

Stwórz grę w „Wisielca”, program musi być wykonany w konwencji obiektowej. Przykładowy sposób implementacji:

Zasady:

Dostajemy informacje o ilości liter w hasle, i widzimy tyle właśnie podkreślników ile liter ma hasło:

```
"/Applications/IntelliJ IDEA CE.app/Contents/bin/java -jar ...
- - - - -
Odgadnij hasło, hasło ma długość: 8
Podaj literkę:
|
```

Po podaniu litery, która występuje w hasle, podkreślnik w odpowiednim miejscu podmienia się na literę:

```
- - - - -
Odgadnij hasło, hasło ma długość: 8
Podaj literkę:
a
- - - - - a -
Liczba pozostałych szans: 4
Podaj literkę:
```

Po podaniu litery, która nie występuje, zmniejsza się ilość szans.

W moim wydaniu wisielca jest rozbity na trzy klasy:

1. Wisielca -> klasa, która wszystko łączy, tutaj mamy losowanie hasła i algorytm gry
2. Runda -> Używam tej klasy do stworzenia obiektu posiadającego informacje o obecnej rozgrywce
3. Obsługa Łańcucha -> klasa ze statycznymi metodami wspomagającymi operacje na łańcuchach znaków

```
9 public class Wisielca {
10
11     String haslo;
12     ArrayList<String> listaHasel;
13     char [] hasloChary;
14     Runda runda;
15     Scanner odczyt;
16
17     public Wisielca() throws FileNotFoundException {
18         listaHasel = new ArrayList<>(); //Tworze liste z hasłami
19         wczytajHasla(); //wczytuje z pliku hasła, można w tej metodzie ustawić
20         //hasła statycznie (pododawać ręcznie)
21         losuj(); //Metoda losująca hasło z tablicy (tak naprawdę losuje indeks)
22         hasloChary = ObslugaLancucha.zwrocChary(haslo); //Zamieniam hasło na tablicę z podkreślnikami
23         runda = new Runda(haslo, hasloChary); //Tworzę obiekt nowej gry
24         //uruchomienie rozgrywki
25         gra();
26
27     }
28
29     private void gra() {...} //Mechanizm gry
61     private void wczytajHasla() throws FileNotFoundException {...} //wczytywanie hasel
75     private void losuj(){...} //losowanie indeksu listy z hasłami
```

```

3 public class Runda {
4
5     String haslo; // haslo jako napis
6     char [] doOdgadniecia; //tablica z podkreslnikami
7     char [] hasloTablica; // tablica ze spacjami
8     int szanse;
9
10
11 @ public Runda(String haslo, char[] hasloTablica){
12     this.haslo = haslo;
13     this.hasloTablica = hasloTablica;
14     doOdgadniecia = new char[hasloTablica.length];
15     szanse = 4;
16     inicjujTablice();
17 }
18
19 private void inicjujTablice() {
20     for(int i = 0, j = 0; i < haslo.length(); i++, j+=2){
21         doOdgadniecia[j] = '_';
22         doOdgadniecia[j+1] = ' ';
23     }
24     System.out.println(doOdgadniecia);
25 }
26 }

```

```

3 public class ObslugaLancucha {
4
5 @ static char [] zwrocChary(String haslo){ //...
6
7
8
9
10     char [] tablicaChar = haslo.toCharArray(); //ta bedzie mniejsza
11     char [] doWyswietlenia = new char[2 * (tablicaChar.length)]; //ta dwa razy wieksza
12
13     for(int i = 0, j = 0; i < tablicaChar.length; i++, j+=2){
14         doWyswietlenia[j] = tablicaChar[i];
15         doWyswietlenia[j+1] = ' ';
16     }
17     return doWyswietlenia;
18 }
19
20
21 @ static void sprawdz(Runda runda, char literka){...}
22
23 //metoda sprawdza czy literka jest w hasle
24
25
26
27
28
29
30 @ static boolean sprawdzWygrana(Runda runda){...}
31
32 //metoda sprawdza, czy zostały jeszcze jakieś podkreślniki do zastąpienia
33 //Jak nie... no to wygrana, bo hasło odkryte
34 }

```