System wyboru najodpowiedniejszego języka programowania dla danej osoby na podstawie jej zainteresowań

Dawid Mikowski 251674

# 1.Wstęp

Tworzona aplikacja jest to system ekspertowy pozwalający uzytkownikowi na ułatwienie podjęcia decyzji, którego języka programowania warto się nauczyć na podstawie pytań o jego preferencje zainteresowania i motywacje. Aplikacja jest napisana w języku C# wykorzystującym silnik graficzny WPF do stworzenia intuicyjnego interfejsu użytkownika. Działa ona na wszystkich komputerach z systemem operacyjnym windows.

# 2.Wymagania Funkcjonalne

Tworzony system musi obsługiwać odpowiedzi użytkownika w taki sposób aby dotarł on do wyniku a także przechowywać bazę wiedzy w taki sposób by użytkownik nie miał do niej dostępu z poziomu aplikacji ale także by mogła ona z łatwością zostać do niej wczytana. Z powodu wymagań przedmiotu musi on zawierać ponad 50 reguł.

|  |
| --- |
| **Przypadek użycia:** Wykonanie testu |
| **Aktorzy:** Użytkownik |
| **Scenariusz Głowny:** |
| 1. Użytkownik uruchamia aplikacje 2. Użytkownik odpowiada na serię pytań 3. Użytkownik dostaje odpowiedź na podstawie jego odpowiedzi |

# 3.Wymagania Niefunkcjonalne

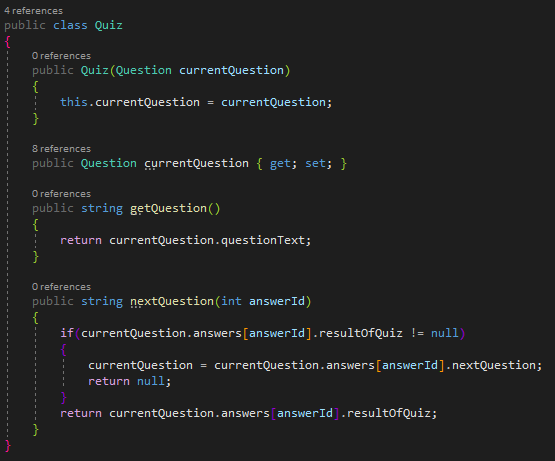
Głowną grupą docelową są programiści, którzy nie mogą podjąć decyzji jakiego języka programowania się nauczyć. Interfejs użytkownika musi być prosty i nie powinno wymagać tłumaczenia. Aplikacja powinna działać na wszystkich urządzeniach z zainstalowanym systemem operacyjnym Windows i nie powinna potrzebować instalacji żadnego innego oprogramowania do poprawnego działania.

# 4.Projekt Aplikacji

//todo wkleic uml diagram klas

Głowna funkcjonalność programu zawarta jest w trzech klasach Quiz, Question i Answer. Do przetłumaczenia pliku w formacie JSON na odpowiednie obiekty wykorzystano klasę pomocniczą JsonParser.

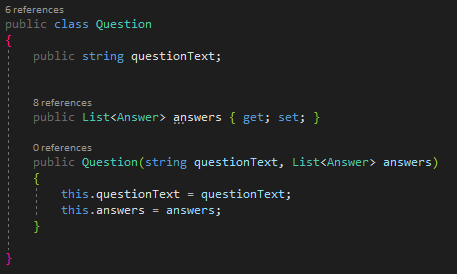
1. Quiz



Klasa Quiz zawiera jedno pole currentQuestion oraz 2 metody getQuestion i nextQuestion

* **currentQuestion** – przechowuje referencje do obiektu pytania na którym aktualnie jest użytkownik
* **getQuestion** – pobiera z obiektu aktualnego pytania jego treść
* **nextQuestion** – jeżeli obiekt aktualnego pytania nie posiada wyniku quizu to zamienia ją na kolejne pytanie w przeciwnym przypadku zwraca wynik testu

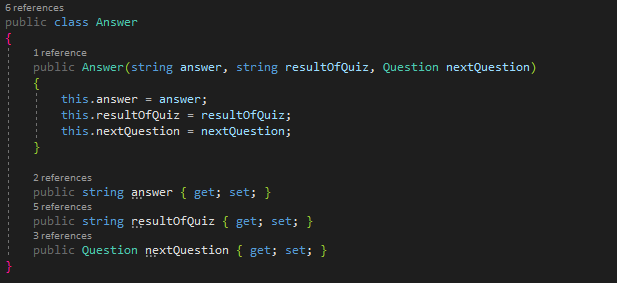
1. Question



Klasa Question zawiera dwa pola questionText oraz Answers

* **questionText** – jest to treść pytania
* **answers** – jest to lista możliwych odpowiedzi na pytanie

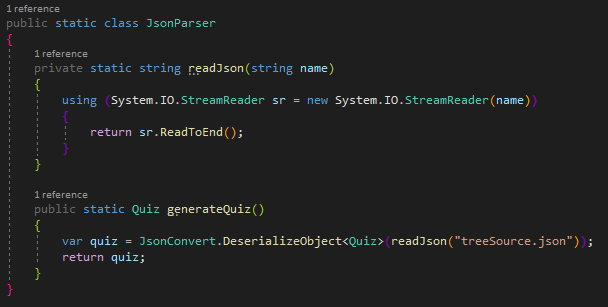
1. Answer



Klasa Answer zawiera 3 pola answer, resultOfQuiz oraz nextQuestion

* **answer** – jest to treść odpowiedzi
* **resultOfQuiz** – jeżeli jest to ostatnie pytanie to zawiera treść wyniku quizu
* **nextQuestion** – jest to referencja do kolejnego pytania które zostanie zadane w przypadku wybrania tej odpowiedzi

1. JsonParser



Klasa JsonParser zawiera 2 metody readJson oraz generateQuiz

* **readJson –** zwraca ona cała zawartość pliku Json o podanej nazwie
* **generateQuiz –** tworzy potrzebną strukturę obiektów na podstawie wczytanego pliku Json wykorzystuje ona bibliotekę JsonConvert

# 5.Wykorzystane Technologie

Aplikacja została napisana w języku C# w wersji .net Framework 4.7.2