

```

1  # Hangman Game (Jogo da Forca)
2  # Programação Orientada a Objetos
3
4  # Import
5  import random
6
7  # Board (tabuleiro)
8  tabuleiro = [ ' '
9
10 >>>>>>>>> JOGO DA FORCA <<<<<<<<<<
11
12 +---+
13 |
14 |
15 |
16 |
17 |
18 ===== ' ' , ' '
19
20 +---+
21 |
22 0
23 |
24 |
25 |
26 ===== ' ' , ' '
27
28 +---+
29 |
30 0
31 |
32 |
33 |
34 ===== ' ' , ' '
35
36 +---+
37 |
38 0
39 /|
40 |
41 |
42 ===== ' ' , ' '
43
44 +---+
45 |
46 0
47 /|\
48 |
49 |
50 ===== ' ' , ' '

```

```

51
52  +---+
53  |
54  0
55  /|\
56  /
57  |
58  ===== ' ', ' '
59
60  +---+
61  |
62  0
63  /|\
64  / \
65  |
66  ===== ' ' ]
67
68
69  # Classe
70  class JogoForca:
71
72      # Método Construtor
73      def __init__(self, palavra):
74          self.palavra = palavra
75          self.letras_erradas = []
76          self.letras_certas = []
77
78      # Método para adivinhar a letra
79      def advinha(self, letra):
80          if letra in self.palavra and letra not in self.
letras_certas:
81              self.letras_certas.append(letra)
82          elif letra not in self.palavra and letra not in
self.letras_erradas:
83              self.letras_erradas.append(letra)
84          else:
85              return False
86          return True
87
88      # Método para verificar se o jogo terminou
89      def fim_jogo(self):
90          return self.jogo_vencido() or (len(self.
letras_erradas) == 6)
91
92      # Método para verificar se o jogador venceu
93      def jogo_vencido(self):
94          if '_' not in self.palavra_oculta():
95              return True
96          return False
97

```

```

98     # Método para não mostrar a letra no board
99     def palavra_oculta(self):
100         rtn = ''
101         for letra in self.palavra:
102             if letra not in self.letras_certas:
103                 rtn += '_'
104             else:
105                 rtn += letra
106         return rtn
107
108     # Método para checar o status do game e imprimir o
board na tela
109     def print_status_jogo(self):
110         print (tabuleiro[len(self.letras_erradas)])
111         print ('\nPalavra: ' + self.palavra_oculta())
112         print ('\nLetras erradas: ',)
113         for letra in self.letras_erradas:
114             print (letra,)
115         print ()
116         print ('Letras corretas: ',)
117         for letra in self.letras_certas:
118             print (letra,)
119         print ()
120
121     # Função para ler uma palavra de forma aleatória do banco
de palavras
122     def palavra_aleatoria():
123         with open("palavras.txt", "rt") as f:
124             banco_palavras = f.readlines()
125         return banco_palavras[random.randint(0,len(
banco_palavras))].strip()
126
127
128     # Função Main - Execução do Programa
129     def main():
130
131         # Objeto
132         game = JogoForca(palavra_aleatoria())
133
134         # Enquanto o jogo não tiver terminado, print do
status, solicita uma letra e faz a leitura do caracter
135         while not game.fim_jogo():
136             game.print_status_jogo()
137             input_usuario = input('\nDigite uma letra: ')
138             game.advinha(input_usuario)
139
140         # Verifica o status do jogo
141         game.print_status_jogo()
142
143         # De acordo com o status, imprime mensagem na tela

```

```
143 para o usuário
144     if game.jogo_vencido():
145         print ('\nParabéns! Você venceu!!')
146     else:
147         print ('\nGame over! Você perdeu.')
148         print ('A palavra era ' + game.palavra)
149
150     print ('\nFoi bom jogar com você! Agora vá estudar
!\n')
151
152 # Executa o programa
153 if __name__ == "__main__":
154     main()
155
```