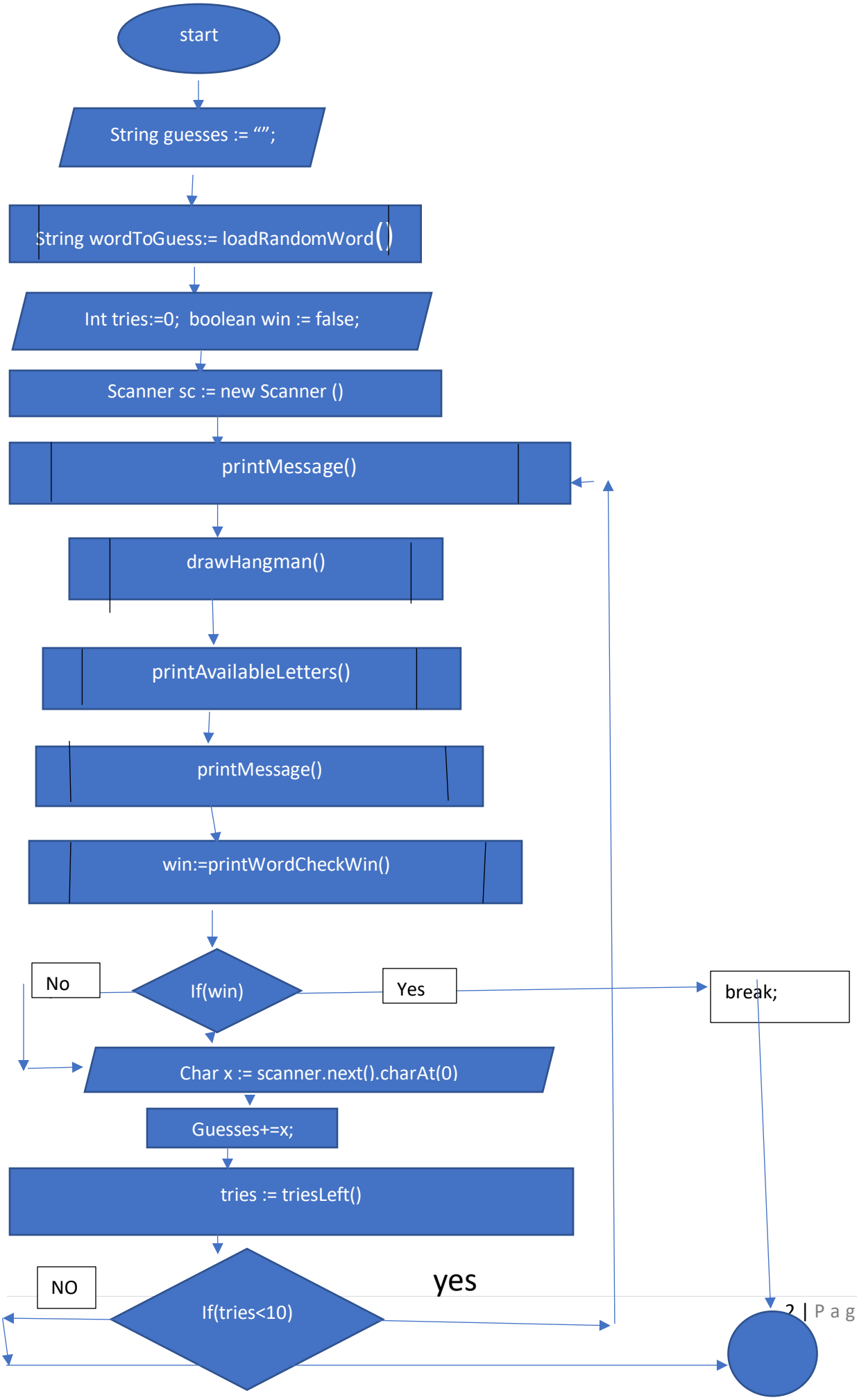
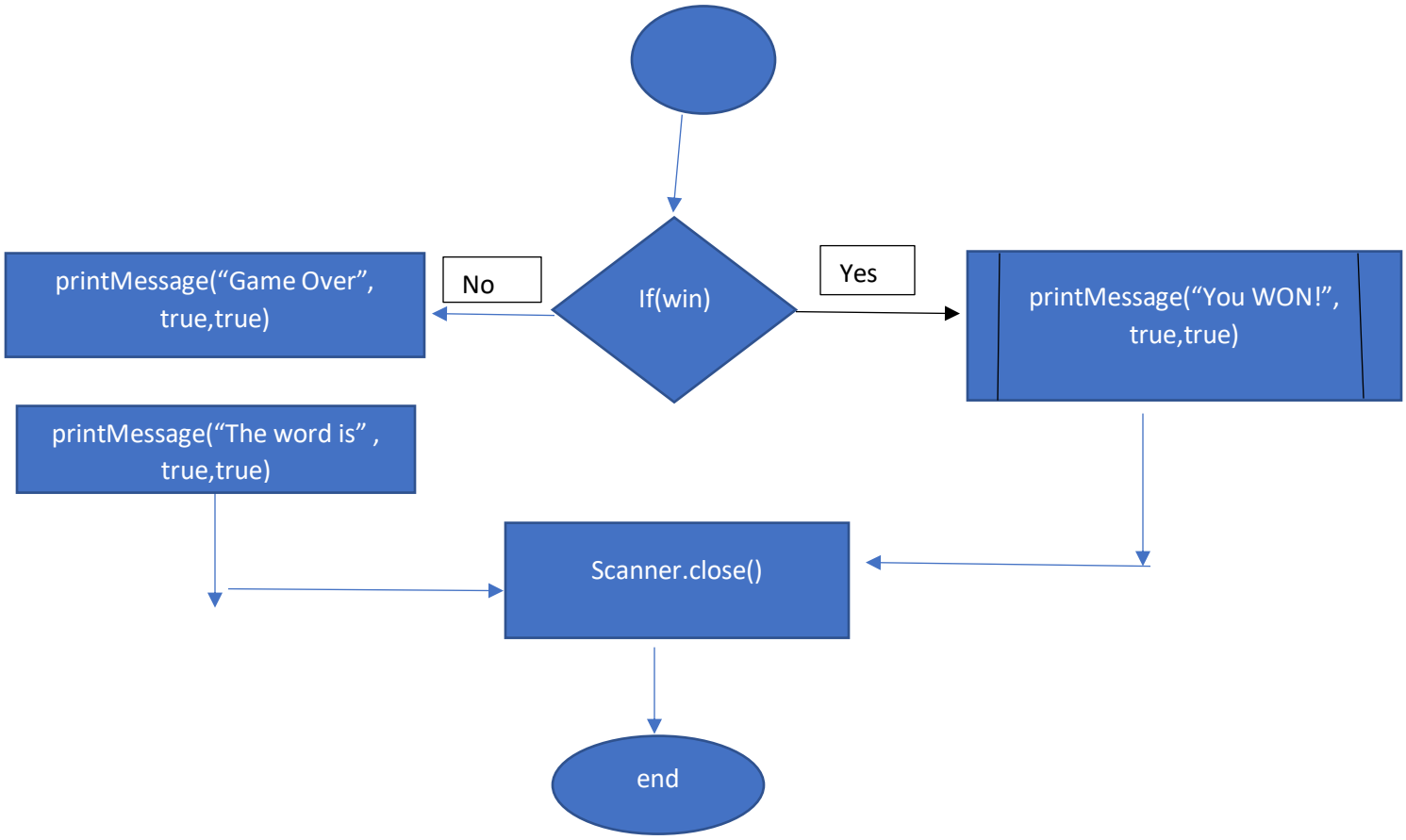
Съдържание


start
String guesses := “”;
String wordToGuess:= loadRandomWord()
Int tries:=0; boolean win := false;
Scanner sc := new Scanner ()
printMessage()
drawHangman()
printAvailableLetters()
printMessage()
win:=printWordCheckWin()
No
If(win)
Yes
break;
Char x := scanner.next().charAt(0)
Guesses+=x;
tries := triesLeft()
NO
If(tries<10)
yes


printMessage(“Game Over”, true,true)
No
If(win)
Yes
printMessage(“You WON!”, true,true)
printMessage(“The word is” , true,true)
Scanner.close()
end

ДОКУМЕНТАЦИЯ НА HANGMAN

Първоначално се реализира речник от който, се избира на случаен принцип думата по която се играе играта *„Бесеница“*. Прочитат се всички думи от файл, по зададен път към файла, в метод readWordsFromFile(String path). Всяка от думите във файла е на нов ред. След прочитането на всяка дума тя се добавя в структура от данни Вектор. Задава се капацитет на вектора 69 900, колкото е броят на думите в речника.

Векторът е структура от данни, която представлява последователност, която поддържа достъп до нейните елементи чрез техните индекси. Съдържа множество от n елемента, съхранявани в определен линеен ред. Всеки елемент има уникален индекс в диапазона [0, n-1], който е равен на броя на елементите, които са преди него.

Векторите, поддържат следните основни функции (в допълнение към стандартните *size()* и *isEmpty()* функции):

- *setElementAt(i, e)* – Заменя елементът с индекс i с e.
- *insertElementAt (i, e)* – Вмъква нов елемент e на позиция i.
- *removeElementAt(i)* – Премахва елемента с индекс i.
- *add(e)* – Добавя елемента e на края на вектора.
- *E elementAt(i)* – Връща елемента с индекс i.

След това от създадения речник, имплементиран с вектор се избира случаен елемент в метод *loadRandomWo*rd (Vector<String> vector). На случаен принцип с инстанция на Random класа се избира индекс, който е между 0 и размера на вектора. Методът връща елемента на тази позиция в структурата от данни.

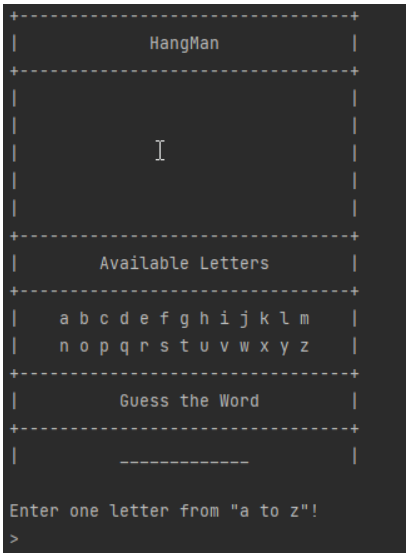
```
/**
 * loadRandomWord().
 * Loads word randomly.
 * @param  vector
 * @return word
 *
 */
static String loadRandomWord(Vector<String> vector) throws Exception {
    if (vector.size() == 0) {
        throw new Exception("Invalid input");
    }
    Random rand = new Random();
    int randLine = rand.nextInt(vector.size());
    String word = vector.elementAt(randLine);
    return word;
}
```

Фигура 1

В следващите методи програмата разпечатва първоначалната версия на играта. Това се случва в цикъл с постусловие, в тялото на do, while цикъл. Тези методи ще се повтарят докато опитите да се познае думата не достигнат 10.

- *printMessage();*
- *drawHangman();*
- *printAvailableLetters();*
- *printWordCheckWin(wordToGuess, guesses.toString());*

Основна функция за рисуването на дъската е *printMessage(String message, boolean printTop, boolean printBottom).* Функцията (Фигура 3) се използва във всички методи, които включват рисуването на играта. Участъците за заглавието Hangman, празния участък за бесеницата, рисуването на бесилката, Available Letters, буквите на латиница от a до z, Guess the Word, изписването на съобщението за победа и за край на играта и изписване на непознатата дума с тирета (Фигура 2).



Фигура 2

Текстът в играта се позиционира винаги по средата на екрана. Ако аргументите *boolean printTop, boolean printBottom* са *true* се рисува горното и/или долното очератние около текста, които са с фиксирана дължина (Фигура 3). Позиционирането на текста по средата на дъската се случва в цикъл, който започва от индекс с големина равна на дължината на текста и продължава до 33 (дължината на рамката на дъската). Основната част на алгоритъма конкатенира празно място преди текста или след текста, като се редуват, двете условия. Флаг front се променя, за да се конкатенират равен брой празни места от двете страни на текста, докато дължината на стринга достигне дължината на рамката на дъската, за която е зададена дължина 33. Ако аргументите *boolean printTop, boolean printBottom ca true* се рисуват отгоре и отдолу очертанията с фиксирана дължина.

```java
/**
 * printMessage() prints the game console box and positions any text in the middle.
 *
 */
static void printMessage(String message, boolean printTop, boolean printBottom) {
    if (printTop) {
        System.out.println("+---------------------------------+");
    }
    System.out.print("|");
    boolean front = true;
    for (int i = message.length(); i < 33; i++) {
        if (front) {
            message = " " + message;
        } else {
            message = message + " ";
        }
        front = ! front;
    }

    System.out.print(message);

    if (printBottom) {
        System.out.println("|");
        System.out.println("+---------------------------------+");
    } else {
        System.out.println("|");
    }
}
```

Фигура 3

Методът *drawHangman(int guessCount)* рисува бесилката и празните пространства на този участък от играта, под заглавието. Използва *printMessage()* за да позиционира обесеното човече по средата на играта, и да конкатенира празните места.

Методът *printAvailableLetters(String taken)*, рисува участъка "Available Letters" с очертания и изписва буквите на латиница на два реда първо от 'a' до 'm' и от 'n' до 'z'. В изписването на буквите се прескачат тези букви, които са били изпробвани в играта. Първоначано се изписват всички малки букви от латинската азбуката, а след всеки ход/опит на играча, на мястото на буквите, които са били използвани в играта се поставя празно място.

Изписването на буквите става с извикването на метод *printLetters(String input, char from, char to)*, който се извиква първоначално с параметри празен стринг, буквите 'a' и 'm' и втори път с празен стринг и буквите 'n' и 'z' (Фигура 4). Методът *printLetters()* инициализира *StringBuilder*. StringBuilder-ът е предпочетен пред String заради по-голямата ефикасност при изпълнение на конкатенации в цикъл. Методът *append(char c)* на StringBuilder осигурява ефективно преоразмеряване на масива, в който се съхранявата буквите, като вика метод *ensureCapacityInternal(int minimumCapacity)*.

В цикъл се конкатенират всички букви между char със стойност 'a'- 97 до char 'm'- 109. Това конкатениране се случва само при условие, че буквите не са били използвани в играта досега. Т.е. ако не се съдържат в низ *lettersUsed*, това се проверява с *lettersUsed.indexOf(i) == - 1*. Ако се съдържат в низ *lettersUsed*, се принтира празно място на тяхната позиция. Следва принтирането на низа от букви в StringBuilder-а с метод *printMessage(),* който описахме по-горе. (Фигура 4)

```java
/**
 * printLetters() prints the menu of Available letters, which are left for guessing, from char to char, avoiding the letters which are in the input string.
 * In their place a space is printed.
 *
 */
static void printLetters(String lettersUsed, char from, char to) {
    StringBuilder s = new StringBuilder( capacity: 33);
    for (char i = from; i <= to; i++) {
        if (lettersUsed.indexOf(i) == - 1) {
            s.append(i);
        }
        s.append(" ");
    }
    printMessage(s.toString(), printTop: false, printBottom: false);
}
```

Фигура 4

Следва проверка и изписване на познати/непознати букви в дума *printWordCheckWin()*. В цикъл с начало индекс равен на 0, и край дължината за думата за познаване, се прави проверка върху стринговия низ, който съдържа буквите, които са опитани от играча в опита. Проверява се дали в него се съдържат букви от думата за познаване, ако не съдържат се маркира флаг won с false, и се изписва долно тире на тяхно място, а ако се съдържат се изписват буквите и се флагът won е със стойност true. (Фигура 5)

```java
/**
 * printWordCheckWin() checks if any of the letters in the word for guessing are in the guessed word.
 * If any letter is correct, it is printed in its place, if any is missing an underscore is printed in its place.
 * If all letters in the word for guessing are in the guessed word the won equals to true, there is winner.
 * @param word
 * @param usedLetters
 * @return
 */
static boolean printWordCheckWin(String word, String usedLetters) {
    boolean won = true;
    StringBuilder s = new StringBuilder(word.length());

    for (int i = 0; i < word.length(); i++) {
        if (usedLetters.indexOf(word.charAt(i)) == - 1) {
            won = false;
            s.append("_");
        } else {
            s.append(word.charAt(i));
            s.append(" ");
        }
    }
    printMessage(s.toString(), printTop: false, printBottom: false);
    return won;
}
```

Фигура 5

Този флаг won прекратява do while цикъла, в който се играе играта. Ако won == true, цикълът се прекъсва. Ако е false, играчът въвежда буква през конзолата в метод *getUserInput(Scanner scanner)*. Буквата е между a-z, това се проверява в метод *isBasicLetter(char c)*. Ако буквата не е опитана до сега , и не се съдържа в *StringBuilder lettersUsed*, се добавя към *lettersUsed*.

Методът *triesLeft()* брои грешните опити за познаване на думата и връща бройката, която се използва за чертаене на бесеницата. Грешните опити се равняват на всички букви, които са в низа *guessedLetters*, но ги няма в думата за познаване. Изчисляват се в променлива *error*. И ако тези опити за познаване на думата са по-малко от десет, започва отново изпълнение на *do, while* цикъла.(Фигура 6)

```java
/**
 * triesLeft() counts the number of letters which are incorrect.
 * @param word
 * @param guessedWord
 * @return
 */
static int triesLeft(String word, String guessedWord) {
    int error = 0;
    for (int i = 0; i < guessedWord.length(); i++) {
        if (word.indexOf(guessedWord.charAt(i)) == - 1) {
            error++;
        }
    }
    return error;
}
```

Фигура 6

Ако играчът познае думата се изписва „You Won”, а ако загуби „Game Over и думата.

```
+---------------------------------+
|             HangMan             |
+---------------------------------+
|                |                |
|                |                |
|                O                |
|               /|\               |
|                                 |
+---------------------------------+
|        Available Letters        |
+---------------------------------+
|           d    h  j k           |
|        p q  s t  v w x y z      |
+---------------------------------+
|          Guess the Word         |
+---------------------------------+
|          a m o e b i c          |
+---------------------------------+
|             You WON!            |
+---------------------------------+
```

```
+---------------------------------+
|             HangMan             |
+---------------------------------+
|                |                |
|                |                |
|                O                |
|               /|\               |
|                |                |
|               / \               |
+---------------------------------+
|        Available Letters        |
+---------------------------------+
|        b c  e f    j  l m       |
|      o p q r s  u v w x  z       |
+---------------------------------+
|          Guess the Word         |
+---------------------------------+
|              ------             |
>
m
+---------------------------------+
|            Game over!           |
+---------------------------------+
+---------------------------------+
|        The word is roofer       |
+---------------------------------+
```

Фигура 7