

# **System deklaracji przez e-mail**

Wojciech Pratkowiecki

Kurs języka Lua 2016/17

## Opis projektu

Projekt ma na celu umożliwienie weryfikacji wiedzy studentów na zasadzie systemu deklaracji w sytuacjach, w których nie jest możliwe przeprowadzenie tradycyjnych zajęć z deklaracjami. Z pomocą przychodzi poczta elektroniczna. Studenci na zajęciach zapoznają się z różnymi zagadnieniami, do których układane następnie są pytania. Aby zwiększyć prawdopodobieństwo samodzielnej pracy każdy ze studentów otrzymuje określoną, losowo wybraną liczbę pytań z całej bazy, dzięki czemu z większą szansą każdy dostaje inny zestaw zadań. Niniejszy program na podstawie pliku z pytaniami, pliku .csv zawierającym informacje o studentach, oraz plików z szablonem wstępu wiadomości i pliku z podstawowymi informacjami o zadaniu generuje wiadomości e-mail zawierające wylosowane pytania, a następnie wysyła je na podane w pliku .csv adresy mailowe przy użyciu protokołu SMTP. Ponadto odpowiada on także za pobieranie odpowiedzi od studentów. Za pomocą protokołu IMAP wybiera ze skrzynki mailowej wiadomości z dopasowaną treścią pola tytułu i umieszcza je w osobnych, specjalnie przygotowanych podfolderach. Następnie z otrzymanych wiadomości losuje określoną liczbę maili, które zostają oznaczone gwiazdą. Odpowiadają one wylosowaniu przez prowadzącego deklaracji podczas zajęć. Maile oznaczone gwiazdką są przeznaczone do weryfikacji przez prowadzącego. Ponadto program aktualizuje plik .csv, dodając do niego punkty tym studentom, którzy wysłali odpowiedzi na zadanie.

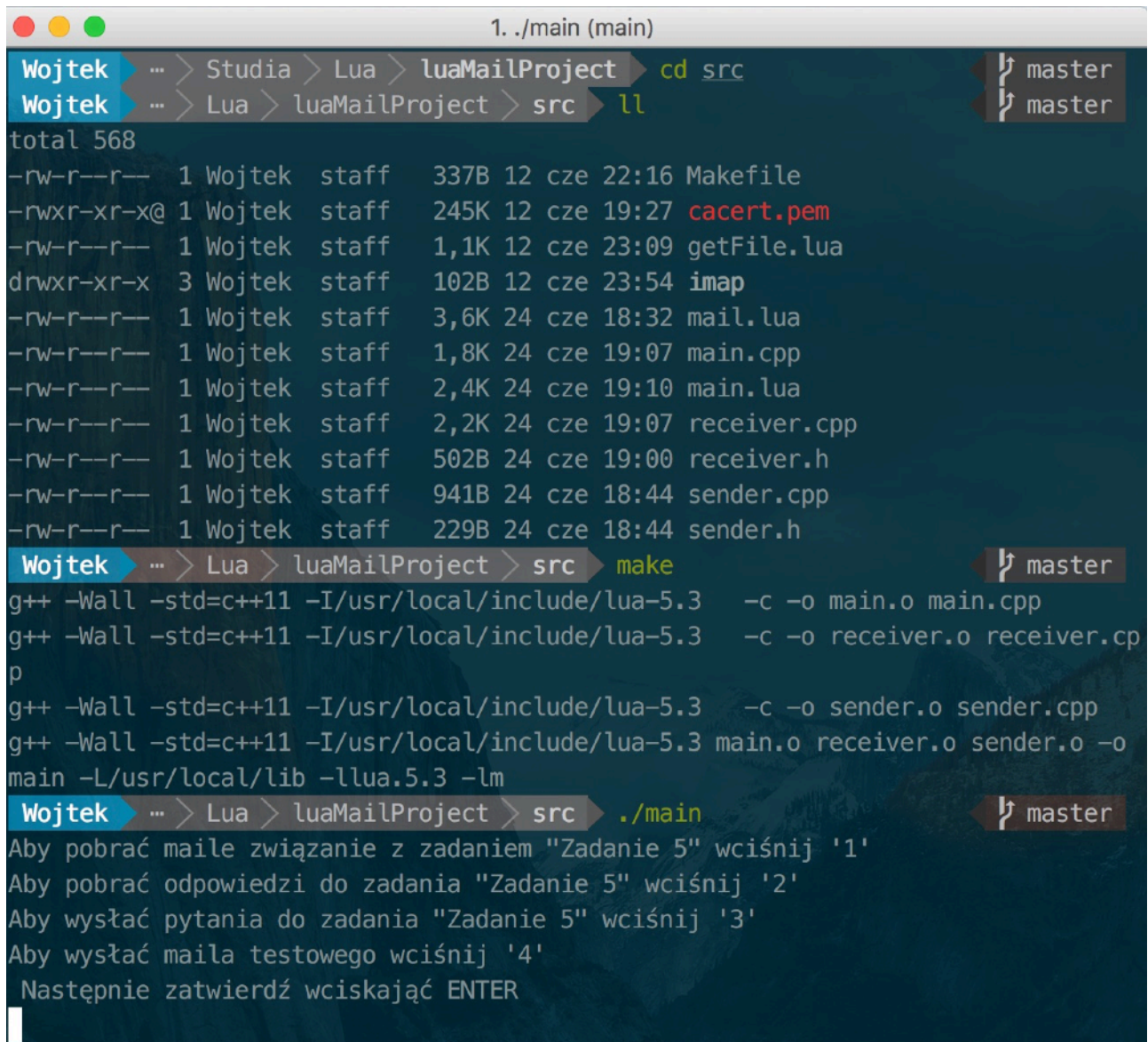
## Katalog główny

Katalog główny zawiera pliki przeznaczone dla użytkownika programu. Są one na tyle intuicyjne, aby osoba nie mająca pojęcia o programowaniu była w stanie skutecznie i efektywnie używać projektu. W przykładowym katalogu znajdują się pliki:

- task.txt - plik zawierający trzy informacje: tytuł zadania, data startu oraz deadline na oddawanie pytań (godzina 00:00:00 wpisanego dnia). Na podstawie tytułu zadania ustalana jest m.in. treść tytułu wiadomości, program główny używa go do prawidłowego wyświetlania informacji o stanie projektu, protokół IMAP wybiera odpowiedzi od studentów czy też stanowi on tytuł kolumny w pliku csv. Daty są natomiast potrzebne dla protokołu IMAP do odpowiedniego filtrowania wiadomości.
- quest.txt - lista pytań, z których losowane są zadania dla studentów
- students.csv - Arkusz którego trzy pierwsze kolumny to imię, nazwisko i adres e-mail studenta. Kolejne z kolumn są generowane przez program i zawierają liczbę punktów za zadanie. W niniejszym arkuszu są wpisane punkty za zadania zgodnie z odpowiedziami, jakie zostały przysłane na testowy adres e-mail w czasie testowania projektu.
- template.txt - Jest to szablon początku wiadomości umieszczany w każdym wysłanym mailu
- config.lua - Plik konfiguracyjny, w którym należy wskazać nazwy plików odpowiedzialnych za wspomniane powyżej funkcjonalności oraz wybrać liczbę pytań do wysłania dla każdego studenta, ilość wiadomości e-mail, które mają następnie zostać oznaczone gwiazdką, a także zdecydować, czy po przeniesieniu do odpowiednich podkatalogów wiadomości mają zostać usunięte z głównego katalogu.

## Działanie programu

Projekt jest napisany w Lua i C++. Skrypty są odpowiedzialne za obsługę skrzynki mailowej, w C++ zostały napisane funkcjonalności uruchamiające odpowiednie akcje ze skryptów oraz utrzymujące plik .csv. W katalogu *src* znajdują się pliki źródłowe oraz plik Makefile. Polecenie *make* kompiluje projekt, tworząc plik binarny *main*, który następnie uruchamiany jest bez podawania argumentów.



```
1. ./main (main)
Wojtek ... > Studia > Lua > luaMailProject > cd src master
Wojtek ... > Lua > luaMailProject > src ll master
total 568
-rw-r--r-- 1 Wojtek staff 337B 12 cze 22:16 Makefile
-rwxr-xr-x 1 Wojtek staff 245K 12 cze 19:27 cacert.pem
-rw-r--r-- 1 Wojtek staff 1,1K 12 cze 23:09 getFile.lua
drwxr-xr-x 3 Wojtek staff 102B 12 cze 23:54 imap
-rw-r--r-- 1 Wojtek staff 3,6K 24 cze 18:32 mail.lua
-rw-r--r-- 1 Wojtek staff 1,8K 24 cze 19:07 main.cpp
-rw-r--r-- 1 Wojtek staff 2,4K 24 cze 19:10 main.lua
-rw-r--r-- 1 Wojtek staff 2,2K 24 cze 19:07 receiver.cpp
-rw-r--r-- 1 Wojtek staff 502B 24 cze 19:00 receiver.h
-rw-r--r-- 1 Wojtek staff 941B 24 cze 18:44 sender.cpp
-rw-r--r-- 1 Wojtek staff 229B 24 cze 18:44 sender.h
Wojtek ... > Lua > luaMailProject > src make master
g++ -Wall -std=c++11 -I/usr/local/include/lua-5.3 -c -o main.o main.cpp
g++ -Wall -std=c++11 -I/usr/local/include/lua-5.3 -c -o receiver.o receiver.cpp
g++ -Wall -std=c++11 -I/usr/local/include/lua-5.3 -c -o sender.o sender.cpp
g++ -Wall -std=c++11 -I/usr/local/include/lua-5.3 main.o receiver.o sender.o -o
main -L/usr/local/lib -llua.5.3 -lm
Wojtek ... > Lua > luaMailProject > src ./main master
Aby pobrać maile związane z zadaniem "Zadanie 5" wciśnij '1'
Aby pobrać odpowiedzi do zadania "Zadanie 5" wciśnij '2'
Aby wysłać pytania do zadania "Zadanie 5" wciśnij '3'
Aby wysłać maila testowego wciśnij '4'
Następnie zatwierdź wciskając ENTER
```

Po uruchomieniu programu pokazuje nam się proste menu. Na podstawie pliku *task.txt* program wyznaczył tytuł zadania. Do wyboru mamy cztery opcje:

- Pobranie maili związanych z zadaniem - w skrzynce mailowej utworzony zostaje katalog „Zadanie 5 - Wszystkie”, do którego zostaną przeniesione wszystkie wiadomości zawierające w tytule „Zadanie 5” wysłane pomiędzy wskazanymi datami.
- Pobranie odpowiedzi - Do folderu „Zadanie 5 - Rozwiązania” zostaną przeniesione maile zatytułowane „Zadanie 5 - Rozwiązanie”. W treści wiadomości do studentów jest informacja, aby

tak właśnie tytułować odpowiedź. Następnie zostanie wybrana odpowiednia ilość wiadomości do oznaczenia, a także uaktualniony o nowe punkty będzie plik .csv

- Wysłanie pytań do zadania - Zostaną wygenerowane i wysłane e-maile do studentów
- Wysłanie maila testowego - Na adres e-mail, z którego się zalogujemy zostanie wysłana przykładowa wygenerowana wiadomość

Jak wynika z treści menu należy z klawiatury wprowadzić odpowiedni numer i zatwierdzić znakiem nowej linii. Po wykonaniu akcji zostanie wygenerowany odpowiedni komunikat. Przykładowe wywołanie dla wysyłania wiadomości i odbierania odpowiedzi:

```
Wojtek > ... > Lua > luaMailProject > src > ./main master
Aby pobrać maile związane z zadaniem "Zadanie 5" wciśnij '1'
Aby pobrać odpowiedzi do zadania "Zadanie 5" wciśnij '2'
Aby wysłać pytania do zadania "Zadanie 5" wciśnij '3'
Aby wysłać maila testowego wciśnij '4'
Następnie zatwierdź wciskając ENTER
3
Podaj adres e-mail:
luaprojectmail@gmail.com
Podaj hasło:
Wysłano maila do wpratkwiecki@gmail.com
Wysłano maila do wojtekp96@op.pl
Wysłano maila do luaprojectmail@gmail.com
Wysłano maila do luaprojectmail@gmail.com
Wysyłanie zakończone.
Wojtek > ... > Lua > luaMailProject > src > master
```

```
Wojtek > ... > Lua > luaMailProject > src > ./main master
Aby pobrać maile związane z zadaniem "Zadanie 5" wciśnij '1'
Aby pobrać odpowiedzi do zadania "Zadanie 5" wciśnij '2'
Aby wysłać pytania do zadania "Zadanie 5" wciśnij '3'
Aby wysłać maila testowego wciśnij '4'
Następnie zatwierdź wciskając ENTER
2
Podaj adres e-mail:
luaprojectmail@gmail.com
Podaj hasło:
Otrzymano odpowiedzi od:
luaprojectmail@gmail.com
wpratkwiecki@gmail.com
Zaktualizowano plik students.csv oraz skrzynkę mailową
Wojtek > ... > Lua > luaMailProject > src > master
```

## Pliki źródłowe

W katalogu src znajdują się pliki źródłowe projektu

- mail.lua - kod odpowiedzialny za obsługę protokołów SMTP i IMAP. Jego główne funkcjonalności to m.in. wysyłanie wiadomości e-mail, komunikacja z serwerem IMAP i obsługa skrzynki mailowej. Wykorzystywane są moduły *ssl* do szyfrowania połączenia z serwerem, *socket* do nawiązywania połączenia za pomocą gniazda, *imap* i *smtp* do obsługi skrzynki. Moduły *ssl*, *socket* i *smtp* można pobrać za pomocą luarocks, moduł *imap* znajduje się w podatalogu *imap/* zawierającym plik *imap.lua*. Jest to kod źródłowy znaleziony w sieci z wprowadzonymi kosmetycznymi zmianami do obsługi poleceń UID protokołu IMAP.
- getfile.lua - kod odpowiadający za obsługę plików tekstowych przez skrypt
- main.lua - wrapper dla funkcjonalności całości skryptu lua, z którym komunikuje się kod C++
- receiver.cpp / receiver.h - Kod odpowiadający za obsługę pobierania wiadomości - odpala odpowiednie funkcje lua, aktualizuje plik .csv
- sender.cpp / sender.h - Funkcje wysyłające wiadomości mailowe
- main.cpp - Funkcja główna generująca menu i odpalająca wybrane przez użytkownika funkcjonalności
- cacert.perm - zestaw kluczy, bez których serwer Gmail odmawia nawiązania połączenia. Klucze te zostały wygenerowane przez autora modułu *imap.lua*