

## ***Dokumentacja klas programu J-Flashcards***

Spis klas:

1. Answer\_set
2. Answer
3. Button
4. Button\_set
5. Constants
6. Comps
7. Frame
8. Frame\_states
9. Program

Aby uruchomić program zachęcam do korzystania z napisanego przez mnie skryptu Makefile. Wystarczy skorzystać w linuxowym terminalu z polecenia make. Uruchomi ona program, zaś po skończeniu usunie niepotrzebne pliki z systemu.

1) Answer\_set – klasa zawierająca zbiór obiektów typu Answer. Posiada jedno pole ans\_list, które zawiera odczytane z pliku question.txt linie.

Metody:

- Answer get\_random\_answer() - zwraca losowo wybraną odpowiedź ze zbioru linii pliku a następnie usuwa ją ze zbioru ans\_list, w celu nie powtarzania się odpowiedzi
- void append\_answer(String ans) – dodaje do pliku odpowiedzi wraz z pytaniem
- int answer\_count() - zwraca ilość odpowiedzi w pliku.

2) Answer – klasa będąca logiczną implementacją danych w grze, prezentowanych w pliku jako linie, każda linia konwertowana jest na “odpowiedź”. Zawiera ona 3 pola klasy:

- int correct\_num – numer oznaczający poprawną odpowiedź
- String question – łańcuch znakowy będący zapytaniem dla użytkownika
- String[] answers – trzy odpowiedzi z których użytkownik może wybrać jedną

Metody:

- String get\_question() - zwraca zapytanie odpowiedzi
- String get\_answer(int which) – zwraca odpowiedź dla which = 0,1,2
- String get\_correct() - zwraca na podstawie int correct\_num poprawną odpowiedź

3) Button – implementacja interfejsu Ibutton, zawiera pole typu Jbutton b, ma na celu późniejsze ułatwienie implementacji poszczególnych przycisków, metody:

- Button(int b1, int b2, int b3, int b4, String name) – tworzy przycisk o nazwie name, szerokości b4, długości b3 we współrzędnej w układzie kartezjańskim:  $x = b1$   $y = b2$
- Jbutton get\_button() - zwraca przycisk typu Jbutton
- void hide\_button() - ukrywa z planszy przycisk
- void show\_button() - pokazuje na planszy przycisk

4) Button\_set – klasa implementująca przyciski na podstawie dziedziczenia klasy Button oraz implementacji klasy ActionListener z zaimportowanego narzędzia java.awt.event.ActionListener. Implementacja jest niezbędna do wykonania akcji w trakcie naciśnięcia przycisku przez użytkownika.

Opis klas zagnieżdżonych w tejże klasie:

- Exit\_button – klasa odpowiadająca za wyjście z programu
- Start\_button – klasa odpowiadająca za rozpoczęcie gry
- Admin\_button – klasa odpowiadająca za wejście do panelu logowania administratora
- Menu\_button – klasa odpowiadająca za powrót do menu startowego aplikacji
- Proceed\_button – klasa odpowiadająca za sprawdzenie, czy hasło wpisane w polu hasła administratora jest poprawne, jeśli tak, to pozwala użytkownikowi na wejście do panelu administratora, w przeciwnym wypadku wyświetla stosowny komunikat o niepoprawnym hasle
- Done\_button – przycisk odpowiedzialny za dopisanie do zbioru tekstowego plansz kolejnej planszy, tylko używane w panelu administratora
- Choice\_button – przycisk odpowiedzialny za wybranie poprawnej odpowiedzi w panelu administratora
- Game\_button – przycisk służący za wybranie poprawnej odpowiedzi przez użytkownika w trakcie gry, jeżeli wybrał dobrze, jest to sygnalizowane stosownym komunikatem
- Next – przycisk służący do przejścia do następnej planszy

5) Constants – klasa pomocnicza zawierająca następujące stałe statyczne:

- String WINDOW\_NAME – nazwa okna przy uruchomieniu programu
- int[] winsize – dwuelementowa tablica która zawiera wymiary okna programu
- String TXT\_PATH – ścieżka do pliku question.txt
- String file\_name – nazwa pliku z zapytaniami (zawsze będzie to question.txt)
- String paswd – hasło do panelu kontrolnego administratora (zawsze będzie to admin)

6) Comps – klasa zawierająca informacje czy plansza została wygenerowana oraz listy komponentów do plansz. Jeżeli plansza została niewygenerowana, trzeba uzupełnić listy komponentów, jeżeli zaś została wygenerowana I chcemy przejść do tej planszy, to taka lista komponentów znacznie ułatwia zadanie, wystarczy wtedy ustawić komponenty na widoczne.

Pola tej klasy informujące o wygenerowaniu planszy (typu boolean):

- admin\_login\_rendered
- admin\_panel\_rendered
- game\_rendered

Pola tej klasy zawierające komponenty:

- Button b\_exit – przycisk służący do wychodzenia z programu
- Button b\_menu – przycisk powracający do menu głównego
- List<Button> s\_buttons – startowe przyciski
- List<Components> admin\_login – przyciski w panelu logowania
- List<Components> admin\_panel – zbiór przycisków w panelu administratora

7) Frame – klasa opisująca okno programu

Pole klasy JFrame służy głównie do manipulacji różnymi parametrami.

Metody:

- void Add\_Button(Button b) – dodaje do okna przycisk b
- void Add\_Comp(Component c) – dodaje komponent c, różnica jedynie taka, że Button jest pochodną typu Component, metoda jest ułatwieniem dla późniejszej implementacji
- void Refresh\_screen() - istotna dla działania ogółu programu metoda, odświeża okno w celu “pokazania” na nim komponentów, bez tej metody byłyby one niewidoczne

8) Frame\_states – klasa opisująca stany w jakim jest obecnie okno, manipuluje głównie polem statycznym Frame main\_frame

Metody:

- void Show\_exit() , void Show\_menu() - pokazuje w oknie przycisk wyjścia lub menu
- void Startup() – inicjuje okno startowe wraz z przyciskami
- void Admin\_password() - stan okna w którym użytkownik ma zamiar zalogować się do panelu administratora
- void Admin\_panel() - stan okna w którym użytkownik z powodzeniem dostał się do panelu administratora I umożliwia mu wykonanie akcji dodania plansz do gry
- void Render\_is\_correct() - pomocnicza metoda, mająca w zamiarze wyświetlić

- informację, czy gracz poprawnie odpowiedział na pytanie
- void Game() - okno w stanie gry, gdzie użytkownik ma połączyć zapytanie z poprawną odpowiedzią

9) Program – klasa z metodą statyczną main, uruchamiająca niezbędne do rozpoczęcia gry komponenty.

Maksymilian Wnuk