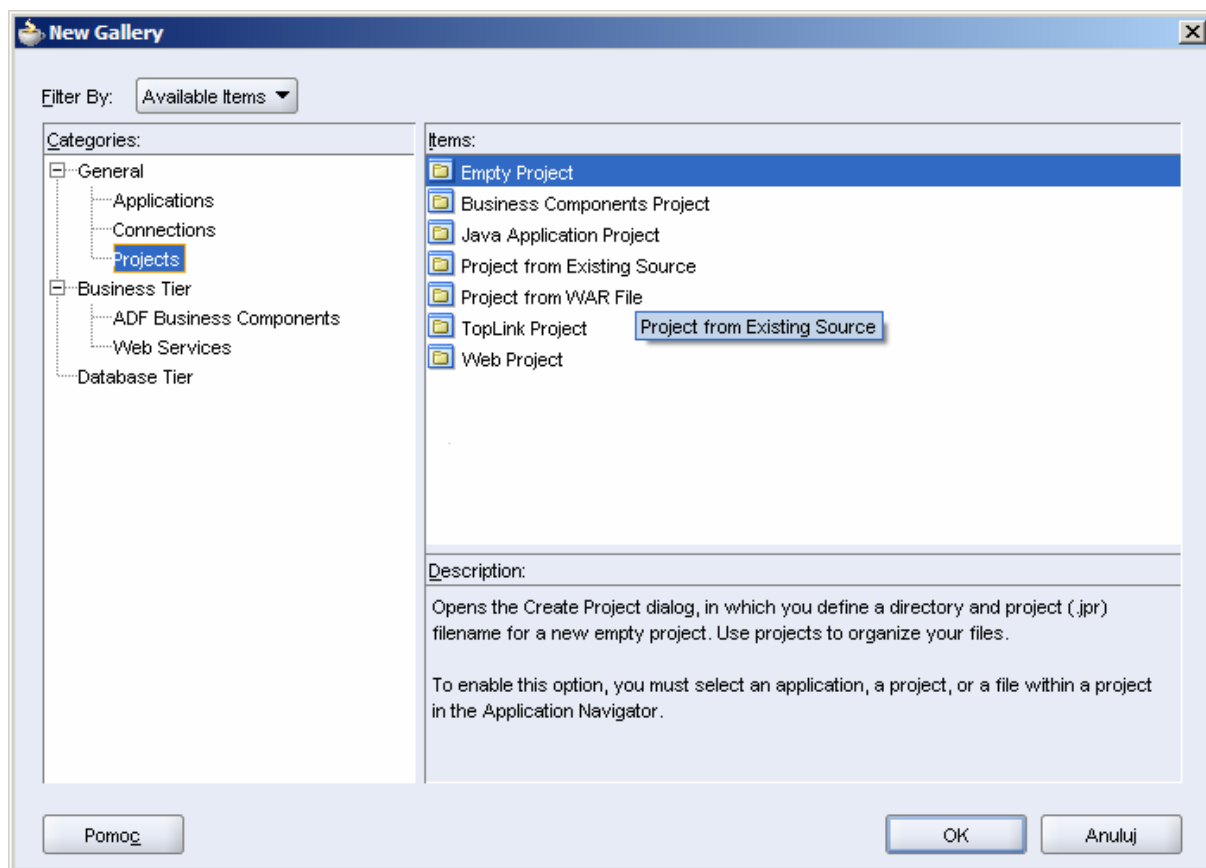


Zaawansowane aplikacje internetowe - laboratorium

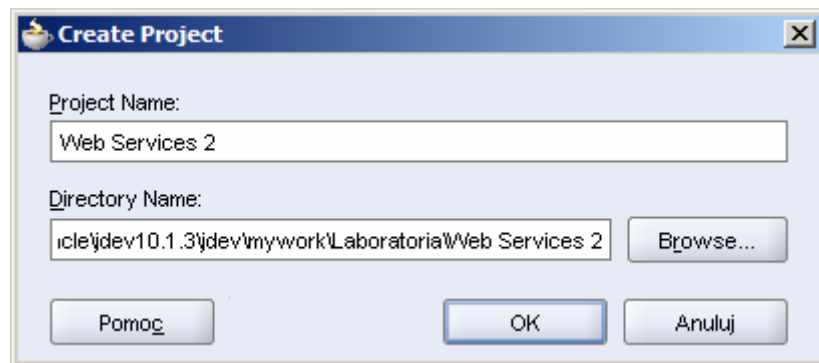
Web Services (część 2).

Celem ćwiczenia jest przygotowanie prostej aplikacji prezentującej technologię usług sieciowych (ang. Web Services). Przygotowana klasa języka Java zostanie opublikowana w postaci usługi sieciowej i udostępniona na zewnątrz przez osadzenie w serwerze aplikacji. Dodatkowo, przygotowany zostanie klient usługi testujący jej poprawność. Do wykonania ćwiczenia potrzebne jest zintegrowane środowisko Developer 10.1.3 (do pobrania z <http://otn.oracle.com>).

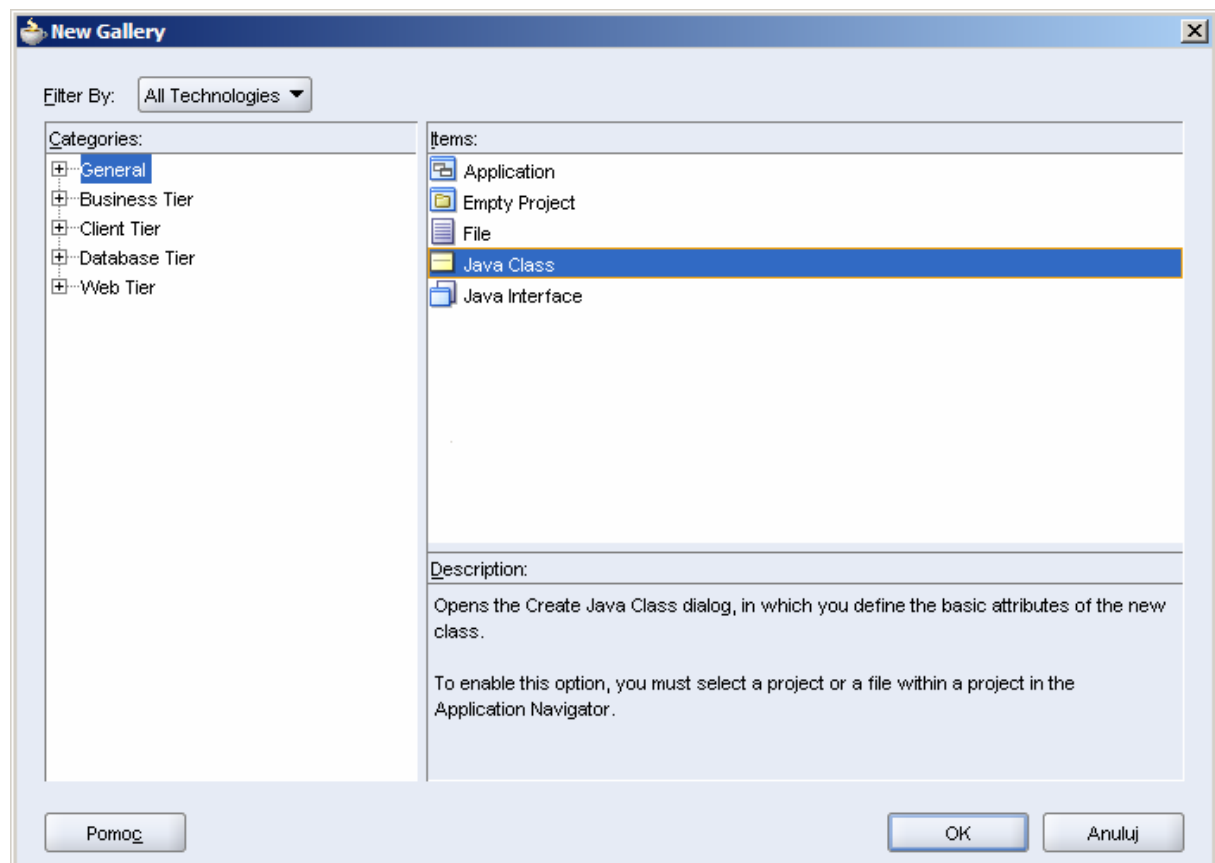
1. Uruchom narzędzie JDeveloper
2. Jeśli nie ukończyła(e)ś laboratorium Web Services (część 1) i nie posiadasz jeszcze aplikacji, to wykonaj kroki 2 i 3 z laboratorium Web Services (część 1).
3. Kliknij prawym przyciskiem myszy na ikonie aplikacji Laboratoria i z menu kontekstowego wybierz **New Project**. Rozwiń gałąź kategorii **General** i zaznacz kategorię **Projects**. W prawym oknie zaznacz element **Empty Project**. Kliknij przycisk **OK**.



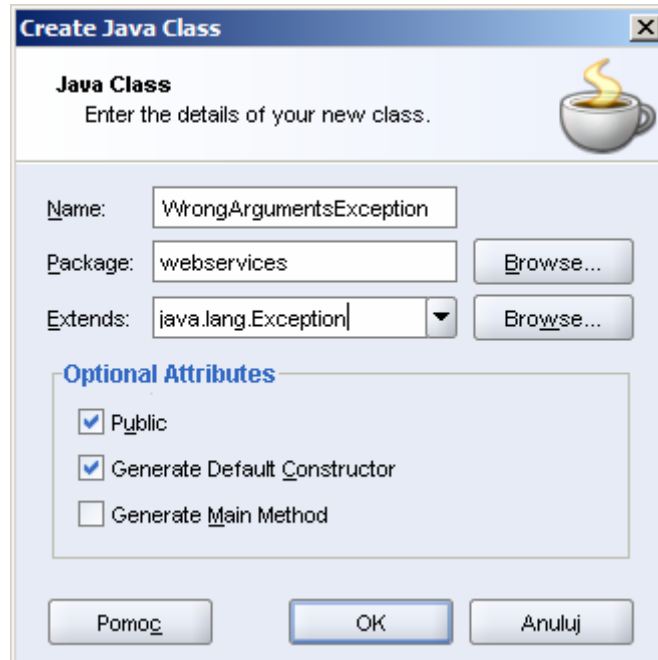
4. Wpisz nazwę projektu **Web Services 2** i kliknij przycisk **OK**.



5. Kliknij prawym przyciskiem myszy na ikonie projektu **Web Services 2**. Z menu kontekstowego wybierz opcję **New...** W oknie kategorii zaznacz gałąź **General**. Z listy dostępnych elementów w oknie po prawej stronie wybierz **Java Class**. Kliknij przycisk **OK**.



6. Pierwszym krokiem w tworzeniu usługi sieciowej będzie zdefiniowanie własnej klasy reprezentującej wyjątek polegający na niepoprawnym wywołaniu usługi, np. poprzez podanie jako parametrów liczb ujemnych albo liczb nie będących liczbami całkowitymi. Wpisz nazwę klasy: `WrongArgumentsException`, nazwę pakietu: `webservices`, nazwę nadklasy: `java.lang.Exception`. Kliknij przycisk **OK**.



Create Java Class

Java Class
Enter the details of your new class.

Name:

Package:

Extends:

Optional Attributes

Public

Generate Default Constructor

Generate Main Method

7. Plik `WrongArgumentsException.java` został automatycznie załadowany do edytora. Uzupełnij plik poniższym kodem.

```
package webservices;  
  
public class WrongArgumentsException extends Exception {  
  
    public WrongArgumentsException(String message) {  
        super(message);  
    }  
}
```

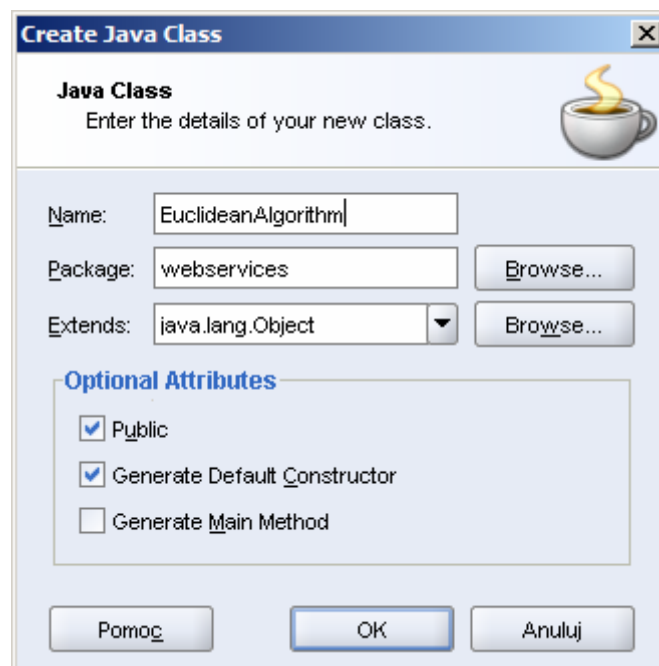
8. Programy stanowiące usługi sieciowe powinny być szczególnie dobrze dokumentowane ze względu na to, że ich głównymi użytkownikami są użytkownicy zdalni. Stąd, na projektant(k)ach usług sieciowych ciąży obowiązek starannego opisywania API. W przypadku języka Java najbardziej odpowiednim narzędziem jest Javadoc. Uzupełnij kod pliku `WrongArgumentsException.java` o komentarze Javadoc. Przykładowa wersja finalna pliku może wyglądać tak.

```
package webservices;

/**
 * The class <code>WrongArgumentsException</code> is a form
 * of <code>Throwable</code> and indicates the condition when input
 * parameters for the Web Service are incorrect.
 *
 * @author Mikolaj Morzy
 * @version 0.1
 */
public class WrongArgumentsException extends Exception {

    /**
     * Constructs a WrongArgumentsException with the specified detail
     * message.
     *
     * @param message the message describing this particular exception.
     */
    public WrongArgumentsException(String message) {
        super(message);
    }
}
```

9. Kliknij prawym przyciskiem myszy na ikonie projektu **Web Services 2** i z menu kontekstowego wybierz **New...**, zaznacz kategorię **General** i wybierz element **Java Class**. Kliknij przycisk **OK**. Wprowadź nazwę klasy: **EuclideanAlgorithm** oraz nazwę pakietu: **webservices**. Kliknij przycisk **OK**.



10. Do znajdowania największego wspólnego dzielnika dwóch liczb wykorzystamy algorytm Euklidesa (http://en.wikipedia.org/wiki/Euclids_algorithm). Wprowadź do pliku poniższy kod, nie zapomnij o udokumentowaniu poszczególnych elementów.

```
package webservices;

/**
 * The <code>EuclideanAlgorithm</code> class is an implementation
 * of the well-known algorithm attributed traditionally to Euclid.
 * The algorithm takes as input two positive integers and computes
 * the greatest common divisor of the two. The first appearance of
 * the algorithm dates back to 300 BC in Euclid's <i>Elements</i>.
 *
 * @author Mikolaj Morzy
 * @version 0.1
 */
public class EuclideanAlgorithm {

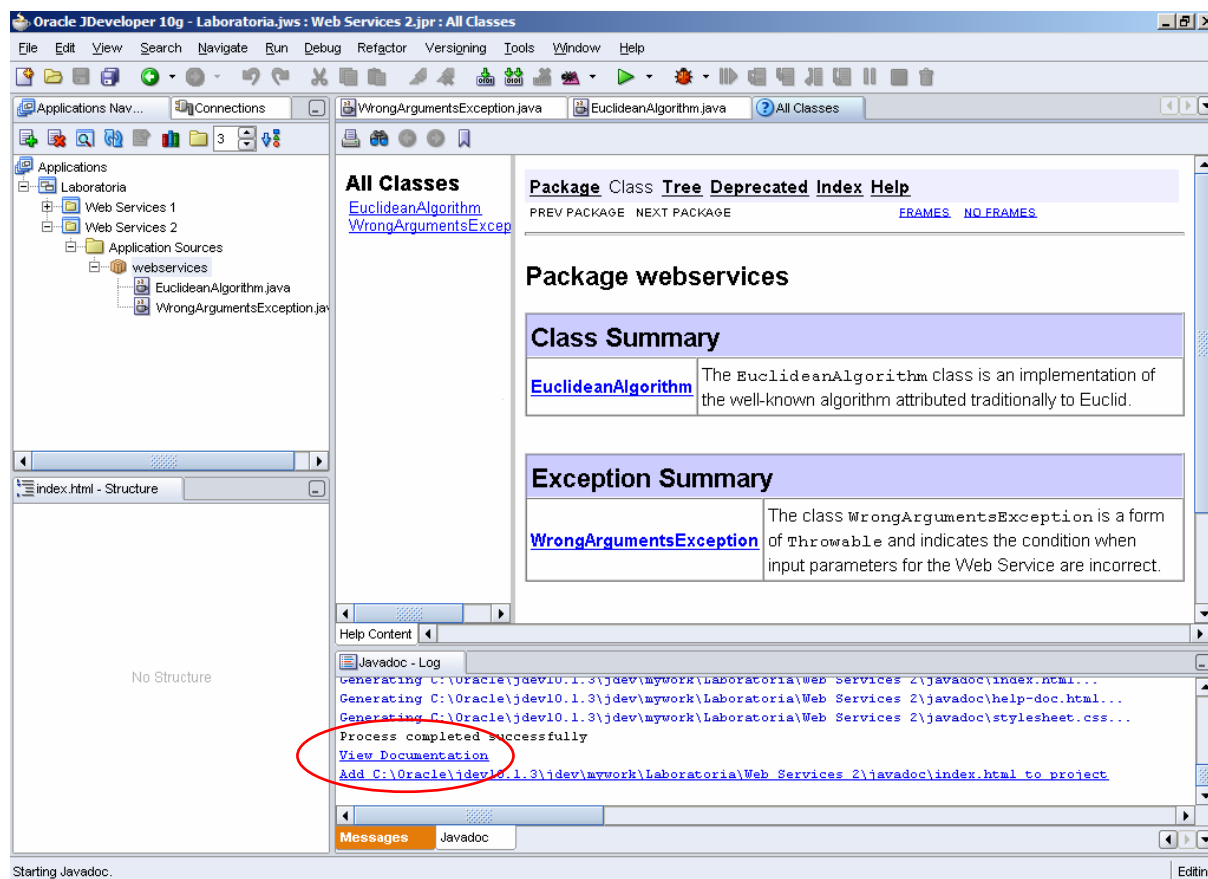
    /**
     * Computes the greatest common divisor of the two
     * input positive integers
     *
     * @param a The first number
     * @param b The second number
     * @return The greatest common divisor of the input parameters
     * @throws WrongArgumentsException If one of the parameters
     * is not positive, or when input parameters are not integers
     */
    public int GreatestCommonDivisor(String a, String b)
        throws WrongArgumentsException {

        int i, j;
        try {
            i = Integer.parseInt(a);
            j = Integer.parseInt(b);
        }
        catch (NumberFormatException e) {
            throw new WrongArgumentsException("Parameters must be integers");
        }

        if ((i < 0) || (j < 0))
            throw new WrongArgumentsException("Parameters must be positive");

        while (i != j)
            if (i > j)
                i -= j;
            else
                j -= i;
        return i;
    }
}
```

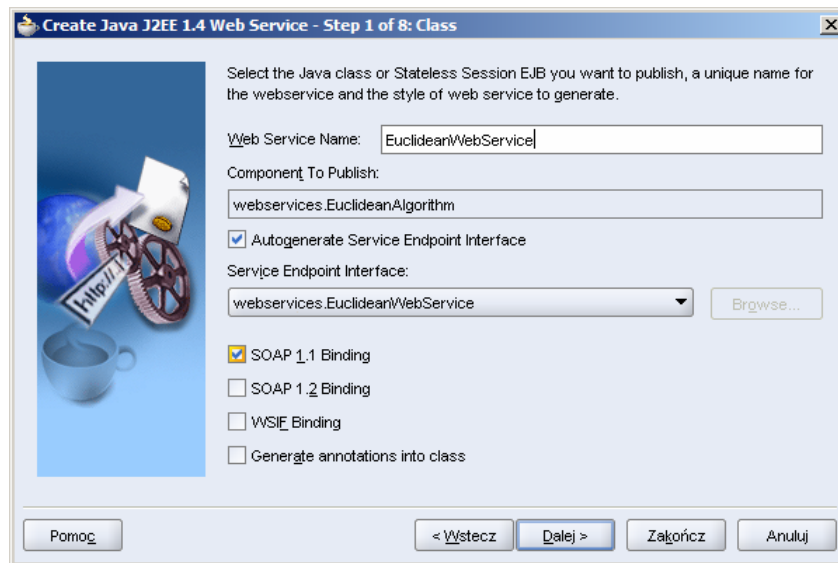
11. W nawigаторze obiektów rozwiń gałąź reprezentującą projekt Web Services 2, a następnie rozwiń gałąź Application Sources. Zaznacz węzeł reprezentujący pakiet webservices. Z menu głównego wybierz Run→Javadoc webservices. Po zakończeniu generowania dokumentacji w oknie dziennika (u dołu ekranu) kliknij na odnośnik [View Documentation](#). Ekran komputera powinien wyglądać teraz tak.



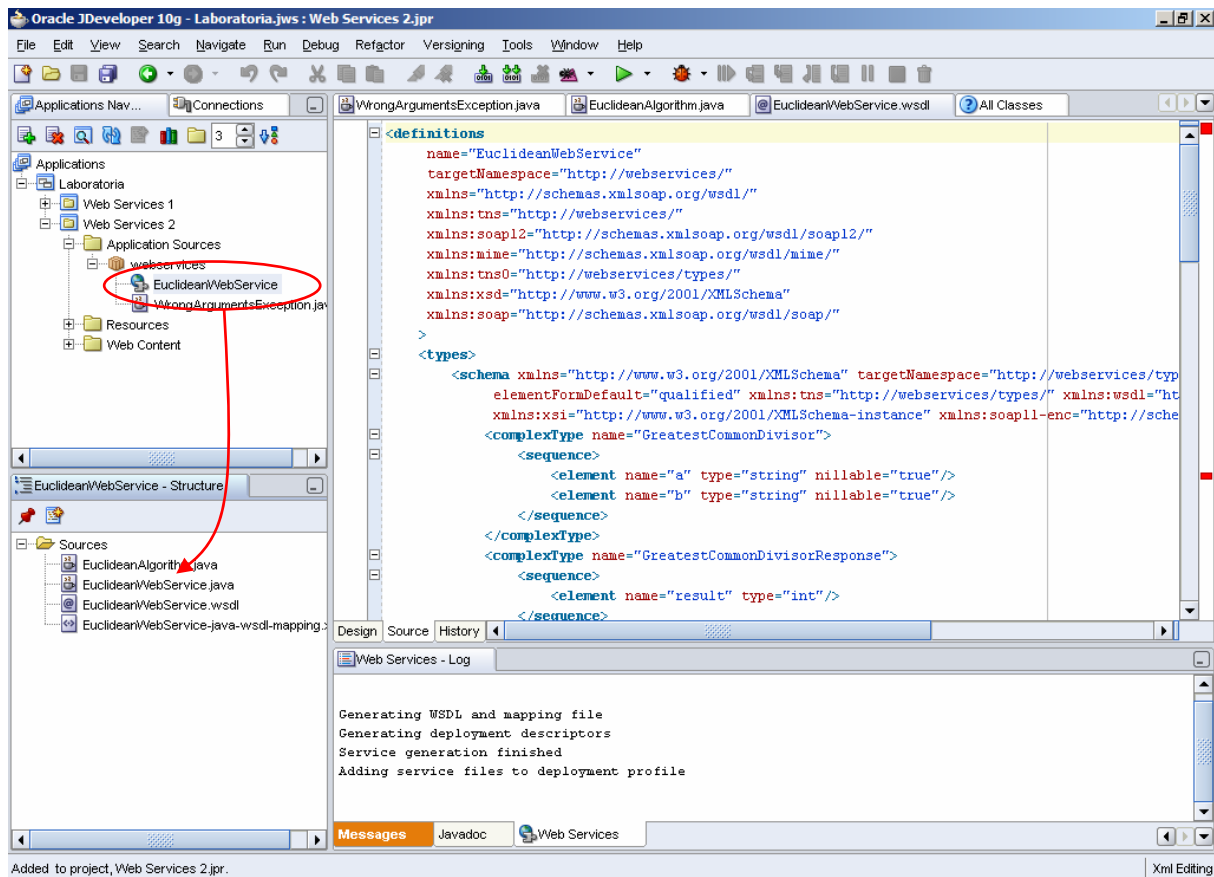
12. W kolejnym kroku udostępniemy stworzoną klasę EuclideanAlgorithm (wraz z pomocniczą klasą WrongArgumentsException) w postaci usługi sieciowej. Kliknij prawym przyciskiem myszy na ikonie pliku EuclideanAlgorithm.java i z menu kontekstowego wybierz opcję Create J2EE Web Service. Wybierz wersję J2EE 1.4 i kliknij przycisk OK.



13. Podaj nazwę usługi: `EuclideanWebService`. Upewnij się, że jest włączona opcja **Autogenerate Service Endpoint Interface**. Kliknij przycisk **Zakończ** (pozostałe parametry usługi sieciowej pozostają domyślne).



14. Zauważ, że w nawigаторze obiektów zmieniła się ikona reprezentująca plik `EuclideanAlgorithm.java`, a zamiast tego pojawiła się ikona reprezentująca całą usługę. W celu obejrzenia całej zawartości usługi zaznacz tę ikonę i z menu głównego wybierz **View**→**Structure**.

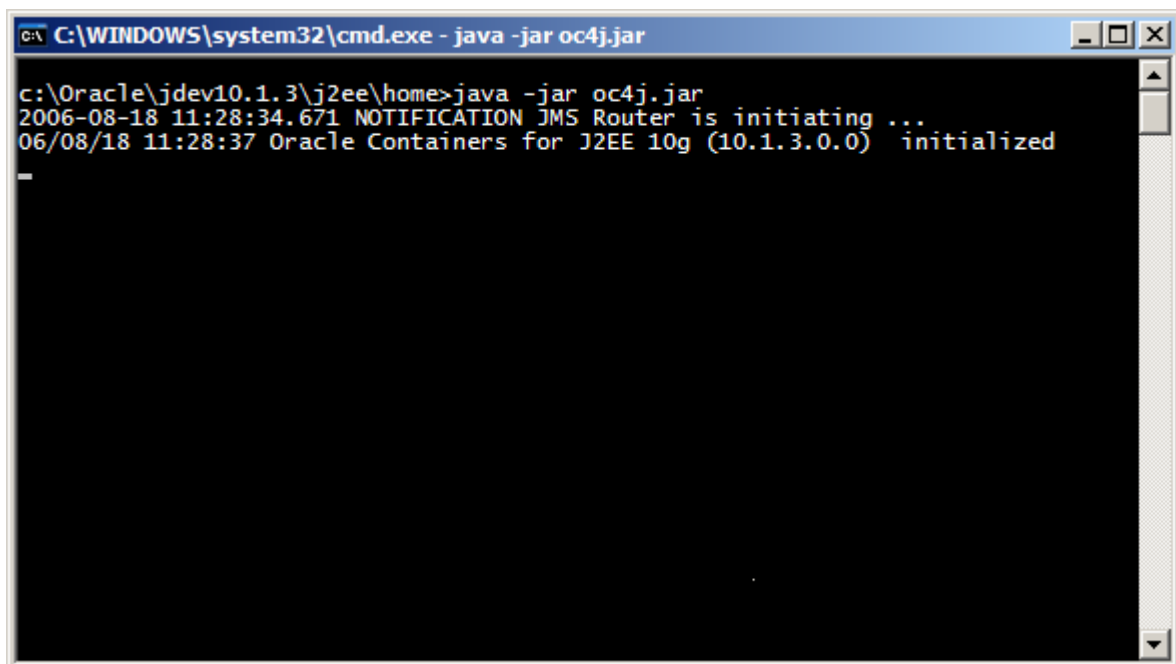


15. Kolejny krok to konfiguracja serwera aplikacji, w którym zostanie zainstalowana usługa sieciowa. Wykorzystamy do tego celu serwer OC4J dostarczany razem z narzędziem JDeveloper. Otwórz okno MS-DOS i przejdź do katalogu %JDEV_HOME%\j2ee\home\config. Domyślna konfiguracja serwera aplikacji OC4J posiada drobny błąd, który musimy skorygować. Otwórz plik server.xml i znajdź w nim znacznik **<java-compiler>**. Dodaj wewnątrz znacznika atrybut o nazwie bindir i wartości wskazującej na katalog zawierający kompilator javac.exe. Przykładowo, wpis w pliku server.xml mógłby wyglądać tak

```
<java-compiler name="javac" in-process="false"
  options="-J-Xmx1024m -encoding UTF8"
  bindir="C:\Program Files\Java\jdk1.5.0_07\bin"
  extdirs="C:\Program Files\Java\jre1.5.0_07\lib\ext" />
```

16. Przejdź do katalogu %JDEV_HOME%\j2ee\home i uruchom serwer aplikacji OC4J przez wydanie poniższego polecenia. Uwaga: przy pierwszym uruchomieniu serwera aplikacji nastąpi automatyczna instalacja wielu komponentów, dodatkowo instalator może się zapytać o hasło dla administratora. Proszę wówczas podać wybrane przez siebie hasło i je potwierdzić.

```
C:\> java -jar oc4j.jar
```

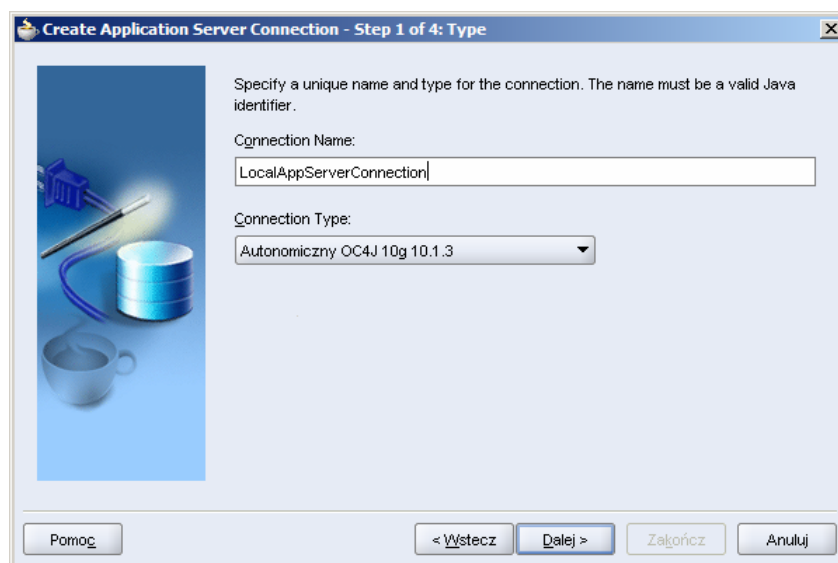


```
C:\WINDOWS\system32\cmd.exe - java -jar oc4j.jar
c:\Oracle\jdev10.1.3\j2ee\home>java -jar oc4j.jar
2006-08-18 11:28:34.671 NOTIFICATION JMS Router is initiating ...
06/08/18 11:28:37 Oracle Containers for J2EE 10g (10.1.3.0.0) initialized
```


17. Sprawdź, czy serwer aplikacji faktycznie się uruchomił poprawnie. W tym celu otwórz przeglądarkę i przejdź do adresu <http://localhost:8888>



18. Wróć do narzędzia JDeveloper. W nawigatorze obiektów (okno z lewej strony ekranu) przejdź na zakładkę Connections (alternatywnie, z menu głównego wybierz View→Connection Navigator). Kliknij prawym przyciskiem myszy na ikonie Application Server i z menu kontekstowego wybierz opcję New Application Server Connection. W pierwszym oknie asystenta kliknij przycisk **Dalej >**. W drugim oknie asystenta wpisz nazwę połączenia: LocalAppServerConnection oraz typ połączenia: Autonomiczny OC4J 10g 10.1.3. Kliknij przycisk **Dalej >**.



19. W kolejnym kroku wpisz hasło administratora serwera aplikacji OC4J. Kliknij przycisk **Dalej >**.

Create Application Server Connection - Step 2 of 4: Authentication

Specify a username and password to authenticate the connection. To bypass authentication at runtime, select Deploy Password.

Username:
oc4jadmin

Password:

Deploy Password

Pomoc < Wstecz **Dalej >** Zakończ Anuluj

20. W trzecim kroku asystenta pozostaw wszystkie wartości domyślne (lokalny serwer, brak ścieżki URL, domyślny port RMI). Kliknij przycisk **Dalej >**. W ostatnim kroku kliknij przycisk **Test Connection**. Kliknij przycisk **Zakończ**.

Create Application Server Connection - Step 4 of 4: Test

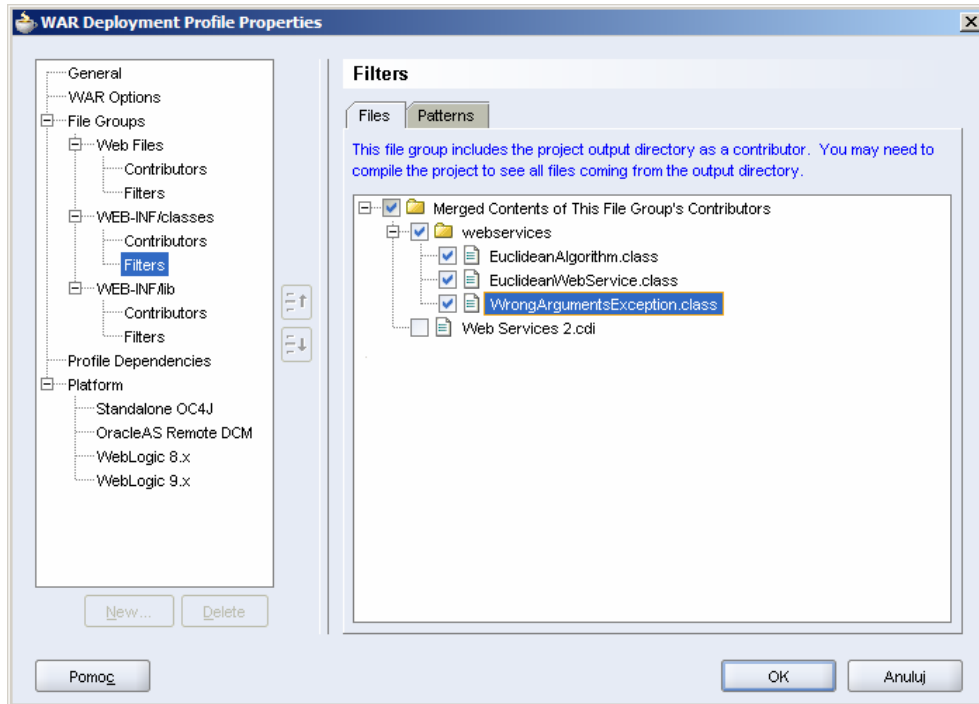
Click Test Connection to determine if the information specified successfully establishes a connection with the application server.

Test Connection

Status:
Success!

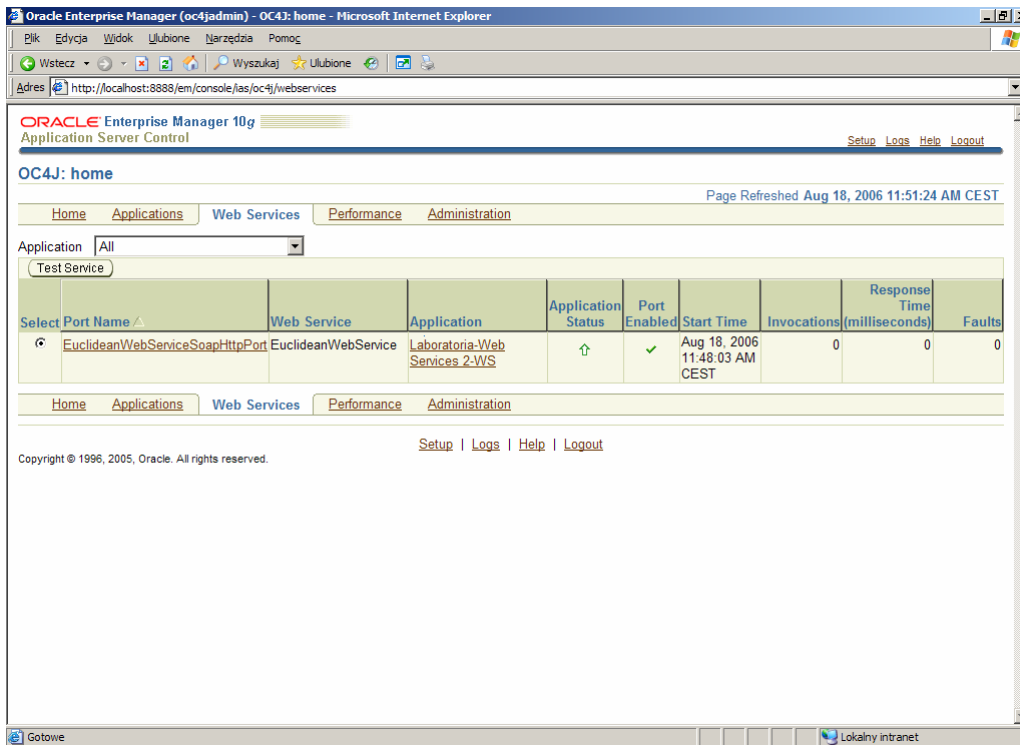
Pomoc < Wstecz **Dalej >** Zakończ Anuluj

21. Wróć do okna navigatora aplikacji (główne menu View→Application Navigator). Rozwiń gałąź Resources i kliknij dwukrotnie na pliku WebServices.deploy stanowiącym deskryptor instalacji usługi sieciowej na serwerze aplikacji. Przejdź do gałęzi File Groups→WEB-INF/classes→Filters i dołącz do instalacji plik WrongArgumentsException.class. Kliknij przycisk OK.

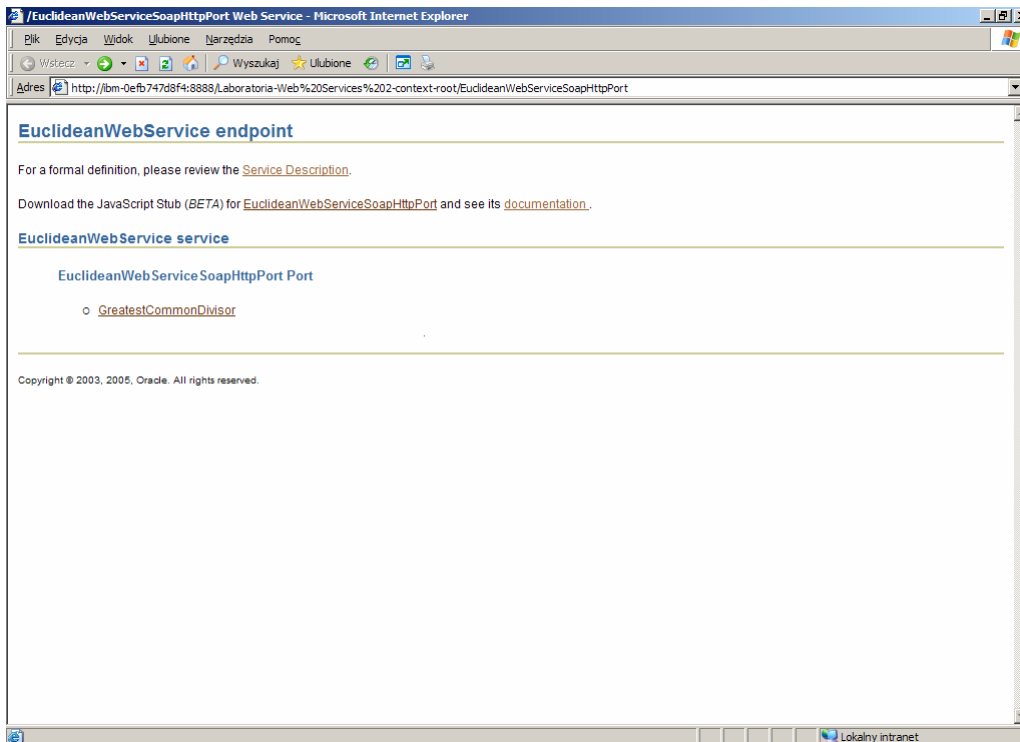


22. Kliknij prawym przyciskiem myszy na pliku WebServices.deploy i z menu kontekstowego wybierz Deploy to→LocalAppServerConnection. W oknie które się pojawi kliknij przycisk OK. Poczekaj na zakończenie procesu instalacji (możesz obserwować postęp instalacji w oknie dziennika u dołu ekranu).

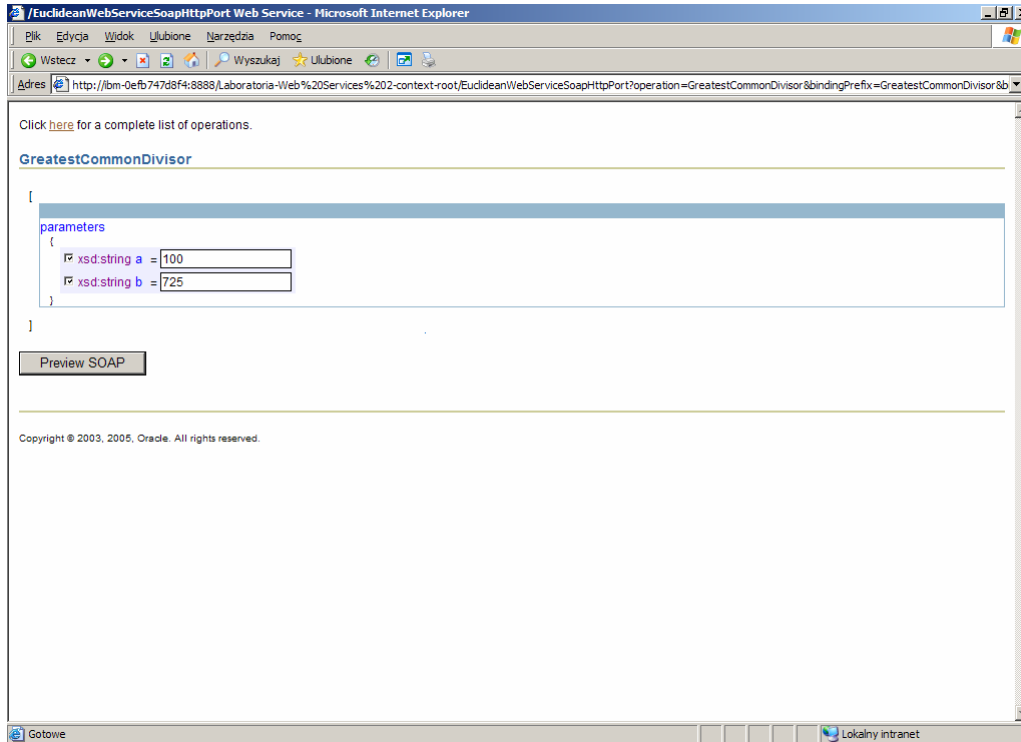
23. Wróć do okna przeglądarki i przejdź do adresu <http://localhost:8888/em>. Zaloguj się do serwisu jako administrator serwera aplikacji. Z głównego ekranu aplikacji wybierz odnośnik [Web Services](#).



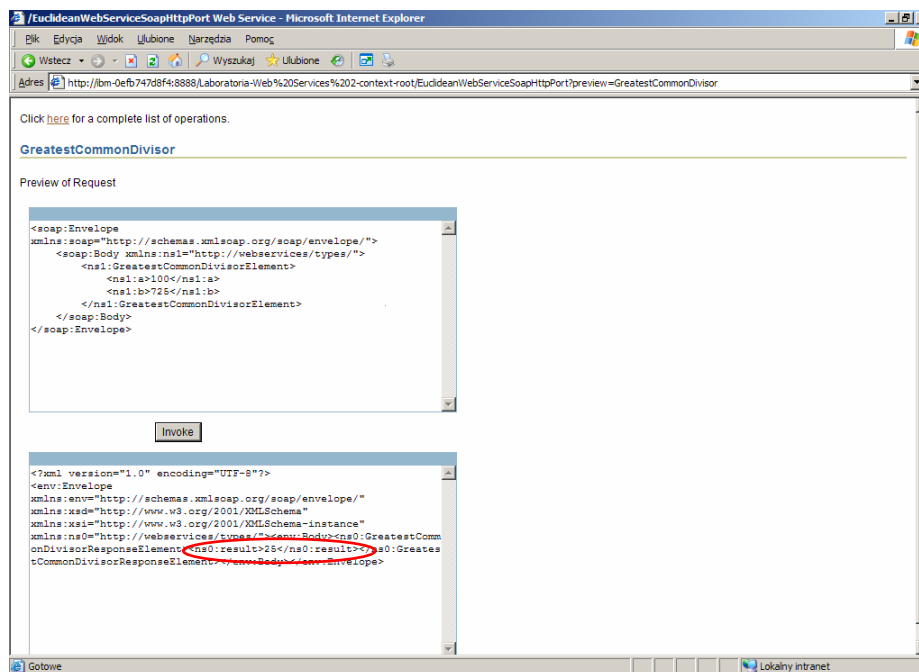
24. Kliknij przycisk **Test Service**. Na kolejnym ekranie kliknij przycisk **Test Web Service**. W przeglądarce zostanie otwarty domyślnie wygenerowany punkt dostępowy do usługi sieciowej.



25. Odnośnik [Service Description](#) prowadzi do pliku WSDL opisującego naszą usługę sieciową. Istnieje także możliwość pobrania pieńka (ang. stub) do usługi sieciowej przygotowanego w języku JavaScript. Dodatkowo, punkt dostępowy umożliwia przetestowanie usługi sieciowej. Kliknij na odnośnik [GreatestCommonDivisor](#), zaznacz pola wyboru przy obu parametrach i wpisz wartości testowe, 100 i 725.



26. Kliknij przycisk **Preview SOAP**. Obejrzyj postać koperty SOAP która zostanie wysłana do usługi sieciowej. Kliknij przycisk **Invoke** i obejrzyj zwróconą odpowiedź.



27. Powróć do okna umożliwiającego podawanie parametrów wywołania usługi sieciowej. Wpisz niepoprawny parametr, np. -5. Obejrzyj uzyskany wynik.

Click [here](#) for a complete list of operations.

GreatestCommonDivisor

Preview of Request

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns1="http://webservicess/types/">
    <ns1:GreatestCommonDivisorElement>
      <ns1:a>100</ns1:a>
      <ns1:b>5</ns1:b>
    </ns1:GreatestCommonDivisorElement>
  </soap:Body>
</soap:Envelope>
```

Invoke

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns0="http://webservicess/types/"><env:Body><env:Fault
xsi:type="env:Fault"><faultcode>env:Server</faultcode><faultstring>Input
parameters must be greater than 0</faultstring><detail><ns0:WrongArgumentsExceptionElement><ns0:
message>input parameters must be greater than 0</ns0:message></ns0:WrongArgumentsExceptionElement></detail></env:
Fault></env:Body></env:Envelope>
```

Gotowe

Lokalny intranet