

1. Wprowadzenie do środowiska eclipse.

Platforma ta najczęściej definiowana jest, jako środowisko służące programowaniu w języku Java, lecz jej dynamiczny rozwój przyczynił się do rozszerzenia jej możliwości na inne języki programowania czy projektowania. Dzięki temu platforma wzbogacana o odpowiednie wtyczki (pluginy) staje się bardzo popularnym narzędziem mającym zastosowanie w dziedzinach informatycznych nie zależnie od tego czy pracujemy w Javie, C++ czy PHP. Środowisko zintegrowane z platformą daje nam możliwości pracy i tworzenia aplikacji w bardzo przyjaznych i sprzyjających użytkownikowi warunkach

1

2. Początki pracy czyli pobranie i instalacja.

Aby pobrać najnowszą wersję Eclipse'a wchodzimy do działu downloads na stronie <http://www.eclipse.org/downloads/>. Do naszych celów wystarczy normalna wersja Eclipse'a czyli *Eclipse Classic*. Pobieramy ją i rozpakowujemy archiwum w wybraną przez nas lokalizację.

The screenshot shows the Eclipse Downloads page for Windows. The page lists several IDE packages, with 'Eclipse Classic 3.7.1' highlighted by a red circle. The packages listed are:

Package Name	Size	Downloaded Times	Details	Other Downloads	Windows 32 Bit	Windows 64 Bit
Eclipse IDE for Java Developers	128 MB	3,235,314	Details		Windows 32 Bit	Windows 64 Bit
Eclipse IDE for Java EE Developers	212 MB	2,279,861	Details		Windows 32 Bit	Windows 64 Bit
Eclipse Classic 3.7.1	174 MB	1,241,158	Details	Other Downloads	Windows 32 Bit	Windows 64 Bit
Eclipse IDE for C/C++ Developers (includes Incubating components)	107 MB	693,681	Details		Windows 32 Bit	Windows 64 Bit
Eclipse IDE for JavaScript Web Developers	110 MB	260,457	Details		Windows 32 Bit	Windows 64 Bit
Eclipse Modeling Tools	271 MB	120,930	Details		Windows 32 Bit	Windows 64 Bit
Eclipse IDE for Java and Report Developers	242 MB	109,547	Details		Windows 32 Bit	Windows 64 Bit

Konieczne będzie również pobranie środowiska uruchomieniowego Javy (JRE) i środowiska programistycznego Javy (JDK). Wyżej wymienione środowiska, w najnowszych wersjach, można znaleźć pod tym <http://www.oracle.com/technetwork/java/javase/downloads/index.html> adresem. Po pobraniu wystarczy to środowisko zainstalować

Java SE Downloads

[Latest Release](#) | [Next Release \(Early Access\)](#) | [Embedded Use](#) | [Previous Releases](#)

Download Java Platform (JDK) 7u2 | **Download** JavaFX 2.0.2 | **Download** JDK 7u2 + NetBeans Bundle | **Download** JDK 7u1 + Java EE Bundle

Here are the Java SE downloads in detail:

Java Platform, Standard Edition	
Java SE 7u2 This release includes performance improvements, bug fixes, support for Solaris 11, and Firefox 5 and up. Learn more ▶	JDK Download JDK 7 Docs ▪ Installation Instructions ▪ ReadMe ▪ ReleaseNotes
"What Java Do I Need?" You must have a copy of the JRE (Java Runtime Environment) on your system to run Java applications and applets. To develop Java applications and applets, you need the JDK (Java Development Kit), which includes the JRE.	JRE Download JRE 7 Docs ▪ Installation Instructions ▪ ReadMe ▪ ReleaseNotes

3. Uruchamianie.

Eclipse uruchamiamy za pomocą pliku wykonawczego eclipse.exe znajdującego się w katalogu, do którego wypakowaliśmy platformę. Zanim program wystartuje w pełni zapyta nas gdzie powinien umieścić folder przestrzeni roboczej (workspace), jest to o tyle istotny katalog, że znajdują się w nim nasze projekty a w nich utworzone przez nas pliki. Można utworzyć dowolną liczbę takich folderów, jednakże rozgraniczenie projektów

będzie łatwiejsze do nawigacji niż tworzenie wielu obszarów roboczych (chyba, że pracujemy na kilku płaszczyznach wtedy warto pomyśleć o rozgraniczeniu na kategorie np. PHP, Java itp.).

Kiedy dokonamy wyboru położenia naszego obszaru roboczego program przejdzie do ekranu powitalnego, na którym odnaleźć można kilka zakładek (w kolejności od lewej do prawej):

- **Overview** – znajduje się tu pomoc odpowiadająca na pytania związane ze środowiskiem jak i te dotyczące bezpośrednio korzystania z niego, pomoc jest obszerna i rozbudowana, co umożliwi dobre zapoznanie się z działaniem samego środowiska przed przystąpieniem do pracy.
- **What’s new** - dosłownie: co nowego, zakładka ta zawiera informacje o nowinkach zawartych w zaktualizowanej wersji względem starszej, zmiany w interfejsie, nowe łatki, czy też nowe funkcje, jest to bardzo istotne w przypadku rozwiązań open source, ponieważ umożliwia obserwację na bieżąco rozwoju narzędzia.
- **Samples** – Zawiera przykładowe projekty wykonane z pomocą platformy (muszą one być dograne osobno)
- **Tutorials** – zakładką ta zawiera samouczki, które pomagają nam rozpocząć prace z poszczególnymi elementami platformy, znajdują się tu również odnośniki do wcześniejszej dokumentacji, którą odnajdowaliśmy pod zakładką Overview.
- **Workbench** – guzik ten umożliwia przejście do obszaru roboczego, w którym będzie odbywała się nasza praca.



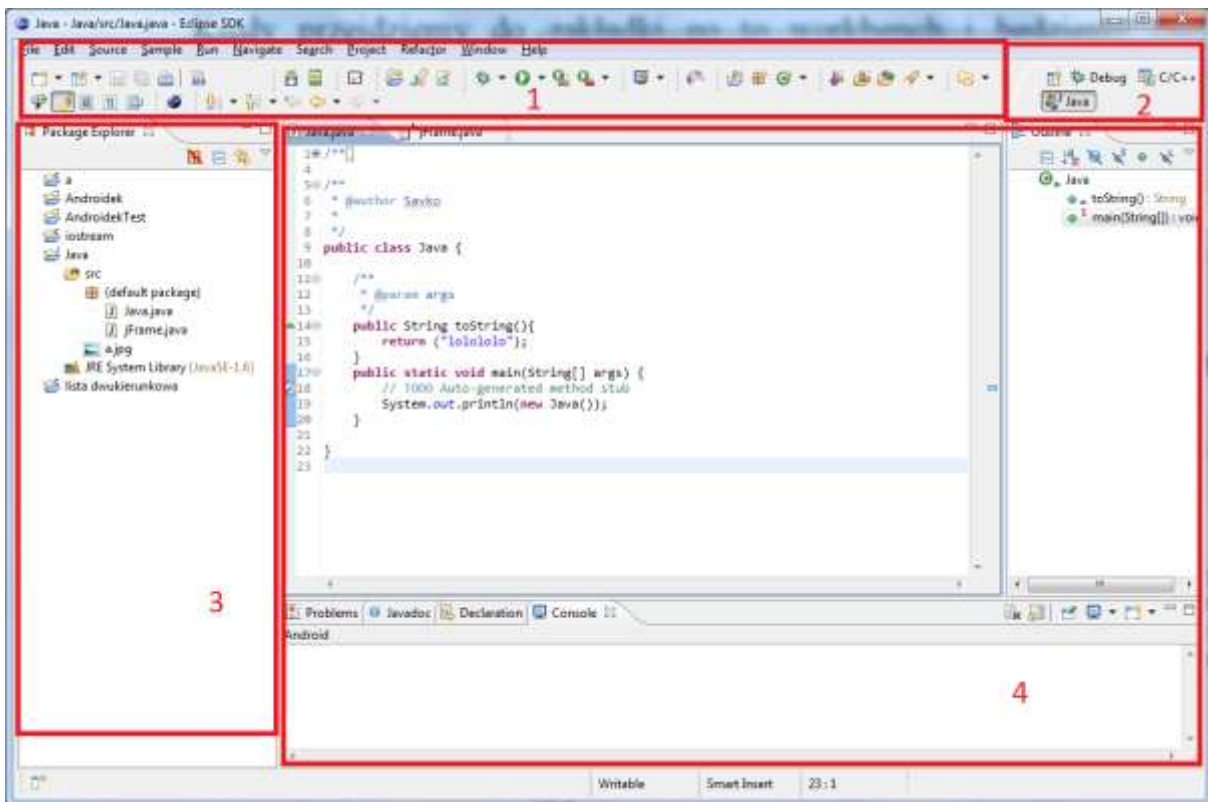
4. Interfejs użytkownika.

4

Interfejsem nazywamy wszystko to co pozwala nam komunikować się z programem, dlatego istotnym jest aby interfejs zawsze był przejrzysty i czytelny, decyduje o tym wiele czynników ułatwiających i utrudniających prace w danym środowisku.

a. Workbench.

Inaczej obszar roboczy, będzie to pierwsze co ujrzymy po przejściu z ekranu powitalnego dalej. Znajdują się tu elementy graficzne które mają ułatwić nam prace z danym projektem, przykładowo znajduje się tu rozwijane menu czy też poszczególne okna programu, dobrze ilustruje to poniżej zamieszczony zrzut ekranu z opisem:



1. Są to dwa istotne dla pracy programu paski jeden z nich dotyczy opcji głównych jak wyszukiwanie, pomoc, zapisywanie projektu czy tworzenie nowego projektu. Drugi pasek to pasek szybkich narzędzi, do których mamy dostęp od ręki (taki układ można zauważyć w większości programów użytkowych).
2. Jest to przycisk odpowiedzialny za perspektywy. Perspektywy są jedną z bardziej charakterystycznych opcji eclipse, służą one do dokonywania zmian w naszym warsztacie, umożliwiają one zarządzaniem oknami tak aby w danym momencie na naszym ekranie znajdowały się tylko te okna i opcje których będziemy potrzebowali

pracując z daną treścią np. po przełączeniu się na perspektywę Javy wiemy że nie odnajdziemy tam rzeczy bezpośrednio z tym językiem niezwiązanych (chyba że sami będziemy chcieli żeby było inaczej), poprzez dowolną zmianę zawartości perspektywy mamy ogromne możliwości kustomizacji naszego miejsca pracy. Tworzenie nowej perspektywy jest równie proste, co zarządzanie nimi, wystarczy ustawić potrzebne nam okna wybrać z paska narzędzi głównych menu **Window** a następnie opcje **Save Perspective As**.

3. Jest to okno widoku projektu tu znajduje się przedstawiona za pomocą drzewa składowa projektu, okno to umożliwia również dodawanie nowych projektów, w znacznym stopniu okno to ułatwia nam poruszanie się po kodzie i odnajdywanie nam potrzebnych w danym momencie danych czy elementów.
4. Jest to okno edytora, w którym wyświetlać będzie się wybrany przez nas fragment kodu, nad którym będziemy pracować, zmianę tego okna jak już nadmieniałem wyżej umożliwia zmiana i odnalezienie interesujących nas treści w okienku widoku i kliknięciu ich dwa razy.

Okno edytora (wyżej oznaczone numerem 4) umożliwia przeglądanie zawartości pliku który otworzyliśmy, jedna perspektywa umożliwia przeglądanie naraz jednego okna edytowalnego, w celu przejrzania innych plików wykorzystujemy zakładki znajdujące się u góry okna które umożliwiają swobodne przemieszanie się wewnątrz zawartości naszego projektu.

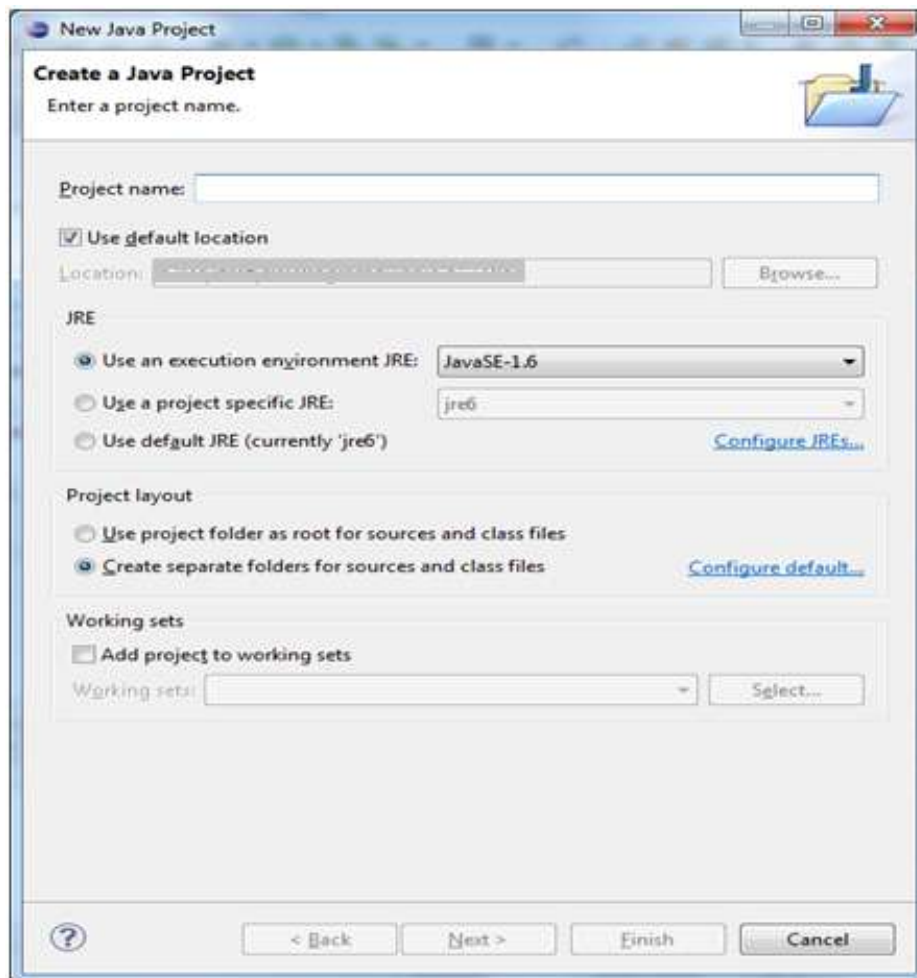


Eclipse dysponuje również ciekawą opcją dotyczącą otwierania pliku w edytorze. Jeżeli program nie będzie posiadał wtyczki do edycji danego formatu plików np. jpg otworzy go w domyślnym narzędziu do tego służącym, w przypadku jpg będzie to przeglądarka obrazów. Gdy przychodzi jednak do pracy z plikami właściwymi danemu językowi programowania program zachowuje się jak profesjonalne narzędzie, podświetla nam składnię, poprawia nas i pomaga uzupełnić błędy lub braki, umożliwia nam również prostą nawigację po kodzie. Oznacza to że platforma to pomimo ogromnej ilości rozszerzeń i możliwości nie zaniedbuje żadnej z dziedzin których dotyczy.

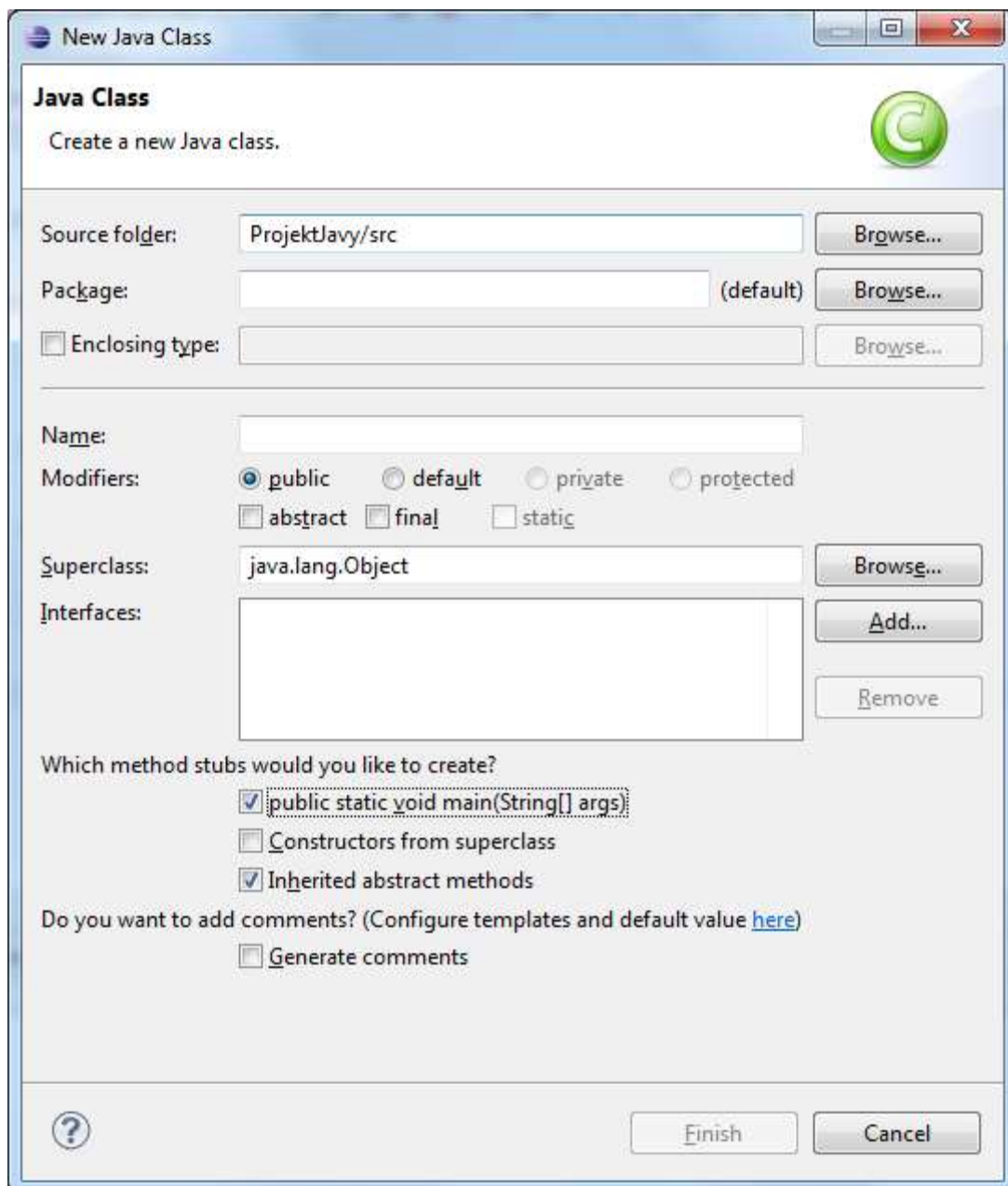
5. Pierwszy program.

Z menu wybieramy nowy projekt Java (np. **File->New->Java Project**). Pojawi się okno:

6



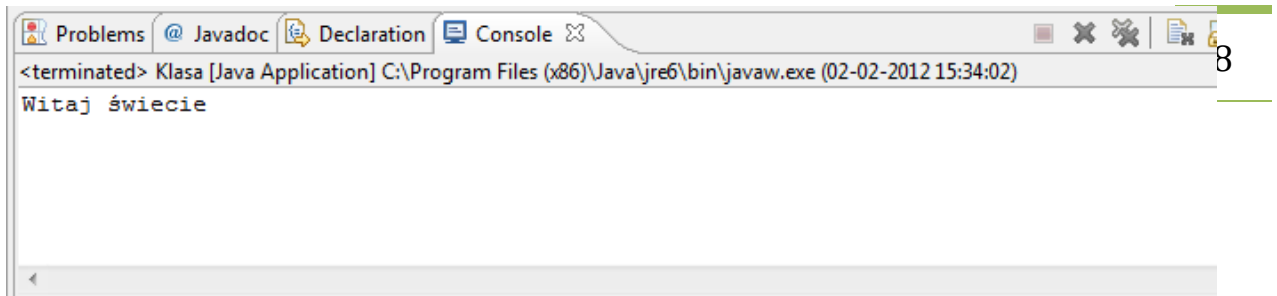
W oknie tym podajemy nazwę naszego projektu i klikamy na przycisk **finish**. Mamy już nowy projekt, teraz wystarczy kliknąć PPM w oknie widoku projektu i stworzymy nową klasę.



Podajemy nazwę klasy, zaznaczamy pole „*public static void main(String[] args)*” (jeśli chcemy aby nasza klasa była wyposażona w statyczną metodę *main*) i klikamy na przycisk *finisz* w celu utworzenia klasy. Mamy gotowy program.

```
package com.klasa;
public class Klasa {
    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Witaj świecie");
    }
}
```

Laboratorium 01: „Poznanie środowiska programowania w języku Java” [2h]



Środowisko wyposażone jest dodatkowo w narzędzie **Javadoc** – jest to generator dokumentacji stworzony przez firmę Sun Microsystems. Narzędzie to generuje dokumentację kodu źródłowego Javy na podstawie zamieszczonych w kodzie komentarzy Javadoc. Komentarz Javadoc oddzielony jest znacznikami `/**` i `*/`, które sprawiają, że ich zawartość (czyli to, co znajduje się między nimi), jest ignorowana przez kompilator. Pierwsza jego linia to opis metody lub klasy, która zadeklarowana jest poniżej. Dalej znajduje się pewna liczba opcjonalnych tagów, które z kolei opisują parametry metody (`@param`), wartość zwracaną (`@return`) itp.

```
package com.klasa;
import java.util.*;
/** Klasa to zwykła klasa testowa
 * @version 1.0
 * @author Anonim
 */
public class Klasa {

    /**statyczna metoda main
     * @param args przyjmuje argumenty podane z linii wywołania programu
     * metoda wypisuje na ekranie komunikat witaj świecie, oraz wyświetla
     * aktualną datę
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Witaj świecie");
        System.out.println(new Date());
    }
}
```

Witaj świecie

Thu Feb 02 17:19:32 CET 2012

Niektóre znaczniki dokumentacyjne:

9

I. Ogólne:

Znacznik	Opis
@author autor	Informacja o autorze (może być wielu).
@version wersja	Informacja o wersji.
@see klasa lub klasa#metoda lub napis	Generuje odwołanie do dokumentacji innej klasy, metody lub informację o odwołaniu.
{@link klasa#metoda etykieta}	Działa tak samo jak @see , z tym że może być stosowany wewnątrz wiersza. Zamiast słów "See also" umieszczany jest napis etykieta.
@deprecated	Oznacza składowe, które zostały zastąpione przez nowsze wersje i używanie ich nie jest zalecane.
{@docRoot}	Generuje względną ścieżkę do głównego katalogu zawierającego dokumentację.
{@inheritDoc}	Umieszcza w bieżącym komentarzu dokumentację najbliższej klasy bazowej aktualnie dokumentowanej klasy.

II. Dla metod:

Znacznik	Opis
@param parametr opis	Opisuje jeden parametr metody. Może ich być dowolnie dużo, jednak na ogół jest ich tyle ile parametrów metody, w takim samym porządku.
@return opis	Opisuje wartość zwracaną przez metodę.
@throws wyjątek opis (albo @exception wyjątek opis)	Opisuje wyjątek, który może być rzucony przez tę metodę.

Aby wygenerować dokumentację należy z menu wybrać **Project -> Generate Javadoc**. Wynikiem jest plik w formacie HTML, który możemy oglądać w dowolnej przeglądarce. Oprócz komentarzy w dokumentacji umieszczony jest również kod klas oraz jej metody i dane.

Wygenerowana dokumentacja:

10

The screenshot shows the JavaDoc documentation for a class named 'Klasa'. The interface includes a navigation bar at the top with 'Package', 'Class', 'Use Tree', 'Deprecated', 'Index', and 'Help'. Below this, there are tabs for 'Prev Class', 'Next Class', 'Frames', and 'No Frames'. The main content area displays the package 'com.klasa' and the class name 'Klasa'. It shows the inheritance hierarchy: 'java.lang.Object' and 'com.klasa.Klasa'. The source code for the class is displayed, showing a public class declaration, inheritance from 'java.lang.Object', and a comment: 'Klasa to zwykła klasa testowa'. Below the code, there are sections for 'Constructor Summary' and 'Method Summary'. The 'Constructor Summary' section shows a single constructor: 'Klasa()'. The 'Method Summary' section shows a table with columns for 'Modifier and Type' and 'Method and Description'. The table lists a static void method named 'main' with the signature 'main(java.lang.String[] args)' and a description 'statyczna metoda main'. Below the table, there is a section for 'Methods inherited from class java.lang.Object' listing methods like 'equals', 'getClass', 'hashCode', 'notify', 'notifyAll', 'toString', 'wait', and 'wait'.

Zadanie:

Zaimplementuj aplikację obsługującą działanie kalkulatora (dodawanie, odejmowanie, mnożenie, dzielenie, potęgowanie, pierwiastkowanie oraz modulo). Operacja wybierana jest na podstawie parametrów podanych z linii wywołania programu, np. jeśli wpiszemy + 4 8 to wykonane zostanie dodawanie 4 + 8, itp.

Do programu należy wygenerować dokumentację za pomocą Javadoc. (w kodzie mają być komentarze).