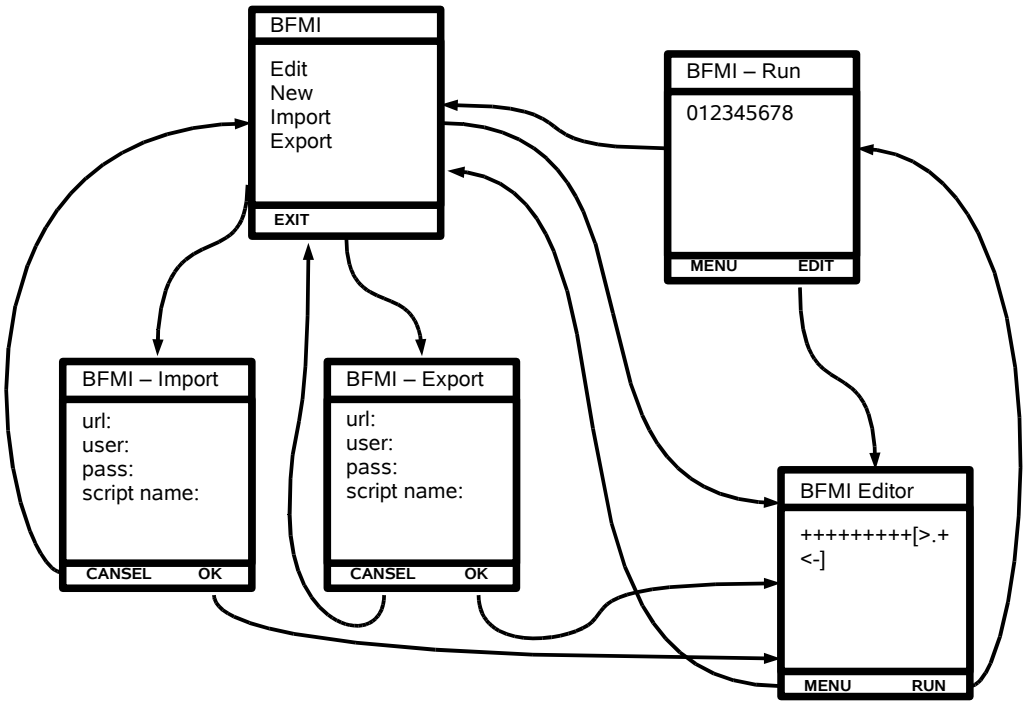
1. Wstęp

Program będzie przenośnym interpreterem języka BrainFuck, składającego się z 8 poleceń, będącego zupełnym w sensie Turinga.
Język posiada:

- dwa polecenia wejścia/wyjścia
- dwa polecenia dostępu do pamięci
- dwa polecenia operacji na danych
- dwa polecenia kontroli przebiegu programu.

Program jest przenośnym odpowiednikiem edytora tekstu z jednym z prostych interpreterów BrainFuck-a (http://en.wikipedia.org/wiki/Brainfuck) gdzie przechowywanie skryptów jest realizowane na zdalnym serwerze wchodzącym w skład pakietu.
Serwer ten będący samodzielną aplikacją Java potrafić będzie przechowywać skrypty zarówno lokalnie (na maszynie na której będzie uruchomiony) pod postacią plików, jak również w bazie danych poprzez interface JDBC.

2. GUI



Interfejs dzielimy na ekrany:

- Menu
  Tutaj posiadamy menu z opcjami. Dzięki niemu możemy stworzyć nowy pusty skrypt. Zaimportować z serwera istniejący. Lub wyeksportować stworzony. Albo wyjść z programu za pomocą “Exit”.
- Import
  To panel pozwalający na import skryptu z serwera. Brak podania nazwy powoduje wylistowanie skryptów znajdujących się na serwerze.
- Export
  To panel pozwalający na export skryptu na serwer.

- Editor
  To panel na którym edytujemy nasz skrypt.
- Run
  W tym panelu uruchamiamy skrypt wprowadzając dane wejściowe z klawiatury oraz obserwując wyniki wypisywane na wyjście.

Moduł serwera wypisuje na standardowe wyjście komunikaty o swoim działaniu.

3. Sposób użycia.

Aplikacja klienta

W midlet-cie w każdym z poszczególnych ekranów:

- Menu
  Wybieramy co chcemy zrobić. “New” gdy chcemy wyczyścić edytor żeby był pusty i zacząć pisać od nowa. “Import”/”Export” gdy chcemy odpowiednio wczytać lub zapisać skrypt. “Edit” gdy chcemy przejść do edytora. Przy czym po włączeniu programu jest on domyślnie pusty.
  Guzikiem “Exit” możemy opuścić program.
- Import i Export
  Tutaj wpisujemy adres serwera “url:” nazwę użytkownika i hasło (“user:” “pass:”). Adres “url” może być zarówno adresem “ip” jak i nazwą hosta do rozkodowania przez serwery “dns”. W pole “script name:” należy wpisać nazwę pod jaką ma być zapisany, lub odpowiednio jaki ma być wczytany program. Gdy przy imporcie zostawimy pole puste, wyświetli się lista dostępnych programów.
  Aby wykonać akcję używamy “OK”. Aby zaniechać “anuluj”.
- Editor
  Tutaj rozwijamy nasze aplikacje w BrainFuck-u. Za pomocą “Menu” przechodzimy do Menu. Za pomocą “Run” przechodzimy do panelu “Run” w którym zostaje nasz skrypt uruchomiony.
- Run
  Tutaj obserwujemy działanie naszego skryptu. Możemy wpisywać mu dane wejściowe i obserwować dane wyjściowe. Możemy od razu przejść do “Menu” (np. celem wyeksportowania programu) albo za pomocą “Edit” wrócić do panelu edycji.

Aplikacja serwera

posiada plik konfiguracyjny posiadający następujące parametry do skonfigurowania:

Dla przechowywania danych w plikach:
[ADDRESS]
[PORT]
FILES
[DIR]

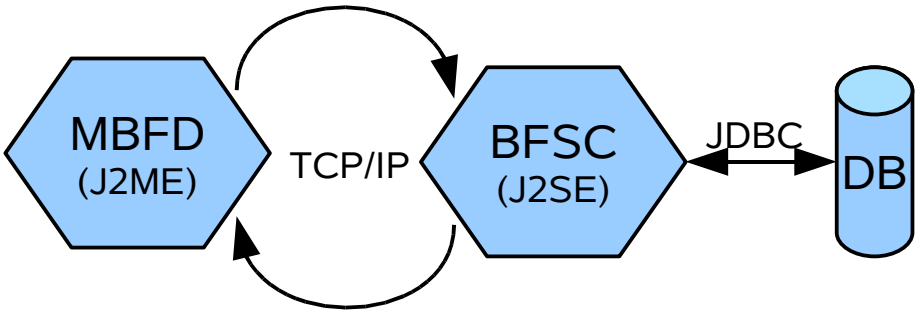
Dla przechowywania danych w bazie danych:
[ADDRESS]
[PORT]
DB
[JDBC_URL]
[USER]
[PASS]

Gdzie pod odpowiednie pola należy wstawić:
[ADDRESS] – adres pod który ma serwer przywiązać gniazdo.
[PORT] – port pod który ma przywiązać gniazdo nasłuchujące.
[DIR] – katalog w którym będą przechowywane pliki użytkowników oraz pliki konfiguracyjne z nimi związane
(podkatalogi to katalogi z programami użytkowników (np. [dir]/fred) natomiast pliki cfg z hasłami np. ([dir]/fred.cfg).
[JDBC_URL] – url do danej bazy danych w formacie “jdbc”
[USER] – użytkownik bazy jako który ma sie logować serwer
[PASS] – hasło dla tego użytkownika.

4. Opis budowy programu

Główne moduły:

MBFD - Mobile BrainFuck Developer - Midlet J2ME
BFSC - BrainFuck Script Container - Serwer Java.



Mobile BrainFuck Developer jest to midlet J2ME który pozwala użytkownikowi tworzyć i programy w języku BrainFuck. Komunikuje się z serwerem za pośrednictwem protokołu TCP/IP. BrainFuck Script Container - Serwer – jest to aplikacja mająca za zadanie przechowywać skrypty użytkowników. W zależności od konfiguracji, albo lokalnie w plikach, albo w bazie danych poprzez interfejs JDBC.

Serwer

przy przechowywaniu danych pod postacią plików posiada w katalogu bazowym dla każdego użytkownika podkatalog oraz plik konfiguracyjny z jego hasłem.
W przypadku przechowywania danych w bazie danych przechowuje dane w dwóch typach encji:

1. program : {*program_id*, user_id, nazwa_programu, kod_programu, data}
2. uzytkownik : { *user_id* , uname, upass }

Protokół

Protokół będzie działał według następującego algorytmu:
C – Client
S - Serwer

- Powitanie i przedstawienie
  - C: Połączenie. Wysłanie informacji o używanej wersji protokołu.
  - S: Odpowiedz akceptująca wersje protokołu lub odrzucająca i rozłączenie
  - C: Jeżeli odpowiedź odrzucająca to koniec procedury/połączenia.
  - C: Wysłanie loginu i hasła.
  - S: Odpowiedz akceptująca lub odrzucająca i rozłączenie
  - C: Jeżeli odpowiedź odrzucająca to koniec procedury/połączenia.
- Żądanie – Odpowiedź
  - C: Wysyła jedno z żądań, czego chce z parametrami.
    - Chce program (jego nazwa)
    - Chce listę programów
    - Chcę wysłać program (jego nazwa i kod)
  - S: W zależności od żądania klienta:
    - Odpowiedz akceptująca lub odrzucająca i rozłączenie.
    - Wykonanie polecenia, odesłanie informacji potwierdzającej wykonanie.
- Zakończenie
  - Obie strony Jeżeli nie dokonały jeszcze rozłączeń, rozłączają.

Główne klasy:

Będą główne klasy , które będą odpowiedzialne za:

- Po stronie klienta
  - brainfuckdeveloper – główna klasa midletu
    - brainfuckjavainterpreter – klasa z intrerpreterem języka brainfuck i stanem maszyny wirtualnej brainfuck-a
- Po stronie serwera
  - brainfuckcontainerserwer – główna klasa serwera
  - container – interface pojemnika na skrypty
    - -> dbcontainer – klasa przechowujaca w bazie danych
    - -> filecontainer – klasa przechowujaca w plikach

5. Zestawienie pakietów.

Pakiety:

JBFI (Java BrainFuck Interpreter)
W tym pakiecie będzie zawierać się funkcjonalność języka BrainFuck jak i jego obsługa (tzn. interpretacja).

BFIE (BrainFuck Scripts Import/Export)
W tym pakiecie będzie zawierać się funkcjonalność zapisu/odczytu programów zapisanych w BrainFuck.
To jest:

- komunikacja klienta z Serwerem
- komunikacja Serwera z klientem.
- zapis/odczyt z/do bazy danych poprzez JDBC
- zapis/odczyt z/do plików

Będzie dzielił się na dwa pod pakiety:

- client
- serwer

W poszczególnych będzie wydzielona funkcjonalność tylko dotycząca klienta bądź serwera.