק – כלומר מהו המידע שרלוונטי להחזיק בזכרון לאורך המשחק כולו

- נרצה לכתוב אלגוריתם ראשי בו נכתוב את כל מהלך המשחק – **ביחידות כמה שיותר גדולות**
  - כלומר, נרצה להגדיר את ה״בלוקים״ הגדולים של המשחק, שאח״כ נפרק לבלוקים קטנים יותר.

State

- ה-State הוא המידע בזיכרון שאנחנו רוצים להחזיק על מנת להגדיר ולנהל משחק
- את המידע הזה נשמור במשתנים שילוו אותנו לכל אורך ריצת המשחק
- בכל זמן נתון של ריצת התוכנית ערכים אלו יתארו:
  - את התוכנית בה אנחנו נמצאים
  - את שלב הריצה בתוכנית בה אנחנו נמצאים

State

המידע שנרצה לשמור:

- המילה שצריך לנחש (מטיפוס String)
- תבנית המילה, שתכיל את האותיות שגילינו עד כה (מטיפוס char [])
- מספר הניחושים שנותרו (מטיפוס int)
- שמירת האותיות שניחשנו עד כה (מטיפוס ???)

String

- מחרוזות הן אובייקטים המכילים רצף של תווים.

String s = “Hello”;

index	0	1	2	3	4
character	H	e	l	l	o

- כל אלמנט במחרוזת הוא מסוג char.
- האינדקס של התו הראשון הוא 0.
- אורך המחרוזת מוחזר ע"י הפונקציה length()
- שרשור מחרוזות נעשה ע"י האופרטור +

String s2 = s + “ World” + 5 // “Hello World5”

String s3 = 6 + 5 + “ World” // “11World”

String

- מחרוזות הן אובייקט המחזיק אוסף של תווים.
- דוגמאות לפונקציות מהמחלקה String:

```
String str1 = "Hello";
char c = str1.charAt(0);
String str2 = str1.toUpperCase();
int strLength = str1.length();
```

// c == 'H'
// str2 == "HELLO"
// strLength == 5

- אופרטור שרשור:
  - `"Hello " + "World"` is `"Hello World"`
  - `"19" + 8 + 9` is `"1989"`

- String הם immutable לאחר היצירה – לא ניתן לשנות אותם

עוד ב-
https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/lang/String.html

String

מתודות בדיקה על String:

Method	Description
`equals(`**str**`)`	whether two strings contain the same characters
`equalsIgnoreCase(`**str**`)`	whether two strings contain the same characters, ignoring upper vs. lower case
`startsWith(`**str**`)`	whether one contains other's characters at start
`endsWith(`**str**`)`	whether one contains other's characters at end
`contains(`**str**`)`	whether the given string is found within this one

String

- נניח ונרצה להשוות שתי מחרוזות (לבדוק האם הן שוות).

```
public static void main(String[] args) {
        String s1 = new String("hello");
        String s2 = new String("hello");
        System.out.println(s1.equals(s2));
        System.out.println(s1 == s2);
}
```

true

false

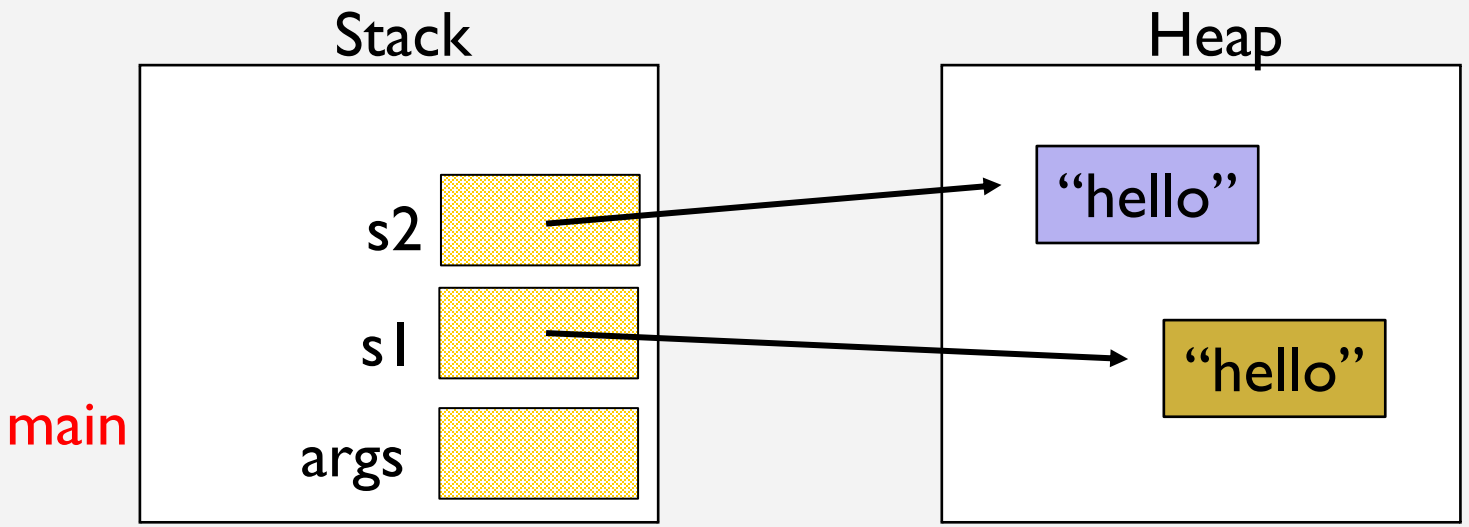
- מה יודפס למסך? למה?

String

```
public static void main(String[] args) {
String s1 = new String("hello");
String s2 = new String("hello");
System.out.println(s1.equals(s2));
System.out.println(s1 == s2);
}
```

true

false

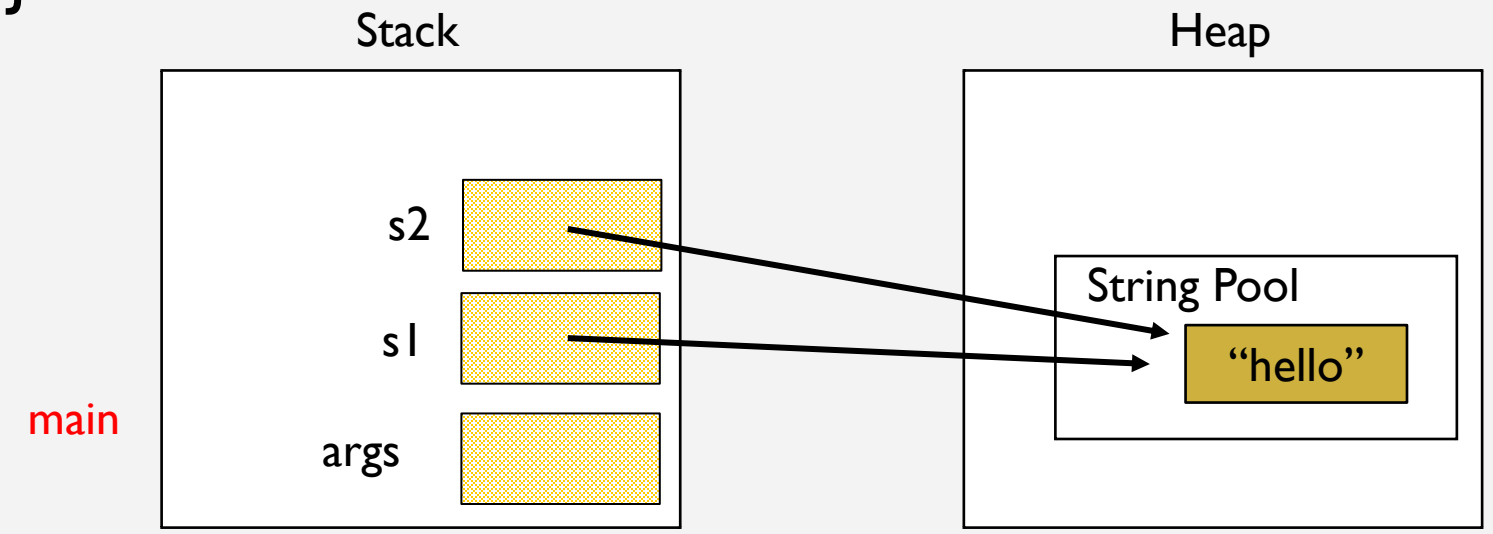


String

```
public static void main(String[] args) {
    String s1 = "hello";
    String s2 = "hello";
    System.out.println(s1.equals(s2));
    System.out.println(s1 == s2);
}
```

true

true



State

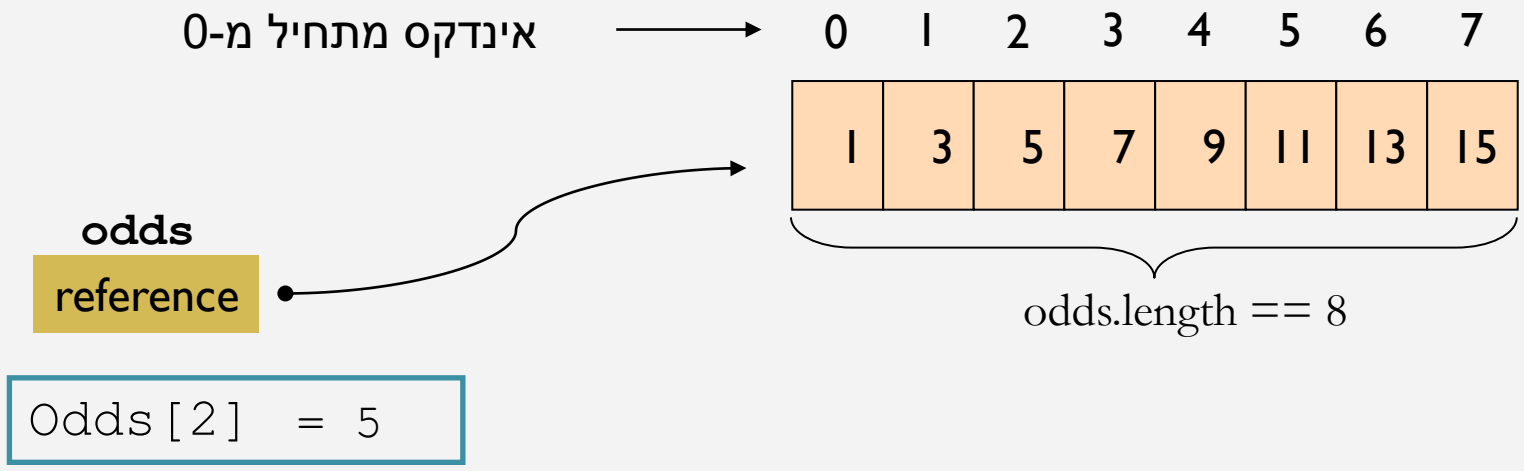
המידע שנרצה לשמור:

- המילה שצריך לנחש (מטיפוס String)
- תבנית המילה, שתכיל את האותיות שגילינו עד כה (מטיפוס char [])
- מספר הניחושים שנותרו (מטיפוס int)
- שמירת האותיות שניחשנו עד כה (מטיפוס ???)

מערכים

- **מערך:** מבנה נתונים בגודל קבוע מראש אשר שומר מספר איברים **מאותו הטיפוס**.
- לדוגמא: מערך עם ערכים אי זוגיים
- מדובר במערך של int בשם odds:

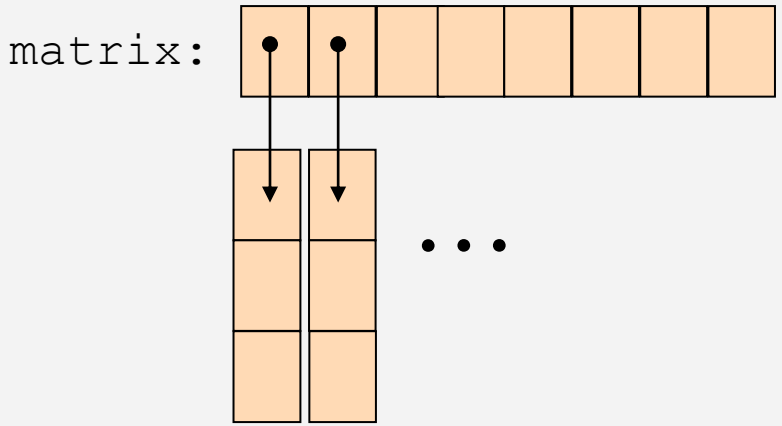
```
int[] odds = new int[8];
```




```
Odds[2] = 5
```

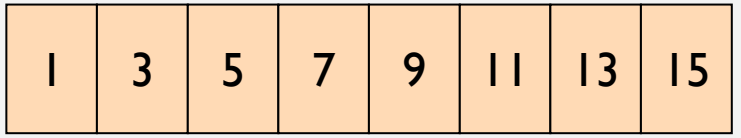
מערכים

- מערכים נסמן בעזרת סוגריים מרובעים [ ]
- דוגמאות:
  - `int[] odds;`
  - `String[] names;`
  - `int[][] matrix;` // an array of arrays

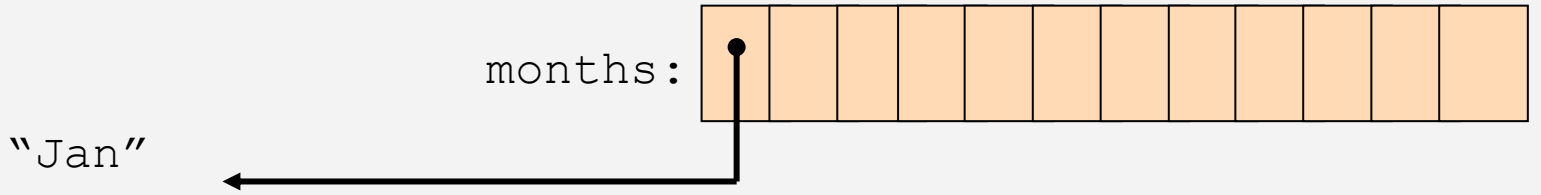


מערכים

- Creating and initializing small arrays with *a-priori* known values:
  - `int[] odds = {1,3,5,7,9,11,13,15};`



  - `String[] months = {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "July", "Aug", "Sep", "Oct", ...,"Nov", "Dec"};`



מערכים

- What is the output of the following code:

```
int[] odds = new int[8];
for (int i = 0; i < odds.length; i++) {
    System.out.print(odds[i] + " ");
    odds[i] = 2 * i + 1;
        System.out.print(odds[i] + " ");
}
```

Array creation: all elements are initialized with the default value for their type (`0` for `int`)

- Output:

```
0 1 0 3 0 5 0 7 0 9 0 11 0 13 0 15
```

מערכים

- By promoting the array's index:

Initialization condition step

```
for (int i = 0; i < months.length; i++) {
    System.out.println(months[i]);
}
```

- foreach:

The variable month is assigned with the next element in each iteration

```
for (String month: months) {
    System.out.println(month);
```

מערכים

- The class Arrays provide operations on array
  - Copy
  - Sort
  - binarySearch
  - Fill
  - ...
- java.util.Arrays
  http://docs.oracle.com/javase/6/docs/api/index.html?java/util/Arrays.html

מערכים

- **`Arrays.copyOf`**
  - 1st argument: the original array
  - 2nd argument: the length of the copy

```
int[] arr1 = {1, 2, 3};
int[] arr2 = Arrays.copyOf(arr1, arr1.length);
```

- **`Arrays.copyOfRange`**
  - 1st argument: the original array
  - 2nd initial index of the range to be copied, inclusive
  - 3rd argumrnt: final index of the range to be copied, exclusive

State

המידע שנרצה לשמור:

- המילה שצריך לנחש (מטיפוס String)
- תבנית המילה, שתכיל את האותיות שגילינו עד כה (מטיפוס char [])
- מספר הניחושים שנותרו (מטיפוס int)
- שמירת האותיות שניחשנו עד כה (מטיפוס ???)

State

??? already_guessed = ???;

boolean [] alreadyGuessed = new boolean [26];

זיכרון: הקצאת זכרון לכל האותיות מראש

השמה:

```
already_guessed[(int)c - 'a'] = true;
```

חיפוש:

```
boolean val = already_guessed[(int)c - 'a’];
```

הערה: נזכור שבכל מקרה סיבוכיות המקום והזמן היא קבועה O(1) שכן בכל מקרה יש מספר סופי של אותיות

String alreadyGuessed = “”;

זיכרון: הקצאת זכרון רק לאותיות שכבר ניחשנו

השמה:

```
already_guessed += c;
```

חיפוש:

```
for (int i = 0; i < already_guessed.length(); i++) {
        if (already_guessed.charAt(i) == c)
                return true;
}
return false;
```

תכנון הקוד

- נגדיר תחילה את הState שנרצה להחזיק – כלומר מהו המידע שרלוונטי להחזיק בזכרון לאורך המשחק כולו

- נרצה לכתוב אלגוריתם שיכיל בתוכו את מהלך המשחק, ולאחר מכן לתכנן מה צריך לממש

MAIN

קלט: מילה ומספר תווים חשופים

מהלך התוכנית:

MAIN

אתחול:

- אתחול שאר רכיבי הstate שהגדרנו
- מציאת תבנית מתאימה בעבור המילה ומספר התווים החשופים

סבבי ניחוש אותיות:

- כל עוד המשתמש עדיין יכול לנחש תווים:
  - קבלת תו מהמשתמש
  - אם התו הוא H גדולה:
    - נגריל שתי אותיות שטרם נוחשו, אחת במילה ואחת לא ונדפיס למשתמש
  - אחרת:
    - אם התו נמצא במילה נשנה את התבנית בהתאם להופעות התו במילה
    - אם התו לא נמצא במילה – נקטין את מספר הניחושים שנותרו ב-1
    - נדפיס את התבנית למשתמש
- סיום הריצה בלולאה:

  נבדוק מדוע יצאנו מהלולאה – ונדפיס הודעה בהתאם

תכנון הקוד

- נגדיר תחילה את הState שנרצה להחזיק – כלומר מהו המידע שרלוונטי להחזיק בזכרון לאורך המשחק כולו
- נרצה לכתוב אלגוריתם שיכיל בתוכו את מהלך המשחק, ולאחר מכן לתכנן מה צריך לממש

MAIN

ובקוד:

```
public static void playGame(String word, int hiddenLetters) {
        char [] puzzle = null;
        int leftGuesses = 3;
        boolean [] alreadyGuessed = new boolean [26];
        puzzle = getTemplate(word, hiddenLetters);

        char move = '\0';
        boolean isGuessedCorrect = false;



        ...
```

אתחול הState

מציאת תבנית

משתני עזר

MAIN

```
while (leftGuesses != 0) {
        printPuzzle(puzzle);
        printEnterYourGuessMessage();
        move = getValueFromUser();
        if (move == 'H') {
                printHint(getHint(word, puzzle, alreadyGuessed));
        }
        else {
                alreadyGuessed[((int) move - (int) 'a')] = true;
                isGuessedCorrect = guessLetter(move, word, puzzle);
                if (checkAllGuessed(puzzle)) {
                        printWinMessage();
                        break;
                }
                if (!isGuessedCorrect) {
                        leftGuesses--;
                }
        }
}
if (leftGuesses == 0) {
        printGameOver();
}
```

בדיקה האם נותרו עוד ניחושים
הדפסת החידה
קבלת קלט מהמשתמש

הדפסת רמז

ניחוש אות במילה

סיום התוכנית אם ניחשנו את המילה

הורדת מספר הניחושים שנותרו

MAIN

```
public static void main (String [] args) {
        playGame(args[0], Integer.parseInt(args[1]));
}
```

תזכורת:

```
public static void main(String[] args){
    int i = Integer.parseInt("1");
    double d = Double.parseDouble("-12.45e2");
}
```

// i==1
// d==-1245.0

המשך התכנון

כעת נרצה להשלים את התכנון שלנו של המשחק:

- נסתכל איזה פונקציות עזר בנינו – וניישם אותם
- נסתכל איזה פונקציות עזר בנינו לפונקציות העזר – וניישם גם אותן.

```
public static void playGame(String word, int hiddenLetters) {
	char [] puzzle = null;
	int leftGuesses = 3;
	boolean [] alreadyGuessed = new boolean [26];
	puzzle = getTemplate(word, hiddenLetters);

	char move = '\0';
	boolean isGuessedCorrect = false;



	...
```

getTemplate

- נחפש את כל הTemplates הקיימים
  - נעבור על כל template
  - נבחן האם הוא template חוקי לפי הגדרות התבנית

- נבחר template אחד באופן רנדומלי

בחירת מימוש: לעבור על כל ה2^word.length() אפשרויות ולבדוק כל אחת האם היא עומדת בתנאים, מבין כל העומדות בתנאים לבחור אחת באופן רנדומלי

- נעשה זאת בעזרת המתודה Integer.*toBinaryString*() אשר מקבלת מספר ומחזירה את String של הייצוג הבינארי שלו
  - נתייחס ל-0 כ-false ו-1 כ-true ונמיר את הString לboolean[]
  - נבדוק האם boolean[] מייצג תבנית חוקית

getTemplate

ובקוד:

```
private static char [] getTemplate(String word, int hidden_letters) {
      boolean [] template = randomTemplate(word, hidden_letters);
      return createPuzzleFromTemplate(word, template);
}
```

```
private static boolean[] randomTemplate(String word, int hidden) {
      boolean [][] templates;
      Random r = new Random();
      templates = getAllLegalTemplates(word, hidden);
      if (templates.length == 0) {
            return null;
      }
      return templates[r.nextInt(templates.length)];
}
```

String Format

```
int a=1805;
double b=123.456789;
System.out.println  ("a=" + a);              //"a=1805";

System.out.format ("a=%d\n", a);          //"a=1805";
System.out.format ("b=%.2f%n", b);        //"b=123.46"
System.out.format ("b=%20.10f%n", b); //"b=      123.4567890000"
```

%n - platform-specific line separator
%d – number
%f – float

http://docs.oracle.com/javase/tutorial/java/data/numberformat.html

getAllLegalTemplates

```
private static boolean[][] getAllLegalTemplates(String word, int k){
	int len = word.length();
	boolean [][] legalBin = new boolean [(int) Math.pow(2, len)][len];
	String binRep = "";
	int legalIndex = 0;
	for(int i=0; i < (int) Math.pow(2, len); i++) {
		binRep = String.format("%" + len + "s", Integer.toBinaryString(i)).replace(' ', '0');
		if (countOnes(binRep) == k && checkLegal(binRep, word)) {
			LegalBin[legalIndex++] = binToBool(binRep);
		}
	}
	return Arrays.copyOf(legalBin, legalIndex);
}
```

getAllLegalTemplates

```
private static boolean[][] getAllLegalTemplates(String word, int k){
        int len = word.length();
        boolean [][] legalBin = new boolean [(int) Math.pow(2, len)][len];
        String binRep = "";
        int legalIndex = 0;
        for(int i=0; i < (int) Math.pow(2, len);i++) {
                binRep = String.format("%" + len + "s", Integer.toBinaryString(i)).replace(' ', '0');
                if (countOnes(binRep) == k && checkLegal(binRep, word)) {
                        LegalBin[legalIndex++] = binToBool(binRep);
                }
        }
        return Arrays.copyOf(legalBin, legalIndex);
}
```

אתחול מערך דו מימדי

מימד 1: חסם עליון על כמות התבניות

מימד 2: אורך המילה

getAllLegalTemplates

```
private static boolean[][] getAllLegalTemplates(String word, int k){
        int len = word.length();
        boolean [][] legalBin = new boolean [(int) Math.pow(2, len)][len];
        String binRep = "";
        int legalIndex = 0;
        for(int i=0; i < (int) Math.pow(2, len);i++) {
                binRep = String.format("%" + len + "s", Integer.toBinaryString(i)).replace(' ', '0');
                if (countOnes(binRep) == k && checkLegal(binRep, word)) {
                        LegalBin[legalIndex++] = binToBool(binRep);
                }
        }
        return Arrays.copyOf(legalBin, legalIndex);
}
```

foramtה עוזר לקבל תבנית באורך קבוע

String.format(%5s,“10”); \\ ” 10”

replaceה עוזר לקבל תבנית מרופדת ב-0

” 10”.replace(‘ ‘, ‘0’); \\“00010”

getAllLegalTemplates

```
private static boolean[][] getAllLegalTemplates(String word, int k){
        int len = word.length();
        boolean [][] legalBin = new boolean [(int) Math.pow(2, len)][len];
        String binRep = "";
        int legalIndex = 0;
        for(int i=0; i < (int) Math.pow(2, len);i++) {
                binRep = String.format("%" + len + "s", Integer.toBinaryString(i)).replace(' ', '0');
                if (countOnes(binRep) == k && checkLegal(binRep, word)) {
                        LegalBin[legalIndex++] = binToBool(binRep);
                }
        }
        return Arrays.copyOf(legalBin, legalIndex);
}
```

getAllLegalTemplates

```
private static boolean[][] getAllLegalTemplates(String word, int k){
	int len = word.length();
	boolean [][] legalBin = new boolean [(int) Math.pow(2, len)][len];
	String binRep = "";
	int legalIndex = 0;
	for(int i=0; i < (int) Math.pow(2, len);i++) {
		binRep = String.format("%" + len + "s", Integer.toBinaryString(i)).replace(' ', '0');
		if (countOnes(binRep) == k && checkLegal(binRep, word)) {
			LegalBin[legalIndex++] = binToBool(binRep);
		}
	}
	return Arrays.copyOf(legalBin, legalIndex);
}
```

countOnes & binToBool

```
private static int countOnes(String bin_str) {
	int counter = 0;
	for(int i = 0; i < bin_str.length(); i++) {
		if (bin_str.charAt(i) == '1') {
			counter++;
		}
	}
	return counter;
}
```

```
private static boolean [] binToBool(String binStr) {
	boolean [] bool = new boolean [binStr.length()];
	for(int i = 0; i < binStr.length(); i++) {
		bool[i] = binStr.charAt(i) == '1';
	}
	return bool;
}
```

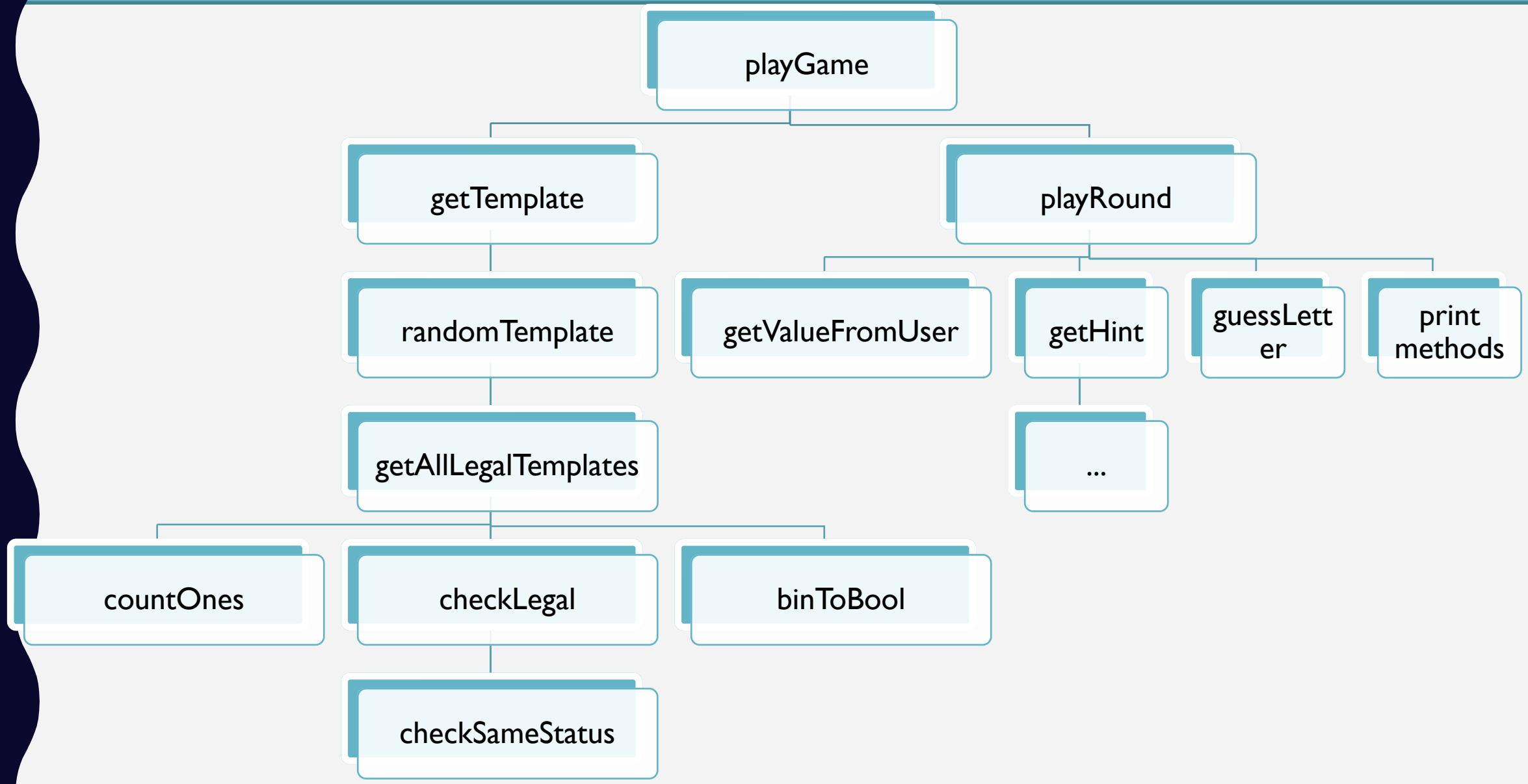
checkLegal

```
public static boolean checkLegal(String word, String strTemplate) {
        boolean [] template = binToBool(strTemplate);
        boolean hidden = false;
        boolean shown = false;
        if (word.length() != template.length) {
                return false;
        }
        for (int i = 0; i < template.length; i++) {
                if (template[i]) {
                        hidden = true;
                }
                else {
                        shown = true;
                }
                if(!checkSameStatus(i, word, template)) {
                        return false;
                }
        }
        return hidden && shown;
}
```

checkSameStatus

```
private static boolean checkSameStatus(int pos, String word, boolean [] template) {
	char c = word.charAt(pos);
	boolean status = template[pos];
	for(int i = pos; i < word.length(); i++) {
		if(c == word.charAt(i) && status != template[i]) {
			return false;
		}
	}
	return true;
}
```

סיכום


playGame
getTemplate
playRound
randomTemplate
getValueFromUser
getHint
guessLett
er
print
methods
getAllLegalTemplates
...
countOnes
checkLegal
binToBool
checkSameStatus