

**Uniwersytet Warszawski**  
Wydział Matematyki, Informatyki i Mechaniki

**Marcin Mieteń**

Nr albumu: 248 266

# **Implementacja QWords - słownika i systemu powtórek**

**Praca licencjacka  
na kierunku INFORMATYKA**

Praca wykonana pod kierunkiem  
**magistra Łukasza Kamińskiego**  
Instytut Informatyki

Wrzesień 2009

## **Oświadczenie kierującego pracą**

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis kierującego pracą

## **Oświadczenie autora (autorów) pracy**

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data

Podpis autora (autorów) pracy

## **Streszczenie**

W niniejszej pracy przedstawiono projekt oraz implementację wielojęzycznego słownika ze zintegrowanym systemem powtórek. Program wspomaga użytkownika w procesie nauki poprzez przyspieszenie tłumaczenia, zapamiętywanie słówek, przeprowadzanie powtórek, drukowanie flashcards oraz zarządzanie zapamiętanym słownictwem. Aplikacja została zaimplementowana w języku C++ z użyciem Qt4, WinAPI oraz SQLite. Do zarządzania źródłami oraz procesem powstawania systemu wykorzystany został rozproszony system kontroli wersji Mercurial oraz serwis Assembla.com.

## **Słowa kluczowe**

słownik, system powtórek, spaced repetition, flash-cards, Langenscheidt, Qt4, Mercurial

## **Dziedzina pracy (kody wg programu Socrates-Erasmus)**

11.3 Informatyka

## **Klasyfikacja tematyczna**

D. Software

D.1. Programming Techniques

D.1.5. Object-oriented Programming

## **Tytuł pracy w języku angielskim**

Implementation of QWords - dictionary and repetition system



# Spis treści

<b>1. Wprowadzenie</b> . . . . .	5
<b>2. Opis projektu</b> . . . . .	7
2.1. Współpraca z wydawnictwem Langenscheidt . . . . .	7
2.2. Ogólna architektura i technologie . . . . .	8
2.3. Najważniejsze funkcje . . . . .	8
2.3.1. Tłumaczenie za pomocą skrótu klawiszowego . . . . .	8
2.3.2. Zachowanie w przypadku nieodnalezienia tłumaczenia . . . . .	9
2.3.3. Zapisywanie tłumaczeń do lokalnej bazy . . . . .	9
2.3.4. Użytkownicy . . . . .	9
2.3.5. Obsługa słownika w formacie XML . . . . .	9
2.3.6. Algorytm pedagogiczny . . . . .	10
2.3.7. Statystyki . . . . .	10
2.3.8. Baza Danych . . . . .	11
2.3.9. Zarządzanie słownictwem . . . . .	11
2.3.10. Flash-cards i testy na życzenie . . . . .	12
2.3.11. System wtyczek . . . . .	12
2.3.12. Ustawienia . . . . .	12
<b>3. Mój wkład pracy</b> . . . . .	15
<b>4. Podsumowanie</b> . . . . .	17
<b>5. Zawartość dołączonej płyty</b> . . . . .	19



# Rozdział 1

## Wprowadzenie

Znajomość języków obcych była zawsze pożądaną umiejętnością dla każdego wykształconego człowieka. Umiejętności lingwistyczne stały się jeszcze bardziej istotne w zglobalizowanym, multikulturowym i wielojęzycznym świecie. Nauka języków wymaga jednak czasu i przede wszystkim systematyczności, cechy, której często brakuje współczesnemu człowiekowi. Uczestnictwo w kursach językowych zazwyczaj wiąże się z dużymi kosztami, natomiast samodzielna nauka, mimo iż przynosi chwilowe efekty, nie gwarantuje właściwego utrwalenia materiału. My sami, jako studenci informatyki, często borykaliśmy się z podobnymi problemami. Oczywiście wydaje się, że pomocny w nauce języka obcego może być komputer, jednakże żadne z istniejących rozwiązań nie było dla nas satysfakcjonujące.

Dzięki upowszechnieniu się dostępu do Internetu powstało wiele programów do nauki języka, które korzystają z zewnętrznych baz danych, dzięki czemu użytkownik może zaoszczędzić miejsce na własnym dysku twardym. Wadą tego rozwiązania jest to, że usługa staje się bezużyteczna, jeśli nie posiadamy aktualnie dostępu do sieci. Z kolei słowniki stacjonarne są rzadko aktualizowane, co często może nawet uniemożliwić właściwe zrozumienie czytanego tekstu. Kolejną słabą stroną wielu testowanych przez nas rozwiązań jest konieczność ręcznego kopiowania frazy, którą chcemy przetłumaczyć, do danego programu. Istnieją programy, które w pewien sposób radzą sobie i z tym problemem. Program Babylon umożliwia szybkie tłumaczenie poprzez zaznaczenie słowa i naciśnięcie odpowiedniego skrótu klawiszowego. Po tej operacji na ekranie pojawia się okienko z tłumaczeniem. Inne rozwiązanie zostało zastosowane w programie kypdict. Po zaznaczeniu danej frazy, automatycznie pojawi się ona w oknie programu. Oba wspomniane programy wymagają, jednakże nakładów finansowych - kypdict zakupu słownika wydawnictwa Young Digital Poland, natomiast za Babylon 8.0 wraz z słownikiem Oxford Dictionary należy zapłacić około stu dolarów. Jednakże nie to jest największym problemem. Babylon działa jedynie pod systemem operacyjnym Windows, natomiast kypdict pod systemem Linux. Ponadto żaden z nich nie umożliwia zapamiętywania wybranego słownictwa i w związku z tym nie wspomaga użytkownika w nauce.

Do powtarzania zapamiętanego słownictwa można używać programu Anki. Jest to program do obsługi tzw. fiszek (ang. flashcards) - kartek dotyczących danego hasła i zawierających związane z nim notatki. Fiszki są często używane w procesie systematycznego uczenia się. Anki korzysta ze specjalnego algorytmu (ang. space repetition system), który przypomina o danym słówku w najlepszym dla uczącego się momencie. Czas powtórki wybierany jest przy użyciu znanego z psychologii tzw. „efektu przerwy” (ang. spacing effect). Efekt ten odnosi się do faktu, iż człowiek łatwiej zapamiętuje materiał powtarzając go kilka razy w dłuższych odstępach czasu, niż robiąc to wielokrotnie w krótkich odstępach. Główną wadą Anki jest to, że materiał do powtórek trzeba wprowadzać ręcznie, co jest szczególnie niewygodne przy

czytaniu dłuższego tekstu, w którym czytelnik nie rozumie wielu słów.

Celem projektu QWords było rozwiązanie opisanych problemów poprzez stworzenie programu, który maksymalnie ułatwiałby poznawanie i naukę nowych słówek w obcym języku. W rozdziale pierwszym został przedstawiony szczegółowy opis stworzonej aplikacji. O moim wkładzie pracy i problemach, które udało mi się rozwiązać traktuje rozdział drugi. Rozdział trzeci zawiera podsumowanie, natomiast czwarty opis płyty CD dołączonej do pracy.



## Rozdział 2

# Opis projektu

QWords to aplikacja działająca pod systemami z rodziny Windows i Linuks, a jej głównym celem jest maksymalne ułatwienie poznawania i nauki słówek w obcym języku poprzez zintegrowanie różnorodnych słowników i systemu powtórek. QWords pracuje w tacce systemowej, a cała funkcjonalność skupia się na następującym przypadku użycia: użytkownik czytając dowolny tekst (np. w przeglądarce internetowej) chce przetłumaczyć nieznaną sobie słowo a następnie zapisać je w celu późniejszych powtórek i docelowego trwałego zapamiętania danego słowa. Aby przetłumaczyć dane słowa wystarczy zaznaczyć tekst i wcisnąć odpowiedni skrót klawiszowy. Możliwa jest również opcja ręcznego wprowadzenia szukanego słowa w pole edycji. Tłumaczenie pojawia się w małym oknie z wynikami z możliwością zapisania nieznanych użytkownikowi słów w lokalnej bazie. QWords dba o regularne utrwalanie wiedzy przypominając o danych słówkach do powtórzenia w momencie najkorzystniejszym dla użytkownika. Program udostępnia również inne funkcje takie jak przeglądanie statystyk nauki, drukowanie flash-cards i generowanie testów na życzenie. Wszystkie wymienione funkcjonalności zostały szczegółowo opisane w dalszej części tej pracy.

### 2.1. Współpraca z wydawnictwem Langenscheidt

Zespół tworzący QWords rozważał różne sposoby dystrybucji tworzonego oprogramowania. Pierwszym pomysłem było umieszczenie QWords w Internecie wraz z jego kodem źródłowym na zasadach licencji GNU GPL. Później pojawił się jednak ciekawszy pomysł. Podjęliśmy decyzję o chęci wydania programu na płycie dołączanej do ogólnopolskiej gazety. Aby ten cel urzeczywistnić odbyliśmy serię spotkań w wydawcami, aby zaprezentować swój pomysł. Pierwsze spotkanie odbyliśmy z panią Anną Kozakiewicz, która pełni funkcję Dyrektora Promocji Sprzedaży Gazety Prawnej. Podczas rozmowy zidentyfikowaliśmy podstawowy problem, który uniemożliwiłby akceptację programu przez wydawcę gazety. Jest to uzależnienie QWords od słowników internetowych i ich niski poziom merytoryczny. Okazało się, że niezbędne będzie pozyskanie bazy danych słownika renomowanego wydawnictwa, która będzie umieszcza na płycie CD, czyniąc QWords niezależnym od połączenia z Internetem.

W związku z koniecznością rozwiązania powyższego problemu skontaktowaliśmy z panią Agnieszką Mizak, która pełni funkcję Dyrektora Programowego w wydawnictwie Langenscheidt. Przedstawiliśmy jej koncepcję oprogramowania i plany jego wydania jako dodatek do gazety. Uzyskaliśmy wstępną zgodę na współpracę, jednak konieczne było spotkanie wszystkich zainteresowanych stron w celu ustalenia szczegółów. Spotkanie takie odbyło się w siedzibie Gazety Prawnej, uczestniczyli w nim: Mariusz Łukasik (Gazeta Prawna), Agnieszka Mizak i Norbert Słupski (Langenscheidt) oraz Julian Jezierski i Michał Świtakowski (zespół QWords).

Podczas spotkania ustaliliśmy podstawowy plan przygotowań programu do wydania. Zakładał on ścisłą współpracę między wydawnictwem Langenscheidt i zespołem QWords w celu dostosowania oprogramowania do obsługi szkolnego słownika angielsko-polskiego Langenscheidt. Gotowy produkt miał zostać zaprezentowany Gazecie Prawnej i przygotowany do wydania na płycie CD.

W czasie tworzenia oprogramowania regularnie kontaktowaliśmy z przedstawicielami Langenscheidt w celu ustalenia wspólnej wizji różnych elementów aplikacji. Odbyliśmy także 3 spotkania mające na celu zaprezentowanie postępu prac i zebranie sugestii przydatnych przy dalszym rozwoju aplikacji. Bardzo pomocne były uwagi dotyczące obsługi dostarczonego przez Langenscheidt słownika. Pomogły one nam na prawidłowy podział słownictwa ze względu na jego

QWords posiadał w pełni sprawną obsługę słownika Langenscheidt w kwietniu 2009. W związku z tym w maju spotkaliśmy się z Mariuszem Łukasikiem prezentując gotowy program. Gazeta Prawna zobowiązała się do zaakceptowania lub odrzucenia oferty. Niestety w momencie tworzenia tego dokumentu takiej odpowiedzi nie uzyskaliśmy. Wydaje się mało prawdopodobne, aby Gazeta Prawna w tej chwili zdecydowała się na współpracę. Wynika to z tego, iż właśnie trwa proces łączenia redakcji Gazety Prawnej i Dziennika.

W związku z wątpliwą

## 2.2. Ogólna architektura i technologie

Produktem finalnym naszej pracy są źródła programu do kompilacji w systemie Linux i Windows oraz instalator dla systemu Windows. W procesie tworzenia źródeł programu oraz dokumentacji użyte zostały następujące technologie, mechanizmy i programy:

- Qt 4 - zestaw przenośnych bibliotek i narzędzi programistycznych,
- Qt Creator - zintegrowane środowisko programistycznych,
- Mercurial - rozproszony system kontroli wersji ,
- Assembla.com - serwis przyspieszający i ułatwiający tworzenie oprogramowania,
- SQLite - system zarządzania bazą danych,
- GlobalShortCutManager - biblioteka Open Source dodająca do Qt obsługę globalnych skrótów klawiszowych,
- WinApi - interfejs programistyczny systemu Windows, użyte zostały funkcje do emulacji fizycznego naciskania skrótów,
- Adobe Flash - program użyty do stworzenia prezentacji QWords,
- Gobby - program do wspólnego tworzenia dokumentacji (ang. collaborative text editor),
- BitRock InstallBuilder - program do tworzenia instalatorów dla systemu Windows.

## 2.3. Najważniejsze funkcje

### 2.3.1. Tłumaczenie za pomocą skrótu klawiszowego

Jedną z istotniejszych funkcji QWords jest możliwość szybkiego tłumaczenia przy użyciu globalnych skrótów klawiszowych. Aby przetłumaczyć daną frazę (może to być dowolny tekst

w dokumencie tekstowym, oknie przeglądarki internetowej, itd.), użytkownik musi ją zaznaczyć, a następnie nacisnąć odpowiedni skrót klawiszowy (domyślnie Win+Z). Po wykonaniu tej czynności, w miejscu gdzie znajduje się kursor myszy, pojawi się okno z wynikami tłumaczenia zaznaczonej wcześniej frazy. Jeżeli użytkownik chce tylko uzyskać dostęp do okna ze słownikiem, musi nacisnąć inny skrót (domyślnie Win+A). Ponadto w ustawieniach programu istnieje możliwość zmiany domyślnych skrótów klawiszowych.

### **2.3.2. Zachowanie w przypadku nieodnalezienia tłumaczenia**

W przypadku gdy słownik nie posiada danego słowa wyświetlone zostaje tłumaczenie dla słowa najbardziej zbliżonego. Jednocześnie pole edycji, w którym wprowadzone jest szukane słowo podświetla się w kolorze czerwonym. Takie zachowanie jest łatwo dostrzegalnym sygnałem dla użytkownika, który posiada do dyspozycji listę alfabetycznie ułożonych słów udostępnianych przez obsługiwane słowniki. Lista ta pozwala nam w szybki sposób skorygować niepoprawne wyszukanie, gdyż użytkownik widzi kilka dostępnych pozycji znajdujących się przed i po szukanej frazie. Dodatkową pomocną opcją jest możliwość filtrowania listy dostępnych słów (np. jeśli pamiętamy tylko część szukanego słowa).

### **2.3.3. Zapisywanie tłumaczeń do lokalnej bazy**

W momencie przetłumaczenia danego słowa użytkownik otrzymuje listę tłumaczeń i przykładów użycia. Każdy element takiej listy można niezależnie zapamiętać do lokalnej bazy klikając w odpowiedni przycisk. Istnieje również możliwość zapisania wszystkich elementów listy za pomocą jednego kliknięcia.

### **2.3.4. Użytkownicy**

Od samego początku tworzenia QWords zakładaliśmy, że program powinien umożliwiać zakładanie kont użytkowników. Wynika to z kilku faktów. Przede wszystkim, każdy użytkownik ma inny zasób znanego słownictwa, a zatem chciałby dołączać do bazy słówek tylko te, których nie zna. Dodatkowo, algorytm pedagogiczny nie będzie działał zgodnie z założeniami, jeśli dopuścimy wykonywanie powtórek z tej samej bazy słownictwa przez kilka osób. Niestety, nie mogliśmy polegać na kontach użytkowników, które zapewniają systemy operacyjne Windows i Linux. Z naszego doświadczenia niewiele komputerów posiada poprawnie skonfigurowane konta dla wielu użytkowników - zwykle istnieje tylko jedno konto dla wszystkich.

W związku z tym wymaganiem wszystkie komponenty oprogramowania były tworzone tak, aby mogły być użyte w przypadku istnienia kilku użytkowników. Następnie został zaimplementowany system kont, który w pełni separuje dane oraz ustawienia każdego użytkownika. Po uruchomieniu QWords użytkownik może wybrać swój profil. Dzięki temu wszystkie przetłumaczone słowa trafiają do jego osobistej bazy słownictwa, a generowane powtórki dotyczą tylko słówek, które go interesują. Co ważne, zmiana profilu może być wykonana praktycznie natychmiastowo i bez konieczności zamykania aplikacji.

### **2.3.5. Obsługa słownika w formacie XML**

Programu QWords do tłumaczenia słów z języka angielskiego na język polski wykorzystuje dostarczony przez wydawnictwo Langenscheidt słownik w formacie XML. Odpowiednie parsowanie zrealizowane za pomocą maszyny stanowej i wspomagane standardowymi metodami przetwarzania dokumentów XML pozwoliło wydobyć interesujące nas tłumaczenia. Proces ten wymagał rozdzielenia dla każdego hasła istotnie różnych tłumaczeń i przykładów użycia

danego słowa. Precyzyjność tego podziału była w naszym przypadku bardzo istotna z uwagi na możliwość zapisywania oddzielnie każdego tak wydzielonego tłumaczenia (w celu późniejszego odpytania). Również sam algorytm pedagogiczny (opisany w sekcji 2.3.6) bazuje na odpytywaniu nas z konkretnego tłumaczenia danego słowa, a nie z wszystkich możliwych tłumaczeń i przykładów użycia. Potencjalnie QWords zaprojektowany został z myślą obsługi wielu słowników. Z uwagi na to obsługa słownika zaimplementowana jest w odpowiedniej dynamicznie ładowanej wtyczce do programu. Mechanizm ten opisano dokładniej w sekcji 2.3.11

### 2.3.6. Algorytm pedagogiczny

Dobór wyrażen do powtórki nie jest przypadkowy. Odpowiedzialna jest za niego pedagogiczna metoda zaimplementowana w programie. Jest to algorytm SM-2 z pewnymi usprawnieniami. SM-2 jest sprawdzonym już w przeszłości sposobem powtarzania materiału. Moglibyśmy wybrać jego nowsze wersje, ale oceniliśmy, że dają nikłe korzyści, a koszt ich zrealizowania jest nieporównywalnie wyższy. Wynika on z skomplikowanej implementacji połączonej z brakiem możliwości oceny jej poprawności, gdyż ciężko dokładnie przewidzieć odległości między powtórkami. Jesteśmy w stanie zaufać algorytmowi SM-2 między innymi dlatego, że znalazł on miejsce w innych szeroko stosowanych aplikacjach do nauki (np: „Anki”). Algorytm użyty w QWords ma kilka istotnych cech:

- mierzy czas w jednostkach dziennych - czyli po wykonaniu całej powtórki możemy się już nie spodziewać więcej przypominania,
- powtórka nie skończy się dopóki każdej odpowiedzi nie będziemy sobie w stanie przypomnieć bez jej oglądania - innymi słowy każda rzecz jest powtarzana w obrębie powtórki do momentu stwierdzenia, że jesteśmy w stanie przypomnieć odpowiedź bez jej podglądania,
- częstość przypominania każdej rzeczy zależy od tego, czy dobrze ją pamiętaliśmy podczas powtórek, do tego zależy od preferencji zapisanych w ustawieniach programu,
- można się spodziewać, że już dobrze pamiętana rzecz będzie przypominana coraz rzadziej,
- algorytm może zacząć się mylić oceniając naszą wiedzę jeśli powtórki są przez wiele dni ignorowane, jednak sytuacja automatycznie się poprawia, jeśli znów zaczniemy na nie odpowiadać.

Choć nasze przełożenie algorytmu SM-2 na program jest wierne, to jego interakcja z użytkownikiem różni się nieco od oryginału. Otóż autor SM-2 przewiduje sześć różnych reakcji użytkownika na każdym kroku, my w celu uczynienia aplikacji bardziej przejrzystą, zmniejszyliśmy tę liczbę do pięciu. Zabraliśmy użytkownikowi jedną opcję semantycznie bardzo podobną do jednej z sąsiednich, co pozwoliło w interfejsie na mniej opisowe, a wciąż łatwe w zrozumieniu nazwy opcji (oryginalne nawet opisane mogą być niejasne dla wielu ludzi). Zdajemy sobie sprawę, że nieco ryzykujemy skuteczność algorytmu, ale to istotnie zwiększyło jej atrakcyjność w oczach potencjalnych klientów.

### 2.3.7. Statystyki

Aplikacja QWords, pozwala nam sprawdzać nasze postępy w nauce. Dla każdego użytkownika prowadzona jest historia. Program zbiera statystyki z powtórek i zestawia je w formie

wykresów. Każdy z nich pozwala na wybór z jak długiego okresu chcemy dane - z tygodnia, miesiąca, roku lub wszystkie dostępne dane. Rodzaje wykresów:

- wykres historii powtórek pokazujący ile słówek powtórzyliśmy względem czasu,
- wykres zaplanowanych już powtórek pokazujący ile słówek czeka nas w przyszłości względem czasu,
- wykres zależności ilości wszystkich słówek, które sobie utrwalamy względem czasu,
- wykres zależności średniej zapamiętania wszystkich słówek, które sobie utrwalamy względem czasu.

### 2.3.8. Baza Danych

Wszystkie dane dotyczące naszej nauki znajdują się w bazie danych. Wybraliśmy SQL-Lite'a ponieważ nie zależy nam na wyrafinowanych systemach zarządzania bazą, gdyż aplikacja na raz jest użytkowana przez jedną osobę, a więc można się spodziewać małego jej obciążenia zapytaniami. Decyzja ta wpływa na lekkość i szybkość programu. Do bazy danych napisany jest moduł pozwalający pozostałym modułom abstrahować od tego skąd pochodzą dane - tj. zwalnia je z pisania zapytań. Zamiast tego oferuje zbiór funkcji dla potrzebnych lub potencjalnie przydatnych operacji na słówkach i innych danych. Każdy użytkownik posiada własną bazę danych, którą może przenosić niezależnie od innych.

Tabele:

- tabela słówek - oprócz pożądanego do nauczania wyrażenia i ich tłumaczeń, znajdują się tam wszystkie informacje i obliczone współczynniki niezbędne algorytmowi do doboru powtórek. Dodatkowo każde słowo ma kilka dodatkowych informacji takich jak tagi i dane związane ze słownikiem, z którego pochodzi,
- tabela historii - są tam codziennie katalogowane informacje związane ze stanem naszej nauki, kolejne kolumny to określone agregacje z tabeli słówek potrzebne modułowi statystyk do rysowania wykresów,
- tabela zbiorów - lista zbiorów z podstawowymi danymi o istniejących zestawach wyrażań do powtórek na żądanie,
- tabela relacji wiele-do-wielu między zestawami, a słówkami.

### 2.3.9. Zarządzanie słownictwem

Aplikacja daje możliwość zarządzania słownictwem, które znajduje się bądź ma się znaleźć w lokalnej bazie danych użytkownika. Korzystający z aplikacji może obejrzeć wszystkie słówka aktualnie dodane do bazy, mając przy tym możliwość posortowania ich według własnych preferencji, a także filtrowania według wybranych kryteriów. W tabeli widoczne jest słowo, jego tłumaczenie oraz informacja o poziomie zapamiętania. Użytkownik posiada również możliwość dodania nowego słówka do bazy, ręcznie wpisując dany wyraz wraz z jego tłumaczeniem. Istnieje także możliwość edycji dodanych już słów. Użytkownik może zmienić bądź dodać informacje o słówku, tłumaczeniu i kategoriach (tags), do których wyraz należy, a także usunąć z bazy dowolne wpisy. Istnieje możliwość cofnięcia wprowadzonych zmian, która znacznie zwiększa wygodę użytkownika.

### 2.3.10. Flash-cards i testy na życzenie

Wszystkie posiadane w lokalnej bazie słówka można łączyć w zestawy. Do zestawu mogą być włączane słówka z danej kategorii, słówka, które użytkownik ma zamiar w najbliższym czasie powtarzać, czy wreszcie po prostu wszystkie słówka, znajdujące się w bazie danych. Utworzone zestawy można wykorzystywać do przeprowadzania testów na życzenie oraz do drukowania karteczek ze słówkami, ułatwiających naukę z dala od komputera, tzw. flash-cards.

Test na życzenie wygląda tak samo jak powtórka przeprowadzana przez algorytm pedagogiczny, z tą różnicą, że słówka, z których jest przepytany użytkownik pochodzą z wybranego, zdefiniowanego wcześniej zestawu. Użytkownik musi pamiętać o tym, że wyniki z testów na życzenie nie są uwzględniane przez algorytm pedagogiczny przy określaniu poziomu zapamiętania słówek.

Drukując zestaw użytkownik definiuje rodzaj wydruku oraz rozmiar drukowanych karteczek. Istnieje możliwość drukowania flash-cards w postaci jedno lub dwustronnych karteczek, które można sobie po wydrukowaniu wyciąć, a także przedstawienie słówek w postaci listy, która trzymana jest na jednej większej kartce. Przed przystąpieniem do drukowania użytkownik może obejrzeć podgląd wydruku i sprawdzić, czy wybrana konfiguracja odpowiada jego preferencjom.

### 2.3.11. System wtyczek

Aby umożliwić łatwiejszy rozwój aplikacji w przyszłości QWords jest zbudowane w oparciu o system wtyczek. Każda wtyczka odpowiada za obsługę jednego słownika lub grupy słowników. Dzięki takiemu podejściu aplikacja może zostać poszerzona o obsługę nowego słownika bez modyfikacji jej kodu i bez udziału jej autorów. Każdy programista może zapoznać się z ustalonym interfejsem (API), który musi implementować wtyczka i napisać odpowiedni kod źródłowy.

Tworząc wspomniany system posłużyliśmy się modulem Qt Plugins, który jest częścią opisaną w sekcji 2.2 biblioteki Qt. Jego główną zaletą jest takie samo działanie bez względu na środowisko, w którym zostanie użyty. Każdy system operacyjny używa innego rodzaju dynamicznych bibliotek, przykładowo w Windows są to pliki `.dll`, a pod Linuksem pliki `.so`. Różnice pojawiają się także w sposobie ich ładowania i usuwania z pamięci. Użycie Qt Plugins pozwoliło na skupienie się na stworzeniu sprawnego systemu wtyczek bez zagłębiania się w szczegóły techniczne każdego systemu operacyjnego.

### 2.3.12. Ustawienia

Z każdym profilem użytkownika (mechanizm użytkowników opisano w sekcji 2.3.4) ściśle powiązany jest specjalny plik z ustawieniami. Rozpoczynając pracę z nową aplikacją każdy lubi przystosować ją do własnych preferencji. W QWords staramy się możliwie ułatwić proces personalizacji aplikacji. W celu ułatwienia tego procesu wyszczególniliśmy dwa sposoby zmiany różnego rodzaju ustawień. Pierwszą klasą są ustawienia, które modyfikuje się bezpośrednio przez GUI samej aplikacji, które zapamiętują się automatycznie:

- rozmiar głównego okna aplikacji
- rozmiar okna tłumaczenia
- tryb okna tłumaczenia (z lub bez listy haseł dostępnych w słowniku)
- szerokość listy haseł w oknie tłumaczenia

- czy okno tłumaczenia ma być zawsze ponad innymi oknami
- sposób wyszukiwania (z lub bez filtrowania listy haseł)

Drugą klasę ustawień modyfikuje się poprzez specjalne okno konfiguracji. Okno to podzielone jest na trzy zakładki, każda powiązana z odpowiednią kategorią ustawień.

- ustawienia główne
  - skrót klawiszowy pokazujący okno tłumaczenia
  - skrót klawiszowy tłumaczący zaznaczoną frazę
  - motyw wyglądu aplikacji (tzw. skóra)
- ustawienia powtórek
  - dolne ograniczeni na czas co jaki pojawiać się będzie przypomnienie o powtórkach
  - maksymalna ilość słów do powtórzenie w ciągu dnia
  - intensywność nauki
- ustawienia okna tłumaczenia
  - rozmiar czcionki w oknie tłumaczenia





## Rozdział 3

# Mój wkład pracy

W wyniku podziału obowiązków przypadło mi w udziale stworzenie wstępnej wersji zakładki zarządzania słownictwem, która znalazła się w pierwszej, testowej wersji programu. Podczas tej implementacji powstała klasa delegata tabeli wyświetlającej zawartość bazy danych, która została wykorzystana w finalnej wersji aplikacji. Oprócz tego miałem także stworzyć użytkownikowi możliwość drukowania w łatwy i wygodny sposób flash-cards, a także udostępnić możliwość oglądania statystyk zaprezentowanych w formie czytelnych wykresów. Poniżej postaram się dokładniej przybliżyć jak wyglądał i na czym polegał proces implementacji podanych przeze mnie funkcji programu.

Tabela zawierająca listę słówek występuje w aplikacji w więcej niż jednym miejscu, dlatego dobrym i bardzo wygodnym rozwiązaniem było stworzenie specjalnej klasy delegata, która odpowiadałaby za poprawne wypełnienie zawartości tabeli i którą można by bez problemu wykorzystać w wielu miejscach kodu. Dodatkowo w polach tabeli miało pojawić się coś więcej, niż tylko niesformatowany tekst, a mianowicie rich text oraz graficznie prezentowany pasek postępu, informujący o poziomie zapamiętania słówka. Możliwość użycia takiej niestandardowej zawartości dawała właśnie klasa delegata.

Klasa delegata tabeli słówek do poprawnego działania wymagała zaimplementowania konkretnych metod, a mianowicie metody `paint` oraz `sizeHint`. Zadaniem pierwszej z metod, było poprawne wypełnienie zawartości poszczególnych komórek. Do użycia niestandardowych rozwiązań zmuszał fakt, iż w tabeli w kolumnach ze słówkiem i jego tłumaczeniem miał być wyświetlany rich text, natomiast w komórce z postępem nauki, graficzny element w postaci paska zapamiętania słówka. Konieczne było zaimplementowanie metody, która w poprawny sposób wyświetliłaby sformatowany tekst i zrobi to, nie przekraczając szerokości kolumny. Zadanie to, okazało się jednak łatwe do rozwiązania, ze względu na istnienie klasy `QTextDocument`, która pozwalała na wyświetlenie sformatowanego tekstu na obszarze o zadanej szerokości. Narysowanie paska postępu wymagało zaś użycia klasy `QPainter` i zdefiniowania metody, która odpowiednio rozmieściłaby i pokolorowała prostokąty na zadanym obszarze. Odpowiednia metoda rysująca zawartość komórki była wybierana na podstawie typu kolumny. Dodatkowo trzeba było też sprawdzić, czy dany wiersz jest zaznaczony, czy nie i odpowiednio na to zareagować, podświetlając go, bądź usuwając podświetlenie. Do zakodowania pozostawała jeszcze metoda `sizeHint`, której zadaniem było ustawienie odpowiedniej wysokości wiersza. Wartość ta była ustalana na podstawie znanej szerokości kolumny, tekstu, który był wpisywany w komórkę, oraz czcionki, która była używana. Metoda zwracała wysokość prostokąta o danej szerokości, w którym dany tekst, pisany konkretną czcionką się mieścił. Zaimplementowana w taki sposób klasa delegata pozwoliła na uniknięcie powtarzania w kilku miejscach tego samego kodu.

Nasza aplikacja udostępnia możliwość drukowania flash-cards i to ja byłem odpowiedzialny za zaimplementowanie tej funkcji w programie. Na początku zastanawialiśmy się, jakie opcje powinny być udostępnione użytkownikowi, żeby można było wygodnie i bezproblemowo drukować fiszki. Doszliśmy do wniosku, że dobrze będzie, jeżeli w aplikacji pojawi się opcja drukowania trzech rodzajów flash-cards, oraz możliwość dostosowania rozmiaru drukowanych karteczek. Stwierdziliśmy, że w ten sposób łatwiej będzie użytkownikowi wybrać sposób drukowania odpowiadający jego preferencjom. Dobrym pomysłem było też umieszczenie funkcji podglądu wydruku, dzięki której można jeszcze łatwiej dostosować drukowane flash-cards do wymagań użytkownika. Wszystkie opisane funkcje udostępnia klasa `Print`. Po zaimplementowaniu klasy przeprowadziłem testy najpierw na wirtualnej, a później na prawdziwej drukarce.

Podstawową metodę klasy stanowi `slot paintRequest`. Metoda ta wywoływana jest, kiedy nadchodzi zadanie drukowania bądź wyświetlenia podglądu. Zaimplementowanie tej metody pozwoliło na uniknięcie konieczności wyodrębniania tych dwóch sytuacji. W obu przypadkach po wywołaniu przebieg programu jest ten sam. Różni się jedynie miejsce wywołania tej metody. Po aktywacji slotu wywołuje się konkretna metoda rysująca, w zależności od tego, który rodzaj drukowania został wybrany przez użytkownika. Istnieją trzy typy druku, jednostronne i dwustronne drukowanie karteczek, bądź drukowanie listy. Dalej uwzględniana jest wielkość karteczek, jaka została ustawiona. W przypadku drukowania listy decyduje ona o ilości kolumn na liście, w pozostałych dwóch przypadkach określa rozmiar pojedynczej karteczki. Jako że wielkości karteczek są z góry zdefiniowane, trzeba było poradzić sobie z problemem różnej ilości znaków w słówkach i tłumaczeniach. Przy rysowaniu zawartości karteczki, sprawdzane jest, czy dany tekst zmieści się w danym obszarze, jeśli nie, to czcionka jest zmniejszana, dopóki tekst nie będzie dokładnie dopasowany. Użyta jest tu ta sama metoda sprawdzania, która używana była w metodzie `sizeHint` klasy delegat. Do drukowania jedno i dwustronnych karteczek służy metoda `printSide`, do drukowania listy zaś, metoda `printList`. Metody te odpowiadają za właściwe rozmieszczenie fiszek na kartce i sprawdzenie, czy nie trzeba zacząć nowej strony. Drukowanie karteczek dwustronnych przebiega dwuetapowo. Po wydrukowaniu słówek użytkownik musi odwrócić kartki w drukarce i zatwierdzić to kliknięciem.

Kiedy prace nad projektem były już na ukończeniu zająłem się implementacją klasy odpowiedzialnej za rysowanie wykresów przy prezentowanych statystykach uczenia się. Trudność implementacji polegała na tym, że rysowane wykresy miały odpowiednio dostosowywać swój rozmiar w zależności od wielkości okna. Dodatkowo dolny i górny zakres wartości, które pojawiały się na rysowanych wykresach nie był stały. Za rysowanie wykresów odpowiada klasa `Statistic`.

Najważniejszą metodą klasy jest metoda `draw`, która zwraca obiekt klasy `QPicture` wyświetlany później w zakładce ze statystykami. Na podstawie listy wartości i wielkości obszaru do rysowania wyliczane są odległości między punktami na wykresie. Rysowane jest tło wykresu. Następnie dla każdego dwóch punktów rysowany jest czworokąt zawierający daną parę punktów i leżące poniżej punkty podstawy wykresu. Na koniec podpisywane są osi wykresu. Starłem się, żeby odległość pomiędzy etykietami nie była za mała i żeby teksty nie nakładały się na siebie.

Jak każdy z członków naszego zespołu miałem swój wkład przy wymyśleniu głównej idei programu, a także przy późniejszym kształtowaniu tej wizji w bardziej szczegółowy zamysł projektowy. Stworzyłem również kilka elementów interfejsu użytkownika, między innymi zakładkę z zarządzaniem słownictwem i okienko, które umożliwiało dodanie nowego słówka, co jednak, przy wykorzystaniu `qt4`, nie należało do zadań najtrudniejszych.

## Rozdział 4

# Podsumowanie

Wspólna praca nad zespołowym projektem programistycznym jakim była aplikacja QWords była niepowtarzalną okazją do nabycia szerokiego wachlarza doświadczeń i niesamowicie praktycznej wiedzy. Mieliśmy okazję brać udział w metodycznym tworzeniu oprogramowania od samego początku do końca w dosyć sporym zespole. Wymagało to oprócz stosowania oprogramowania usprawniającego efektywność tworzenia, podejmowania nierzadko trudnych decyzji dotyczących podziału pracy, jak i również umiejętności rozwiązywaniu częstych sporów dotyczących decyzji projektowych. Proces tworzenia przybliżył nam wiele technologii i postaraliśmy się wybrać takie, które są popularne i cały czas rozwijane, by inwestycja w wiedzę zwracała się też w przyszłości. Warto jednak zaznaczyć, że proces tworzenia to tylko część projektu, gdyż rozpoczęliśmy proces wdrażania, który miałby szansę się powieść, co pociągnęłoby za sobą prace programistyczne związane z przystosowywaniem aplikacji do warunków biznesowych. Dodatkowo od nas wymagało to tworzenia dodatkowych dokumentacji i prowadzenia rozmów z osobami trzecimi, które najczęściej nie były informatykami, co było sprawdzianem dla naszych umiejętności miękkich. Podsumowując, dotknęliśmy większości, jeśli nie każdego aspektu procesu deweloperskiego i wdrożeniowego tworzenia aplikacji, która innowacyjnie podchodzi do ważnej problematyki i pozwoliło to nam na nieoceniony rozwój jako przyszli (a może już obecni) programiści.



## Rozdział 5

# Zawartość dołączonej płyty

- W katalogu źródła/QWords znajdują się pliki źródłowe programu QWords.
- W katalogu źródła/dictgenerator znajdują się pliki źródłowe pomocniczego programu generującego specjalny index słownika. Index ten wykorzystywany jest w celach optymalizacyjnych przez QWords.
- Plik QWords.exe to instalator QWords pod systemy z rodziny Windows (XP, Vista).
- W pliku wprowadzenie.pdf opisano podstawowe funkcjonalności programu QWords.



# Bibliografia

[Qt 4] Dokumentacja Qt 4.4, <http://doc.trolltech.com/4.4/>

[Qt 4] Jasmin Blanchette, Mark Summerfield *C++ GUI Programming with Qt 4*, Second Edition

[Qt 4] Daniel Molkenin, *The Book of Qt 4: The Art of Building Qt Applications*, Open Source Press GmbH, Monachium, Niemcy, 2007

[Algorytm SM2] <http://www.supermemo.com/english/ol/sm2.htm>

[Mercurial] <http://mercurial.selenic.com/wiki/>, <http://mercurial.selenic.com/wiki/UnderstandingMercurial>

[Windows API] [http://msdn.microsoft.com/en-us/library/cc433218\(VS.85,loband\).aspx](http://msdn.microsoft.com/en-us/library/cc433218(VS.85,loband).aspx)

[SQLite] Dokumentacja SQLite, <http://sqlite.org/docs.html>

[Anki] <http://ichi2.net/anki/>

[Kydpdict] <http://members.elysium.pl/ytm/html/kydpdict.html>

[Babylon] <http://babylon.com>