

Specyfikacja implementacyjna chatu z generatorem tekstu

Konrad Olszewski

18 maja 2015

Spis treści

1	Język	1
2	Opis	2
2.1	Użytkownik docelowy.	2
3	Podział na klasy - opis	2
3.1	Opis klas i metod	2
3.2	Opis GUI	3
4	OPIS FUNKCJONALNOŚCI	3
4.1	Jak korzystać z programu?	3
5	SCENARIUSZ DZIAŁANIA PROGRAMU.	3
5.1	Scenariusz ogólny:	3
6	TESTOWANIE.	4
6.1	Ogólny przebieg testowania:	4
7	DIAGRAM UML	4

1 Język

Program będzie napisany w języku java.

2 Opis

2.1 Użytkownik docelowy.

Pogram dedykowany jest studentom oraz pracownikom Politechniki Warszawskiej.

3 Podział na klasy - opis

3.1 Opis klas i metod

Program zostanie podzielony na następujące klasy:

1. Ngram.class - klasa będąca pojedynczym ngramem danego rzędu prefix (String ngram) + sufixy(String []nastepcy) + ilość wystąpień danego sufixu(ileRazy[])
 - rozszerzNgram(String slowo) - rozszerza Ngram o dodatkowy sufix
 - NgramRandom() - zwraca losowy sufix Ngramu
 - NgramSuma() - zwraca ilość wystąpień Ngramu
2. Read.class - klasa wczytująca plik bazowy, generuje ona listę ngramów z pliku z tekstem, bądź ze Stringa. Zawiera listęNgramów danego stopnia, listęngramów jednoskowych, poziom ngramu, oraz ścieżkę do bazy
 - String czytaj(String path) - klasa czyta plik który dostaje i zwraca go w formie Stringa
 - przetwórz(ArrayList <Ngram> ngramy,String in, Integer ngramLvl) - Pełną listę słów (in) dzieli na ngramy, następnie sprawdza zawieranie kolejnych ngramów w już istniejącej ich liście. Po sprawdzeniu czy istnieje dodaje nowy Ngram do listy lub rozszerza istniejący ngram.
 - Sortuj() - sortuje listę nramów po ilości wystąpień
3. Gui.class - klasa generująca interfejs graficzny
 - public void actionPerformed(ActionEvent e) - klasa odpowiadająca za akcje
4. Txtgenerator.class - klasa generująca tekst zadanej dlugosci na podstawie listy ngramow oraz odpowiedzi użytkownika.

5. `Statistics.class` - klasa generująca statystyki tekstów. Zawiera

- `String najczestszeWyrazy()`
- `String najczestszeSlova()`
- `String najczestsze(String in, List $\langle Ngram \rangle$ ngramy)` - wyznacza listę n najpopularniejszych, w zależności od listy, ngramów/ słów

3.2 Opis GUI

Gui będzie podzielone na dwie części. Główne okno czatu które wygląda jak na Załączonym obrazku `g1.png`. Po prawej stronie będzie widoczna część statystyki. Tutaj wyświetlana będzie bieżąca statystyka tekstu.

4 OPIS FUNKCJONALNOŚCI

4.1 Jak korzystać z programu?

Aby korzystać z programu należy wczytać plik bazowy. Następnie pojawia się możliwość wpisywania wiadomości na które uzyskiwane będą odpowiedzi w oknie chatu.

Po prawej stronie okna wyświetlana będzie aktualna statystyka tekstu.

Ponadto program będzie posiadał następujące funkcje:

1. wybrania nicku
2. wychwycenie błędów we wpisywanym tekście

5 SCENARIUSZ DZIAŁANIA PROGRAMU.

5.1 Scenariusz ogólny:

1. Uruchomienie programu
2. Wprowadzenie nicku
3. Wprowadzenie wiadomości
4. Uzyskanie wygenerowanej odpowiedzi
5. Uzyskanie wygenerowanych statystyk

6 TESTOWANIE.

6.1 Ogólny przebieg testowania:

Program będzie testowany za pomocą programu eclipse, który będzie znajdował poszczególne błędy. Będzie również testowany na plikach różnej wielkości. Program zostanie również przetestowany za pomocą testów jednostkowych JUnit. Przykład testu dla klasy Read.

```
public class ReadTest
```

```
@Test
```

```
public void testRead()
```

```
String tekst = "Alibaba i czterdziestu rozbójników buszuj. Alibaba i 12";
```

```
Read r = new Read( tekst, true);
```

```
assertEquals(r.listaNgram.get(0).slowo, "Alibaba i");
```

```
assertEquals(r.listaNgram.get(1).slowo, "i czterdziestu");
```

```
assertEquals(r.listaNgram.get(2).slowo, "czterdziestu rozbójników");
```

```
assertEquals(r.listaNgram.get(3).slowo, "rozbójników buszuj.");
```

```
assertEquals(r.listaNgram.get(4).slowo, "buszuj. Alibaba");
```

```
assertEquals(r.listaNgram.get(0).nastepcy[0], "czterdziestu");
```

```
assertEquals(r.listaNgram.get(0).nastepcy[1], "12");
```

```
assertEquals(r.listaNgram.get(1).nastepcy[0], "rozbójników");
```

```
assertEquals(r.listaNgram.get(2).nastepcy[0], "buszuj.");
```

```
assertEquals(r.listaNgram.get(3).nastepcy[0], "Alibaba");
```

```
assertEquals(r.listaNgram.get(4).nastepcy[0], "i");
```

7 DIAGRAM UML

