







Zaimplementuj jednookienkową aplikację analogiczną do "Monkeytype" ([zobacz tutaj](#)).

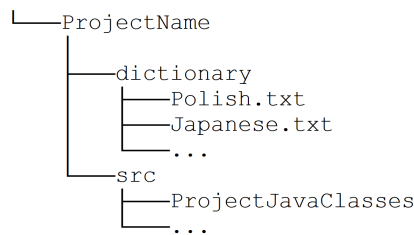
Aplikacja ma na celu zbadać sprawność szybkiego i precyzyjnego pisania na klawiaturze oraz wyświetlić statystyki po przeprowadzonym teście.

W ramach aplikacji, na górnym panelu wyświetlamy menu z dostępnymi opcjami (wybór języka, wybór czasu testu). Na dolnym panelu wyświetlamy stopkę z informacjami na temat dostępnego skrótu klawiszowego (zaimplementuj wskazane funkcjonalności):

-  +  - restart test
-  +  +  - pause
-  - end test

W ramach aplikacji użytkownik wybiera jeden z dostępnych języków. Lista języków zależy od liczby dostarczonych plików zawierających słowa. W dołączonym pliku **dictionary.zip** udostępnionych jest 19 plików tekstowych (kodowanych w UTF-8) reprezentujących różne języki (dla każdego języka po 50k-600k słów). Należy stworzyć aplikację tak, że w razie zmiany liczby plików, bez potrzeby żadnej modyfikacji aplikacja dostosuje liczbę dostępnych języków w aplikacji. Każdy plik nosi nazwę zgodną z językiem jaki reprezentuje.

Katalog **dictionary** należy umieścić bezpośrednio w strukturze projektu, t.j.



Podczas testu użytkownik stara się jak najszybciej napisać widoczny przed sobą tekst. Rozpatrujemy każde słowo z osobna, a momentem przejścia do wpisywania kolejnego słowa jest wcisnięcie przycisku spacji. Zadbaj o zmianę koloru znaków, zgodnie z poniższym przykładem:

- Tekst jeszcze nie wprowadzony ma kolor szary:
make turn interest write time present little around eye
- Znaki poprawnie wprowadzone mają kolor zielony:
make turn interest write time present little around eye
- Znaki błędnie wprowadzone mają kolor czerwony:
make turn interest write time present little around eye
- Znaki nadmiarowo wprowadzone mają kolor pomarańczowy:
make turn interest write time present littleeee around eye
- Znaki ominięte mają kolor czarny:
make turn interest write time present littleeee around eye

Zaimplementuj możliwość wyboru czasu trwania rozgrywki: 15, 20, 45, 60, 90, 120, 300 sekund. Każdorazowo wyświetlamy użytkownikowi jeden akapit, a w sytuacji, gdy dojdziemy do jego końca, podmieniamy ukończony akapit nowym. Każdy akapit ma po 30 losowych słów ze słownika danego języka.

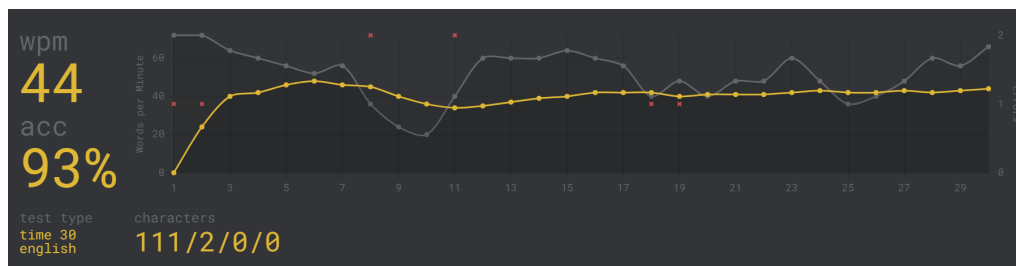
Należy zaimplementować również animację podskakiwania literek całego tekstu tworząc "falę" (przestrzegam przed wykorzystaniem gotowych rozwiązań z internetu). Ponad to zaimplementuj do dowolnych elementów okna dwie inne dowolne animacje.

Po teście należy wyświetlić wykres prezentujący w rozkładzie na kolejne sekundy rozgrywki:

- aktualny WPM (word per minute) w konkretnej sekundzie
- średni WPM w konkretnej sekundzie

Ponadto należy wyliczyć średni WPM dla całego testu, statystyki wprowadzonych znaków w postaci (correct/incorrect/extra/missed) oraz podać jaki % znaków został wpisany poprawnie.

Przykład takiego zestawienia prezentuje się następująco:



W efekcie rozgrywki powinien powstać w katalogu projektu plik zawierający datę i godzinę wykonania testu, w ramach którego zaprezentowane będzie zestawienie kolejnych słów i wyliczonego WPM dla danego słowa.

Zawartość takiego pliku powinna prezentować się następująco:

```
become -> 70wpm
will -> 83wpm
begin -> 99wpm
no -> 106wpm
see -> 80wpm
keep -> 58wpm
...
```

Pamiętaj:

- Zadbaj o wyjątki w programie i wyświetlaj stosowne komunikaty użytkownikowi.
- Zadbaj o skalowalność okna i spójny efekt wizualny całej aplikacji.
- ***Aplikację zaimplementuj z użyciem technologii JavaFX oraz wzorca MVC***

Uwaga:

- ***W przypadku otrzymania projektu niezgodnego w powyższymi wymaganiami lub ze znacznymi brakami w implementacji, z niezaimplementowanym testem lub niekompilującego się skutkować to będzie wyzerowaniem punktacji za projekt***
- ***Zabrania się używania narzędzi WYSIWYG (tzw. Window/Scene Builder'ów)***
- ***Zabrania się wykorzystania plików FXML do tworzenia aplikacji***
- ***Brak znajomości dowolnej linii kodu lub wysokie podobieństwo nadesłanych rozwiązań skutkować będzie wyzerowaniem punktacji***
- ***Poza praktyczną i merytoryczną poprawnością będzie brana również pod uwagę optymalność, jakość i czytelność napisanego przez Państwa kodu.***