Write a Java program for a simple version of ***"hangman"*** game.

"*Hangman*" is a popular word puzzle game where you guess a secret word letter by letter.

The computer specifies a word and the user tries to guess it by suggesting letters.
The word to guess is represented by a row of dashes, representing each letter of the word.

If the user suggests a letter which occurs in the word, the computer writes it in all its correct positions.
If the suggested letter does not occur in the word, the computer decreases the score of the user by 3 or 2, for vowel and consonant letter respectively.

The game is over when:
- The user completes the whole word correctly or
- The user score reaches to equal or less than zero.

The words should be the name of animals such as monkey, penguin, dolphin, umbrellabird, etc.

Initially
Create an ***AnimalStack*** and insert 14 different animals into it.

Example:

AnimalStack	monkey	penguin	dolphin	umbrellabird	wildebeest	...	tarantula
	top						

All English letters should be stored in another stack (i.e. ***LetterStack***).

Example:

LetterStack	A	B	C	D	E	F	G	…	Z
	top								

The user has 12 points.

The Beginning of the Game

Generate a random number (*n*) and get n^{th} word of the *AnimalStack.*
Store this word in another stack (i.e. ***WordStack***).

Example:

WordStack	T	A	R	A	N	T	U	L	A
	top								

Game Playing
Game board should also be another stack (i.e. ***BoardStack***).

Example:

BoardStack	-	-	-	-	-	-	-	-	-
	top								

The missing letters should be stored in another stack (i.e. ***MissingLetterStack***).
You should delete the missing letter from ***LetterStack*** and insert it into ***MissingLetterStack***.
For each missing letter, you should decrease the points. wovel → decrease 3 consonant → decrease 2

If the user enters the same letter a second time, an error message must be displayed and it must not decrease the points that the user has.

Joker: The user has a chance to open a letter randomly only once time.

The program must display all steps until the game is over.

All solution must be implemented on the stacks, not on the strings.

In your program, you can use STACK as you want, but you must use <u>ONLY</u> stack, don't use other data structures such as normal (pure) array.

Sample output:

```
Word:  - - - - - - - - -        Misses:           Score: 12    ABCDEFGHIJKLMNOPRSTUVWXYZ
Guess: A
Word:  - A - A - - - - A        Misses:           Score: 12    BCDEFGHIJKLMNOPRSTUVWXYZ
Guess: Z
Word:  - A - A - - - - A        Misses: Z         Score: 10    BCDEFGHIJKLMNOPRSTUVWXY
Guess: E
Word:  - A - A - - - - A        Misses: Z E       Score: 7     BCDFGHIJKLMNOPRSTUVWXY
Guess: Z
You entered the same letter before.
Word:  - A - A - - - - A        Misses: Z E       Score: 7     BCDFGHIJKLMNOPRSTUVWXY
Guess: S
Word:  - A - A - - - - A        Misses: Z E S     Score: 5     BCDFGHIJKLMNOPRTUVWXY
Guess: T
Word:  T A - A - T - - A        Misses: Z E S     Score: 5     BCDFGHIJKLMNOPRUVWXY
Guess: Joker
Word:  T A - A - T U - A        Misses: Z E S     Score: 5     BCDFGHIJKLMNOPRVWXY
Guess: B
Word:  T A - A - T U - A        Misses: Z E S B   Score: 3     CDFGHIJKLMNOPRVWXY
Guess: O
Word:  T A - A - T U - A        Misses: Z E S B O Score: 0     CDFGHIJKLMNPRVWXY
You lost !!
```

Sample output:

```
Word:  - - - - -                Misses:           Score: 12    ABCDEFGHIJKLMNOPRSTUVWXYZ
Guess: A
Word:  - - - - A                Misses:           Score: 12    BCDEFGHIJKLMNOPRSTUVWXYZ
Guess: Z
Word:  Z - - - A                Misses:           Score: 12    BCDEFGHIJKLMNOPRSTUVWXY
Guess: Joker
Word:  Z - B - A                Misses:           Score: 12    CDEFGHIJKLMNOPRSTUVWXY
Guess: U
Word:  Z - B - A                Misses: U         Score: 9     CDEFGHIJKLMNOPRSTVWXY
Guess: R
Word:  Z - B R A                Misses: U         Score: 9     CDEFGHIJKLMNOPSTVWXY
Guess: E
Word:  Z E B R A                Misses: U         Score: 9     CDFGHIJKLMNOPSTVWXY
You win !!
```