

Nazwa Przedmiotu: Programowanie w Języku JAVA	Temat Projektu: Przeglądarka internetowa	Grupa wykonawcza: Bartłomiej Marzec Sebastian Wójcik
Rok Studiów: 2	Grupa Dziekańska: 2ID14A	

Ogólny opis projektu wraz z informacjami o technologiach, framework'ach i bibliotekach użytych w projekcie:

Tematem naszego projektu było zaprojektowanie przeglądarki internetowej w języku Java. Do wykonania projekt posługiwaliśmy się wersją Javy SDK 11 oraz przy wykorzystaniu narzędzia Maven w wersji 4.0.0. Jeżeli chodzi o frameworki, wykorzystaliśmy z mechanizmów Junit w wersji 5 do napisania testów jednostkowych. Do napisania przeglądarki skorzystaliśmy z bibliotek Java Swing, JavaFX oraz z biblioteki Abstract Window Toolkit, znanej szerzej jako AWT. Dzięki użyciu bibliotek Java Swing oraz AWT mieliśmy większą dowolność w operowaniu na akcjach zdarzeń przeglądarki oraz budowy jej okna, natomiast dzięki JavaFX rozszerzył te opcje o dodatkowe funkcjonalności.

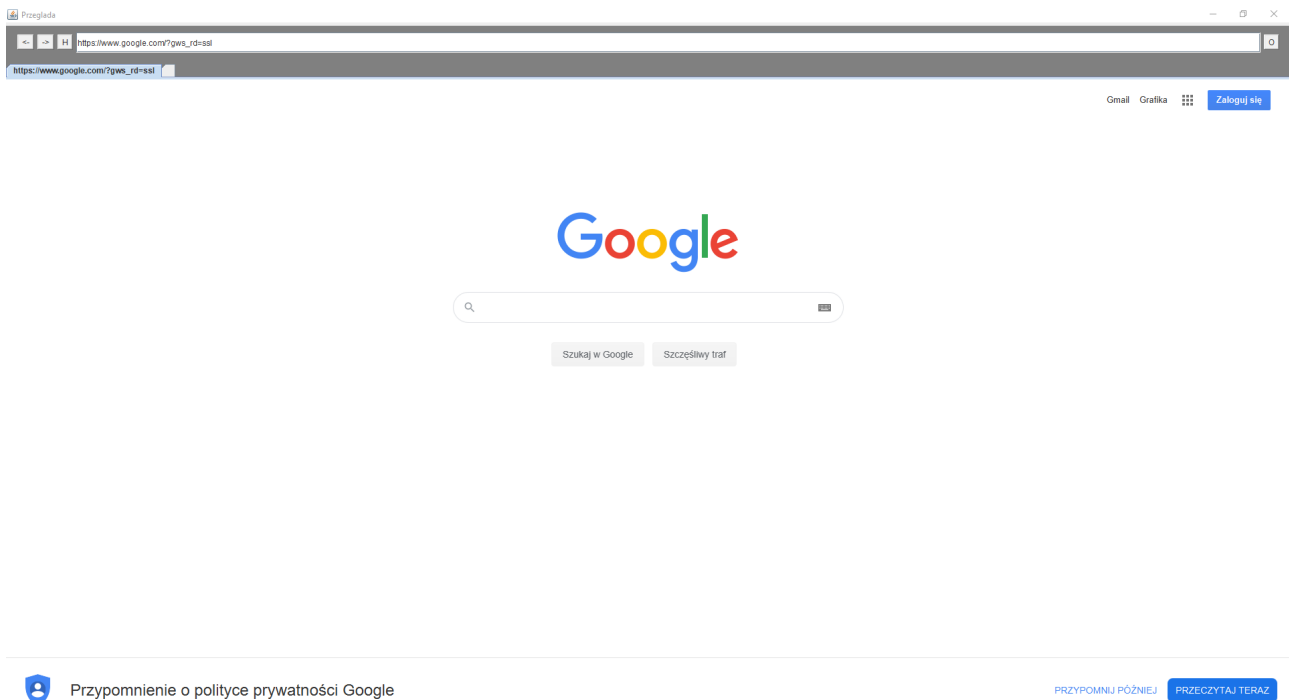
Informacje na temat funkcjonalności projektu:

Przeglądarka zawiera podstawowe funkcjonalności dla typowej przeglądarki internetowej. Po otwarciu przeglądarki możemy zaobserwować pasek adresu url u góry okna, w którym możemy wpisywać dowolny adres, przeglądarka rozróżnia różne ich wersje. Natomiast jeżeli żadna z wersji adresu nie zostanie odnaleziona w warunkach, fraza zostanie wyszukana w google. Możemy również poprzez naciśnięcie przycisku z boku kart zakładek dodać nową zakładkę, gdzie mamy możliwość przełączania się między nimi. Wraz z przełączaniem się zmienia się strona oraz adres paska adresu automatycznie się aktualizuje. Adresy zakładek internetowych również aktualizują swój adres url na podstawie stron na których się obecnie znajdujemy. Wszystkie te funkcjonalności natomiast wymagają pełnego załadowania strony, dlatego wymaga się odczekania na jej całkowite załadowanie, gdyż nie stosowanie się do tego może skutkować błędem wyświetlanego adresu lub ładowania strony. Przeglądarka posiada również cztery funkcjonalne przyciski, kolejno przycisk cofania stron do tyłu, który pozwala na cofnięcie się na poprzednią stronę. Adekwatnie do poprzednie przycisku jest również przycisk pozwalający na przesuwanie stron do przodu, jeżeli te istnieją. Jest również przycisk który pozwala na odświeżenie strony internetowej oraz przycisk wyświetlający historię w konsoli środowiska programistycznego w którym się znajdujemy.

Informacje na temat sposobu uruchamiania oraz obsługi projektu:

Program można uruchomić poprzez środowisko programistyczne, należy uruchomić program w typowy sposób dla danego środowiska. Po uruchomieniu programu pokazuje nam się okno przeglądarki wraz z pierwszą jego kartą oraz stroną główną google. Po wpisaniu adresu u góry w pasku adresowym, strona po chwili zostanie załadowana, a następnie adres zostanie zaaktualizowany wraz z podpisem na karcie zakładek. Obok pierwszej z kart zakładek występuje mały przycisk pozwalający na dodanie nowej karty, po naciśnięciu wyświetli nam się ponownie strona początkowa czyli google.com, a adres zaaktualizuje się adekwatnie do strony internetowej. Jeżeli będziemy chcieli się cofnąć na stronę do tyłu, należy nacisnąć przycisk strzałki w tył lub adekwatnie przycisk strzałki do przodu przeniesie nas na kolejną stronę, jeżeli takowa występowała.

Możemy również w razie potrzeby nacisnąć przycisk odświeżania strony po prawej stronie paska zadań, lub wyświetlić historię przeglądarki w konsoli środowiska programistycznego po naciśnięciu przycisku H.



informacje na temat stworzonych klas, metod, funkcji (bez kodu źródłowego)
z opisem ich podstawowej funkcjonalności (przyjmowanymi parametrami,
wartościami zwracanymi) oraz ich przeznaczeniem

W programie znajduje się dokładnie 5 klas, są to kolejno Main, Okno, PasekAdresu, Historia oraz Panel. Idąc po kolei z funkcjonalnością mamy klasę Main, jest to główne ciało programu wywołujące w metodzie main program poprzez utworzenie obiektu Okna dla klasy okna.

Następnie mamy klasę Okno, w której wykonywane są główne czynności przeglądarki. W klasie tej występuje wiele pól, są to **frame**, czyli zmienna odpowiadająca za wyświetlane dla programu okno, **Panel** czyli spod, panel1 i panel2 czyli zmienne wykonane czysto do procesu pozycjonowania elementów przeglądarki, tak by ta utrzymywała elementy takie jak pasek adresu czy przyciski w odpowiednich miejscach oraz o odpowiednich rozmiarach. Następnie występuje **pasekAdresu** klasy PasekAdresu, czyli zmienna odnosząca się do występującego u góry ekranu paska adresu wraz z podstawowymi funkcjonalnościami, które zostaną opisane w późniejszej fazie sprawozdania.

Kolejnym z elementów pól klasy Okno jest **jfxPanel**, czyli panel okna zawierający w sobie wszystkie zakładki kart przeglądarki. Podobną funkcję posiada **tabbedPane**, natomiast on uwzględnia ich podział na zakładki i zajmuje się ich podstawowymi funkcjonalnościami.

Pola klasy **button**, są to przyciski funkcyjne wykonujące zróżnicowane akcje dla przeglądarki.

Historia, czyli klasa przechowująca dane oraz podstawowe funkcjonalności związane z historią przeglądarki. Występuje tu również tablica obiektów klasy **WebView**, która ma za zadanie kontrolować na której z zakładek obecnie znajduje się użytkownik i wyświetlać mu odpowiednią stronę w zależności od tego. Ostatnimi z pól są zmienne typu **Int** o nazwach **aktualnyWebView** oraz **cofnij**, które kolejno są indeksem obecnie wyświetlanej karty przeglądarki oraz cofnij czyli liczba wymagana przy kontroli zmiany stron internetowych przy pomocy przycisków cofania i przechodzenia do przodu.

Klasa ta posiada jedynie **konstruktor domyślny**, w momencie jego wywołania pobierana jest rozdzielczość okna ekranu użytkownika, a okno przeglądarki automatycznie ustawia się na maksymalną szerokość i wysokość ekranu. Tworzy się tu również okno wyświetlane przez program oraz inicjowane są jego podstawowe parametry, takie jak rozmiar, pozycja czy widoczność.

Metody Klasy Okno:

void actionPerformed(ActionEvent event) – Jest to funkcja przyjmująca jako parametr kolejkę zdarzeń programu. Nic nie zwraca. Jeżeli zauważy, że zdarzenie zachodzące w przeglądarce pochodzi z paska Adresu oraz że jest użytkownik jest na nim skupiony (pasek jest naciśnięty przez użytkownika) to ten zapisuje dane z paska do odpowiednich pól oraz następuje wyszukiwanie witryny.

void zainicjujLayout(int width, int height) – Jest to funkcja przyjmująca jako parametr szerokość i wysokość okna, Nic nie zwraca. Funkcja ta zawiera wiele funkcjonalności, są to między innymi:

- Odpowiednie pozycjonowanie elementów przeglądarki
- Tworzenie oraz ładowanie strony startowej przeglądarki
- Umożliwienie funkcjonalności dodawania kart
- Umożliwienie funkcjonalności działania wszystkich przycisków

Elementy takie jak przyciski czy dodawanie kart działają na zasadzie Listener'ów czyli sprawdzają bez przerwy czy w kolejce eventów nie zachodzi żadne ze zjawisk, które mogłyby je uaktywnić. W momencie wystąpienia takiej sytuacji program wykonuje przypisaną mu czynność. Dokładniejsze opisanie tego procesu znajduje się w punktach powyższych sprawozdania lub w dokumentacji Javadoc.

void wyszukaj(String url) – Jest to funkcja przyjmująca jako parametr ciąg znaków przekazany przez pasek adresu. Nic nie zwraca. Funkcja ma za zadanie sprawdzać czy przekazany adres z paska adresu odpowiada, którejkolwiek z napisanych przez nas kombinacji adresów czyli:

- sprawdza czy zawiera . , jeżeli tak to adres będzie przekazany do metody załaduj
 - sprawdza czy zawiera https://
 - sprawdza czy zawiera http://
 - jeżeli poprzednie warunki się nie spełniły przekazuje adres url tak jak jest
- jeżeli żaden z powyższych warunków się nie sprawdził to przekazuje frazę do metody wyszukajWGoogle

void załaduj(String url, boolean zapis) – Jest to funkcja przyjmująca jako parametry ciąg znaków oraz zmienną typu logicznego. Nic nie zwraca. Ładuje przekazany ciąg znaków w odpowiedniej zakładce przeglądarki oraz jeżeli wartość logiczna parametru zapisu jest prawdą to zapisuje adres strony do historii przeglądarki.

void wyszukajWGoogle(String fraza, boolean zapis) – Jest to funkcja przyjmująca jako parametry ciąg znaków oraz zmienną typu logicznego. Nic nie zwraca. Ładuje przekazany ciąg znaków jako efekt wyszukiwania hasła w wyszukiwarce google oraz jeżeli wartość logiczna parametru zapisu jest prawdą to zapisuje adres strony do historii przeglądarki.

void aktualizujURL() – Jest to funkcja nie przyjmująca żadnych parametrów oraz która nic nie zwraca. Funkcja na bazie obecnie ładowanego adresu url w przeglądarce sprawdza status ładowania strony internetowej, a następnie jeżeli ta jest w pełni załadowana to aktualizuje adres url oraz podpis zakładki, zapisuje jego adres oraz dodaje go do historii.

void nowaKarta(int width, int height) – Jest to funkcja przyjmująca jako parametr szerokość i wysokość okna, Nic nie zwraca. Funkcja odpowiedzialna jest za utworzenie nowej karty w momencie wywołania, następnie nadania jej numeru indeksu pierwszego wolnego miejsca karty oraz ostatecznej nadania jej wszystkich bazowych ustawień karty.

`void zmianaKarty(ChangeEvent event)` – Jest to funkcja przyjmująca jako parametr kolejną zdarzeń programu. Nic nie zwraca. Ma za zadanie aktualizować pasek adresu u góry przeglądarki w momencie gdy w kolejce zdarzeń funkcja napotka zdarzenie związane z przełączeniem się inną zakładką.

`int szukajOstatniej()` – Jest to funkcja, która nie przyjmuje żadnych parametrów natomiast zwraca liczbę całkowitą. Funkcja ma zadanie wyszukać w istniejących zakładkach pierwszy wolny indeks na którym będzie można załadować nową kartę.

Kolejną z klas jest klasa **Panel**, posiada ona jedynie konstruktor do którego wywołania są potrzebne kolejno parametry, szerokość panelu, wysokość panelu, kolor panelu oraz zmienna typu logicznego potwierdzająca czy panel ma być pozycjonowany w sposób relatywny względem pozostałych okien czy w sposób statyczny w centrum okna przeglądarki. Konstruktor definiuje podstawowy kolor, rozmiar oraz pozycję panelu utworzonego przez użytkownika.

Następną z klas jest Klasa **Historii**, w klasie tej znajdują się wszystkie niezbędne funkcje do obsługi historii przeglądarki. Posiada ona dwa pola, są to kolejno tab czyli tablica przechowująca adresy stron historii przeglądarki oraz index, który iteruje po adresach przechowywanych w tablicy historii. Występuje tu również konstruktor domyślny, który tworzy nową tablicę oraz zeruje indeks.

Metody klasy Historia:

`void dodaj(String url)` – Jest to funkcja przyjmująca jako parametr ciąg znaków przekazany do dodania do tablicy historii oraz zwiększa indeks. Nic nie zwraca.

`void wyczysc()` – Jest to funkcja nie przyjmująca żadnych parametrów oraz która nic nie zwraca. Funkcja odpowiedzialna za usuwanie całej zawartości z tablicy historii.

`void wypisz()` – Jest to funkcja nie przyjmująca żadnych parametrów oraz która nic nie zwraca. Funkcja odpowiedzialna za wypisanie całej zawartości tablicy historii w konsoli programu.

Ostatnią z klas jest klasa **PaskaAdresu** opisująca podstawowe funkcjonalności oraz mechanizmy działania paska adresów URL przeglądarki. Klasa ta zawiera dwa pola, a konkretniej zmienna typu string przechowującą adresy stron internetowych wpisywane w pasek oraz pole typu logicznego, które sprawdza czy użytkownik posiada obecnie naciśnięty pasek do wpisywania. W konstruktorze klasy przekazywany jest rozmiar, czyli szerokość paska. Konstruktor ten ustawia podstawowe parametry dla paska adresu.

Metody klasy PasekAdresu:

`void zapiszPole(int strona)` – Jest to funkcja przyjmująca jako parametr numer zakładki w z której ma zostać pobrany adres. Nic nie zwraca. Funkcja odpowiedzialna za pobieranie adresu strony internetowej do tablicy, oraz przypisanie jej indeksu odpowiadającego karcie przeglądarki.

`void zapiszPole(String tekst, int strona)` – Jest to funkcja przyjmująca dwa parametry, jako tekst który ma zostać zapisany do pamięci paska adresu oraz numer karty z której ma zostać pobrany. Nic nie zwraca. Funkcja odpowiedzialna za pobieranie tekstu do tablicy, oraz przypisanie jej indeksu odpowiadającego karcie przeglądarki.

`String wezPole(int strona)` – Jest to funkcja pobierająca jako parametr numer karty przeglądarki strony. Zwraca ciąg znaków pobrany z pamięci paska adresu o podanym numerze strony.

`void focusGained(FocusEvent event)` – Jest to funkcja pobierająca jako parametr kolejkę zdarzeń. Nic nie zwraca. Sprawdza czy użytkownik „skupił” się na obiekcie paska adresu, a ściślej mówiąc czy nacisnął na niego i się nie przełączył.

`void focusLost(FocusEvent event)` – Jest to funkcja pobierająca jako parametr kolejkę zdarzeń. Nic nie zwraca. Sprawdza czy użytkownik nie stracił „skupienia” z obiektu, a ściślej mówiąc czy nie nacisnął na inny obiekt okna.

informacje na temat ilości pracy włożonej przez poszczególnych członków zespołu w tworzenie projektu.

Wkład uczestników projektu:

Praca była podzielona możliwie po równo, ze względów ilościowych oraz funkcjonalności programu.

Bartłomiej Marzec:

- Podstawowy szablon programu
- Implementacja bibliotek wraz z budową okna
 - Pozycjonowanie paneli okna
 - Przyciski oraz ich funkcjonalności
- Budowa oraz funkcjonalność kart przeglądarki
- Budowa oraz funkcjonalność Historii przeglądarki

Sebastian Wójcik:

- Podstawowy szablon programu
- Implementacja bibliotek wraz z budową okna
- Budowa oraz funkcjonalność Paska Adresów przeglądarki
 - Przyciski oraz ich funkcjonalności
- Funkcjonalność oraz mechanizmy działania kart przeglądarki
 - Dokumentacja oraz JavaDoc