GUI

Project III

14 V 2023

* Opis projektu

Pierwszy w historii SMS został wysłany 3 grudnia 1993 roku. Od tego czasu przeważająca większość urządzeń mobilnych ma możliwość wysyłania wiadomości tego typu. Rezultat tego projektu będzie aplikacją symulującą przesyłanie wiadomości od nadawcy (urządze-nia mobilenego) przez szereg warstw sieci (stacje pośrednie), po odbiorców (urządzenia mobilne).

Wykorzystując framework SWING, utwórz wielowątkową aplikację okienkową symu-lującą przesyłanie wiadomości SMS od nadawców do odbiorców. W warstwie wizualnej aplikacja będzie się składała z 3 interaktywnych paneli (obiektów dziedziczących po klasie JPanel), rozmieszczonych przez automatyczny rozkład BorderLayout.

Skrajny lewy panel, będzie dedykowany wyświetleniu urządzeń nadawczych. Panel ten będzie składał się, z komponentu JScrollPane umieszczonego w centralnej części i przycisku JButton „Add” położonego na dole panelu. Wewnątrz komponentu JScroll-Pane umieszczony zostanie JPanel, który przy pomocy pionowego rozkładu BoxLayout rozmieści wszystki komponenty wizualizujące „wirtualne urządzenia nadawcze” (VBD).

Wciśnięcie przycisku „Add” w panelu urządzeń nadawczych, wyświetli okienko dia-logowe pozwalające na wprowadzenie i zatwierdzenie krótkiej wiadomości tekstowej. Na podstawie wprowadzonej wiadomości powstanie obiekt realizujący wirtualne urządzenie nadawcze (VBD), którego instancje będą wizualizowane w panelu.

Wizualizacja obiektu VBD będzie oparta o komponent JPanel i będzie się składała z opisanych przez JLabel komponentów:

* JSlider – określającego częstotliwość wysyłania wiadomości;
* JButton – pozwalającego na zakończenie pracy VBD (co będzie skutkowało usunię-ciem urządzenia);
* JTextField – nieedytowalnego pola przedstawiającego numer urządzenia;
* JComboBox – pozwalający na wskazanie stanu WAITING lub ACTIVE, na pod-stawie których wątek realizujący ten VBD będzie w stanie wstrzymania lub pracy;

Skrajny prawy panel, będzie dedykowany wyświetlaniu urządzeń odbiorczych. Struk-tura tego panelu będzie analogiczna do przedstawionej struktury panelu urządzeń nadaw-czych. Przedstawiać będzie jednak wirtualne urządzenia odbiorcze (VRD), których obiekty będą tworzone w konsekwencji wciśnięcia przycisku „Add”.

Wizualizacja obiektu VRD, będzie składała się z opisanych przez JLabel komponen-tów:

* JButton – pozwalającego na zakończenie pracy VRD (co będzie skutkowało usu-nięciem urządzenia);
* JLabel – przedstawiającą informację o ilości otrzymanych wiadomości;
* JCheckBox – określającą czy informacja o ilości otrzymanych wiadomości ma być usuwana co 10 sekund w przypadku zaznaczenia, czy nie w przeciwnym przypadku.

1 z 3

GUI

Project III

14 V 2023

Panel środkowy pozwoli na wizualizację stacji bazowych (BTS) w skrajnych warstwach i stacji kontrolerów (BSC) warstw pośrednich. Panel ten będzie się składał z położonego w centralnej części komponentu JPanel, zawierającego poziomo zorientowane wizualizacje warstw BTS i BCS oraz położonych w dolnej części komponentu przycisków dodawania i usuwania warstw kontrolerów (BSC).

Każda warstwa BTS lub BSC jest realizowana przez JPanel, w którym komponenty wizualne przedstawiające pojedynczą stację (BTS lub BCS) są rozmieszczone pionowo.

Każda stacja BTS lub BSC jest reprezentowana jako komponent JPanel, składający się z komponentów:

* JLabel – przedstawiający kolejny, unikalny i nielosowy numer stacji;
* JLabel – ilość przetworzonych SMSów;
* JLabel – ilość oczekujących na wysłanie SMSów;
* Działanie aplikacji

Użytkownik aplikacji tworzy pewną ilość obiektów VBD, z których każdy jest oddzielnym wątkiem i natychmiast rozpoczyna transmitowanie wprowadzonej podczas tworzenia wia-domości. Ponieważ zgodnie ze standardem, każdy SMS ma zakodowany numer nadawcy i odbiorcy wraz z wiadomością, jako odbiorca wybierany jest losowy element VRD. Utwo-rzony SMS zostaje przekazany do stacji BTS, o najmniejszej liczbie oczekujących na wysłanie SMSów.

Aplikacja w stanie uruchomieniowym zawiera trzy warstwy. Pomiędzy warstwą wej-ściową a warstwą wyjściową musi się znajdować przynajmniej jedna warstwa pośrednia kontrolerów BSC. Dokłada ilość warstw pośrednich jest zależna od akcji użytkownika apli-kacji, który za pomocą klawiszy może dodać lub usunąć warstwę. Każda nowo utworzona warstwa komunikacyjna będzie tworzona z jednym BSC, natomiast usunięcie warstwy skutkuje zaprzestaniem przyjmowania wiadomości przez tą warstwę i natychmiastowym przekazaniem wiadomości z wszystkich BSC z pominięciem czasu przekazania.

Przekazywanie przykładowej wiadomości SMS będzie wyglądało następująco:

VBD →BTS →BSC →···→BTS →VRD

Należy przyjąć że:

* każdy BSC będzie przechowywał SMS przez losowy czas (od 5 do 15 sek) a następnie będzie go przekazywał do kolejnej warstwy;
* każdy BTS będzie przekazywał SMS do kolejnej warstwy lub VRD po upływie 3 sek;
* podczas przekazania SMSa do kolejnej warstwy zawsze wybierany jest ten BTS/BSC który zawiera najmniej SMSów;
* jeżeli w danej warstwie ilość SMS w każdym z BSC lub BTS jest większa od 5, automatycznie dodawany jest nowy BTS/BSC;

2 z 3

GUI

Project III

14 V 2023

Jeżeli okaże się że numer odbiorczy, pod który został wysłany SMS nie istnieje, program podniesie wyjątek a system obsłuży go wyświetlając komunikat w konsoli.

Przyjmij że:

* wiadomości przekazywane są przekazywane w postaci PDU zdefiniowanej w nor-mach Europejskiego Instytutu Norm Telekomunikacyjnych (ETSI) opisanych w do-kumentach GSM 03.38, GSM 03.40, GSM 03.41, natomiast załączony dokument opisujący kodowanie jest jedynie wypisem z tych norm;
* w momencie zamknięcia aplikacji, powstanie plik binarny przedstawiający informa-cje o każdym VBD, zawierający informacjo o ilości wysłanych wiadomości i zawie-rający wiadomość jaka była wysyłana.
* Wymagania formalne

Rozwiązanie w postaci dostarczonej aplikacji należy przedstawić do dnia 28.V.2023 na platformie Teams.

W projekcie wymagane jest wykorzystanie wszystkich zagadnień omówionych pod-czas wykładów i ćwiczeń, co obejmuje m.in. programowanie obiektowe, kolekcje, wyjątki, wątki, wejście/wyjście i wiele innych. Jednocześnie, nie można korzystać z żadnych ze-wnętrznych bibliotek, aby zagwarantować, że cały kod będzie opierał się tylko na stan-dardowej bibliotece Javy.

W celu zapewnienia prawidłowego i bezpiecznego wykonywania operacji współbież-nych, konieczne jest realizowanie ich z wykorzystaniem wątków (Thread lub Runnable). Używanie innych narzędzi, takich jak Timer czy Executor, nie będzie akceptowane.

W celu poprawnego zrealizowania projektu, należy dostarczyć dwie wersje programu:

* pojedynczy plik o nazwie SxxxxxProjekt03.java, który zawierać będzie cały kod źródłowy programu, wraz z klasami zewnętrznymi. Plik ten powinien kompilować się bez błędów;
* skompresowane archiwum o nazwie SxxxxxProjekt03.zip, zawierające wszystkie pliki projektu, wraz z plikami źródłowymi.

Należy pamiętać, że plik SxxxxxProjekt03.java powinien być samowystarczalny i zawie-rać wszystkie niezbędne klasy zewnętrzne, a także kompilować się bez błędów. Archiwum z kolei powinno zawierać kompletny projekt, włącznie z plikami źródłowymi.

Uwaga: niespełnienie wymagań formalnych będzie skutkowało pomnożeniem uzyska-nej oceny przez współczynnik 0.3.

3 z 3