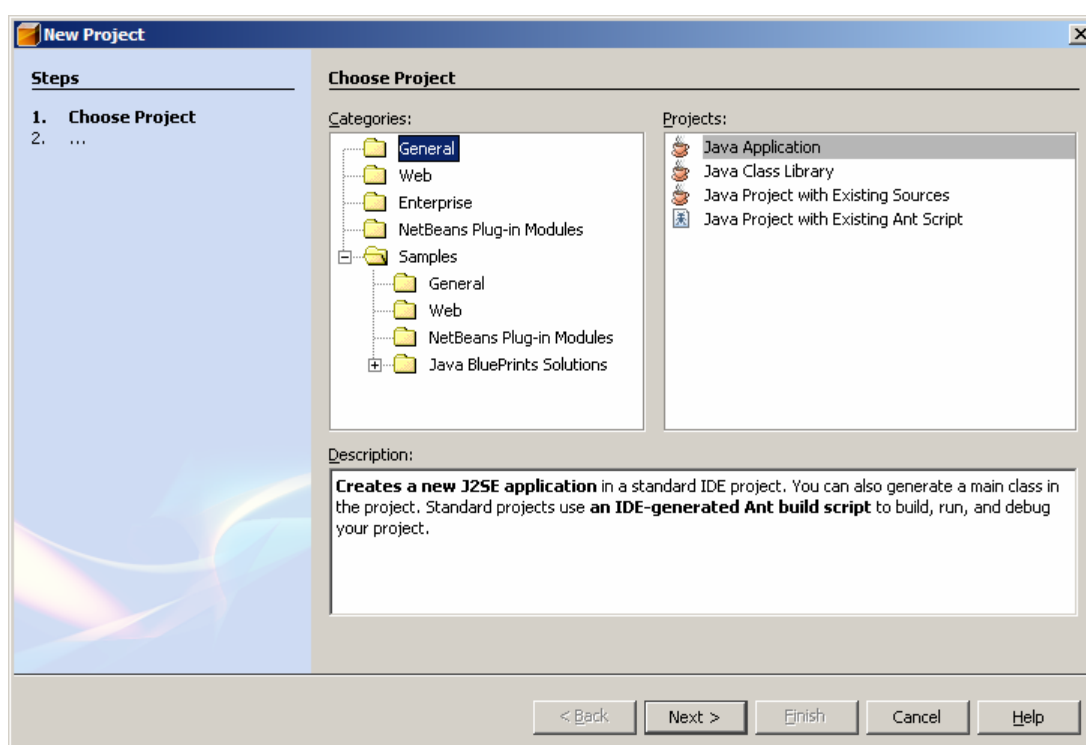


Zaawansowane aplikacje internetowe - laboratorium

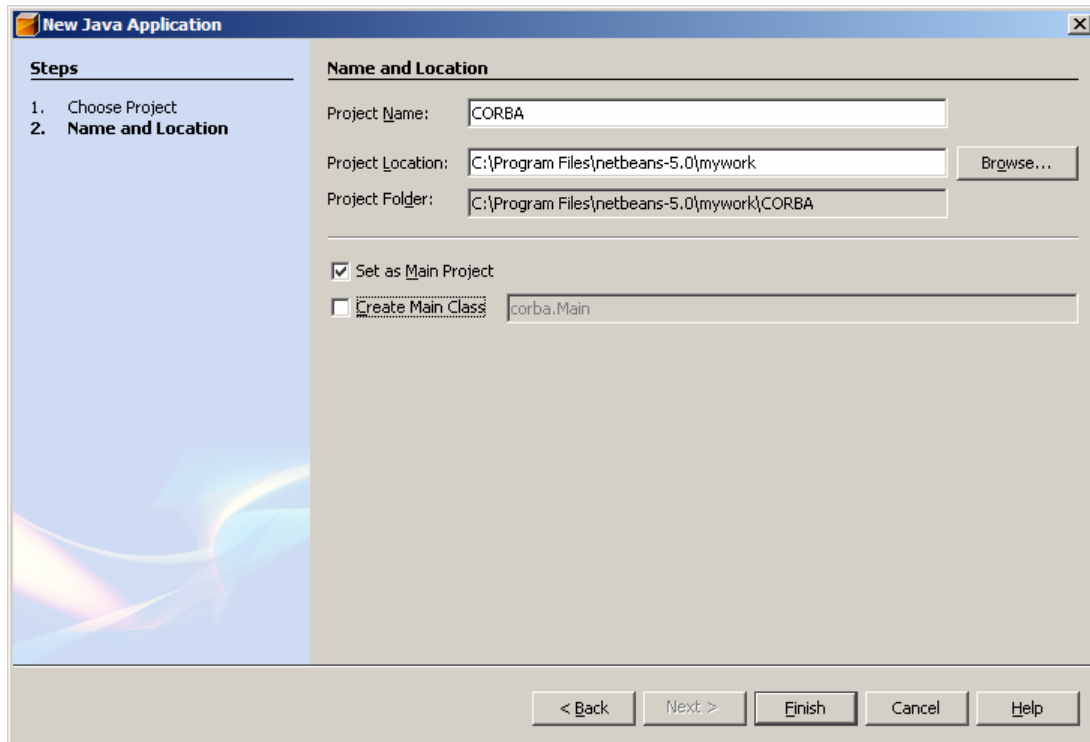
Architektura CORBA.

Celem ćwiczenia jest przygotowanie prostej aplikacji wykorzystującej architekturę CORBA. Aplikacja składa się z usługodawcy (serwera) i klienta. Usługa polega na podaniu bieżącego czasu. Do wykonania ćwiczenia potrzebne jest zintegrowane środowisko programistyczne NetBeans IDE 5.0 (do pobrania z <http://www.netbeans.org>) oraz środowisko J2SE 1.4.2 (lub wyższe).

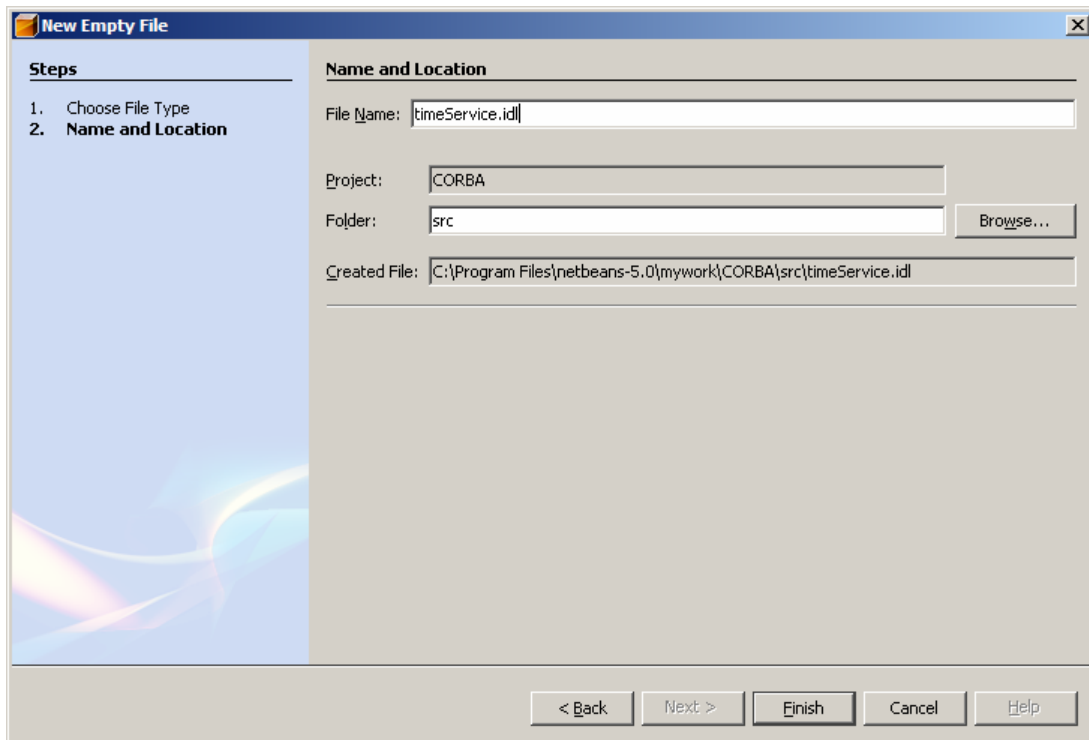
1. Uruchom narzędzie NetBeans IDE 5.0
2. Z menu głównego wybierz File→New Project. Wybierz kategorię General i typ projektu Java Application. Kliknij przycisk Next >.



3. Podaj nazwę projektu, np. „CORBA”. Odznacz pole wyboru przy opcji Create Main Class. Kliknij przycisk Finish.



4. Kliknij prawym przyciskiem myszy na nazwie ikonie Source Packages i z menu kontekstowego wybierz New→Empty File... Wprowadź nazwę pliku timeService.idl. Kliknij przycisk Finish.

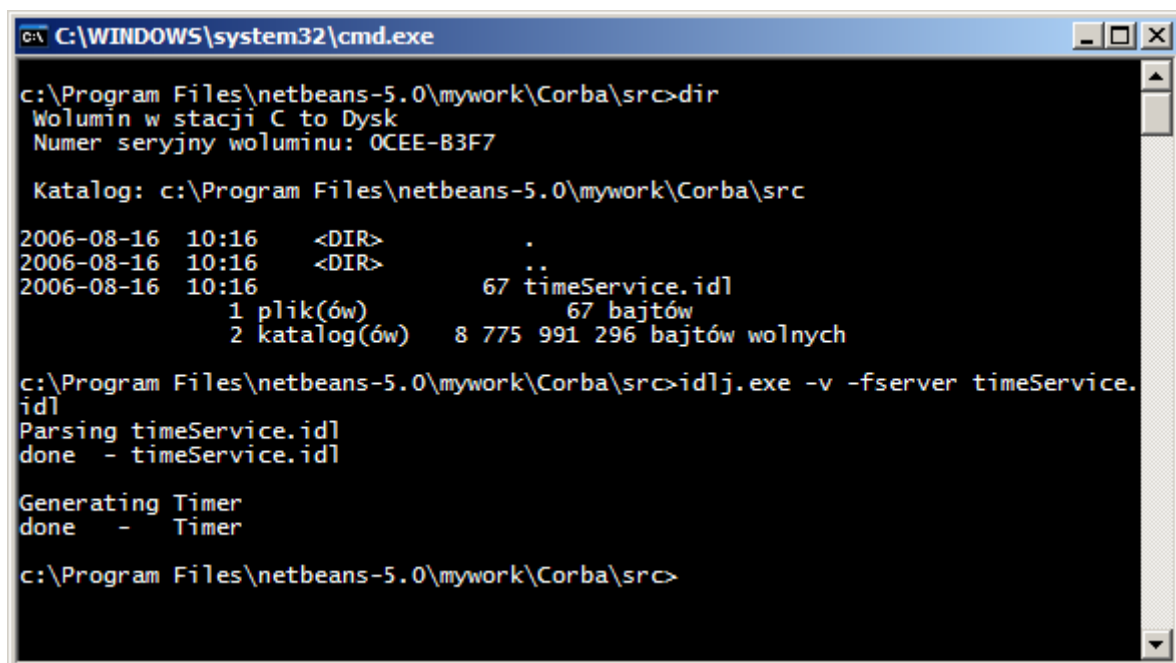


5. Plik `timeService.idl` jest otwarty w głównym oknie do edycji. Umieść w pliku poniższy kod IDL i zapisz zmiany.

```
module Timer {  
    interface TimeService {  
        string getTime();  
    };  
};
```

6. Uruchom konsolę MS-DOS i przejdź do katalogu zawierającego źródła aplikacji, `%NETBEANS_HOME%\mywork\CORBA\src` (gdzie `%NETBEANS_HOME%` to lokalny katalog zawierający instalację NetBeans IDE 5.0. Wykorzystaj prekompilator `idlj.exe` do wygenerowania plików po stronie serwera przez wydanie komendy

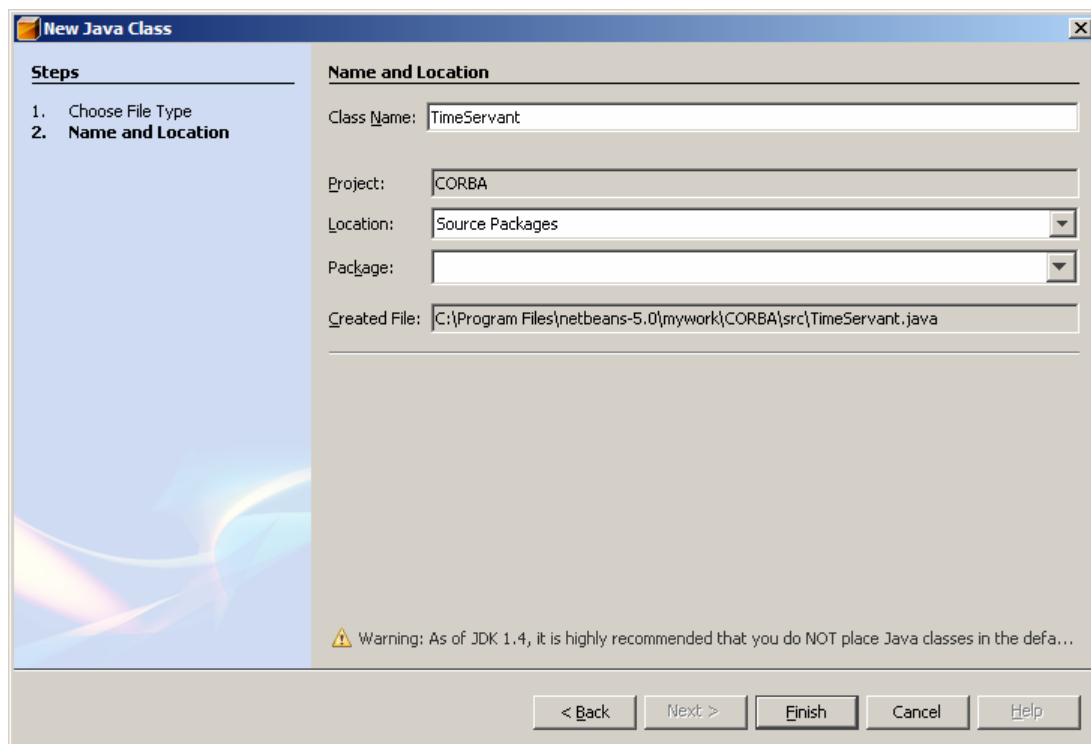
```
c:\> idlj.exe -v -fserver timeService.idl
```



```
C:\WINDOWS\system32\cmd.exe  
c:\Program Files\netbeans-5.0\mywork\Corba\src>dir  
Wolumin w stacji C to Dysk  
Numer seryjny woluminu: 0CEE-B3F7  
  
Katalog: c:\Program Files\netbeans-5.0\mywork\Corba\src  
2006-08-16 10:16 <DIR> .  
2006-08-16 10:16 <DIR> ..  
2006-08-16 10:16 67 timeService.idl  
1 plik(ów) 67 bajtów  
2 katalog(ów) 8 775 991 296 bajtów wolnych  
  
c:\Program Files\netbeans-5.0\mywork\Corba\src>idlj.exe -v -fserver timeService.  
idl  
Parsing timeService.idl  
done - timeService.idl  
  
Generating Timer  
done - Timer  
  
c:\Program Files\netbeans-5.0\mywork\Corba\src>
```

7. Wróć do środowiska NetBeans IDE 5.0. Zauważ, że w drzewie Source Packages pojawił się pakiet o Timer. Rozwiń zawartość pakietu i kliknij dwukrotnie na każdym pliku, obejrzyj zawartość każdego pliku.

8. Kliknij prawym przyciskiem myszy na ikonie Source Packages i z menu kontekstowego wybierz opcję **New→Java Class...** Wprowadź nazwę klasy `TimeServant`. Kliknij przycisk **Finish**.



9. Plik `TimeServant.java` jest otwarty w głównym edytorze do edycji. Klasa `TimeServant` posłuży jako główna implementacja usługi. Wprowadź do pliku następujący kod.

```
import Timer.*;

import java.text.*;
import java.util.*;
import org.omg.CORBA.*;

public class TimeServant extends TimeServicePOA {
    private ORB orb;

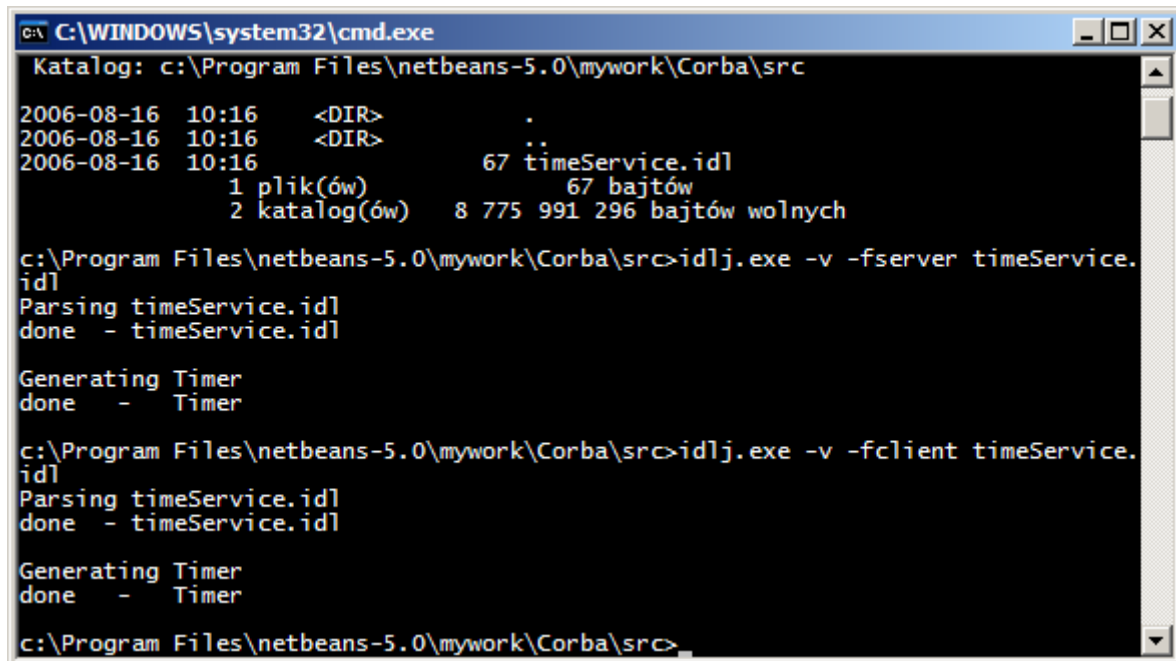
    public TimeServant() {
    }

    public void setORB(ORB orb) {
        this.ORB = orb;
    }

    public String getTime() {
        SimpleDateFormat formatter =
            new SimpleDateFormat("dd-MM-yyyy, HH:mm:ss:SSS z");
        Date now = new Date();
        return formatter.format(now);
    }
}
```

10. Dostęp do obiektu usługodawcy wymaga wygenerowania odpowiednich pniaków i klas pomocniczych. Wróć do okna MS-DOS i wydaj komendę

```
c:\> idlj.exe -v -fclient timeService.idl
```



```
C:\WINDOWS\system32\cmd.exe
Katalog: c:\Program Files\netbeans-5.0\mywork\Corba\src
2006-08-16 10:16 <DIR>      .
2006-08-16 10:16 <DIR>      ..
2006-08-16 10:16           67 timeService.idl
                1 plik(ów)      67 bajtów
                2 katalog(ów) 8 775 991 296 bajtów wolnych

c:\Program Files\netbeans-5.0\mywork\Corba\src>idlj.exe -v -fserver timeService.
idl
Parsing timeService.idl
done - timeService.idl

Generating Timer
done - Timer

c:\Program Files\netbeans-5.0\mywork\Corba\src>idlj.exe -v -fclient timeService.
idl
Parsing timeService.idl
done - timeService.idl

Generating Timer
done - Timer

c:\Program Files\netbeans-5.0\mywork\Corba\src>
```

11. Kolejny krok to przygotowanie programu serwera, który powoła do życia instancję usługodawcy (TimeServant), zarejestruje usługodawcę w brokerze obiektów i aktywizuje brokera. Kliknij prawym przyciskiem myszy na ikonie Source Packages i z menu kontekstowego wybierz opcję New→Java Class... Wprowadź nazwę klasy Server. Kliknij przycisk Finish. Plik Server.java zostanie otwarty w edytorze kodu. Wprowadź do pliku następujący kod.

```
import Timer.*;

import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextPackage.*;
import org.omg.CORBA.*;
import org.omg.PortableServer.*;

import java.util.*;
import java.text.*;

public class Server {

    public Server() {

    }

    public static void main(String args[]) {
    try {
        Properties props = new Properties();
        props.put("org.omg.CORBA.ORBInitialPort", "1050");
        props.put("org.omg.CORBA.ORBInitialHost", "localhost");
        ORB orb = ORB.init(args, props);

        POA rootpoa
            = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
    }
    }
}
```

```

rootpoa.the_POAManager().activate();

TimeServant timeServant = new TimeServant();
timeServant.setORB(orb);

org.omg.CORBA.Object ref = rootpoa.servant_to_reference(timeServant);
TimeService href = TimeServiceHelper.narrow(ref);

org.omg.CORBA.Object objRef =
    orb.resolve_initial_references("NameService");
NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);

String name = "TimeServant";
NameComponent path[] = ncRef.to_name(name);
ncRef.rebind(path, href);

System.out.println("TimeServant ready and waiting ...");

orb.run();
}

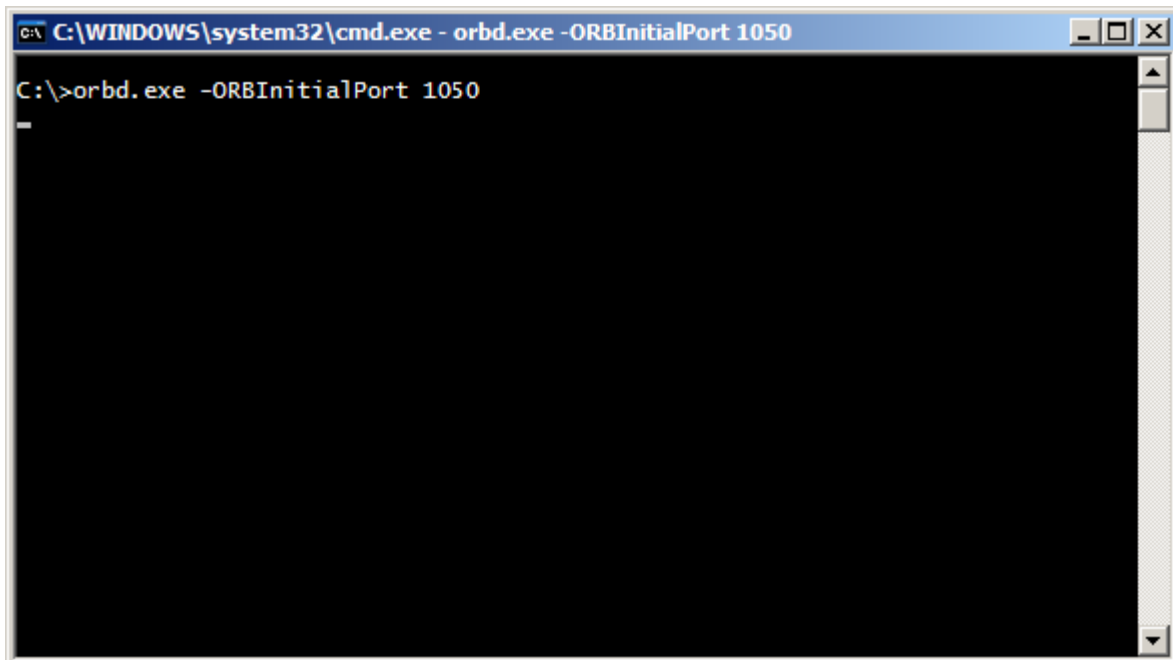
catch (Exception e) { e.printStackTrace(System.out); }

System.out.println("TimeServant Exiting ...");
}
}

```

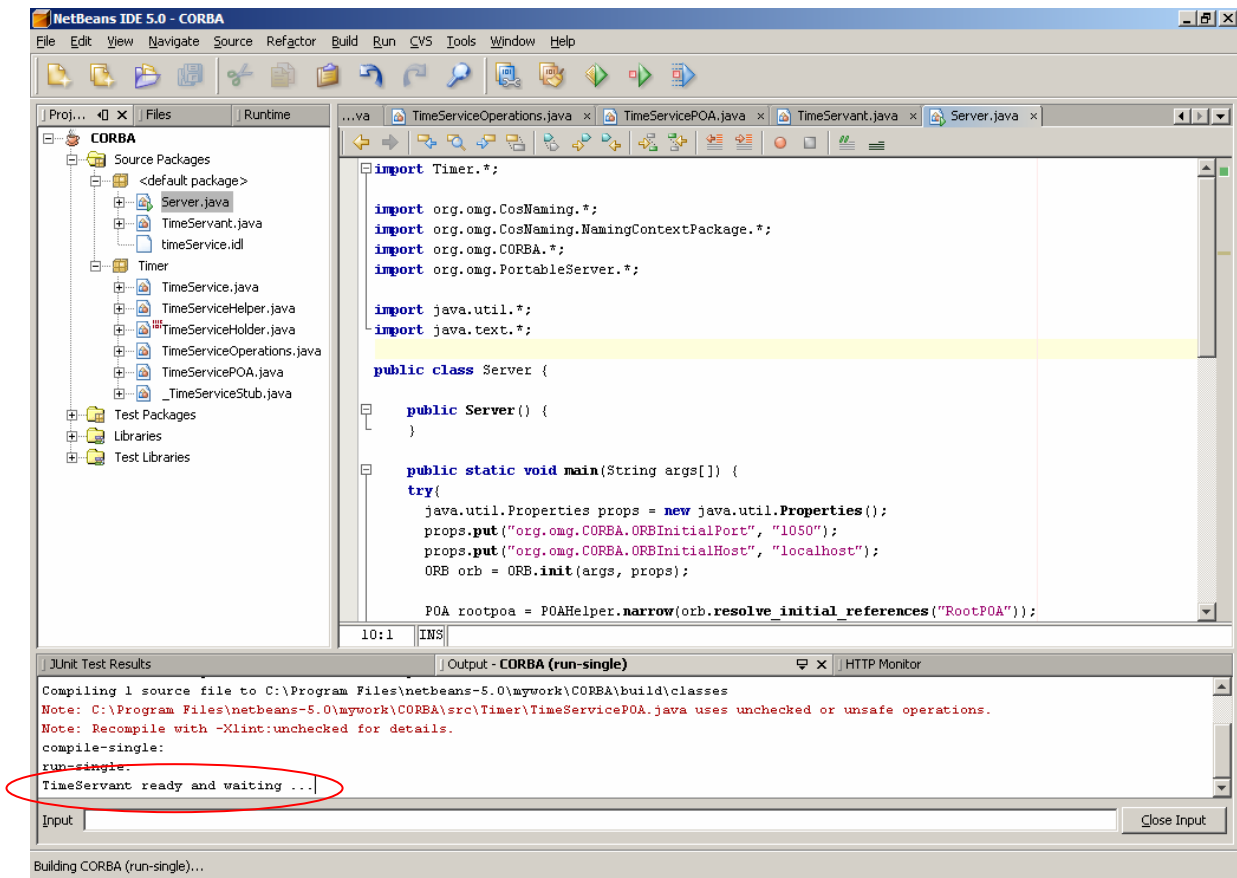
12. :Uruchom dodatkowe okno MS-DOS. Uruchom brokera obiektów ORB za pomocą komendy

```
c:\> orbd.exe -ORBInitialPort 1050
```



The image shows a screenshot of a Windows command prompt window. The title bar reads "C:\WINDOWS\system32\cmd.exe - orbd.exe -ORBInitialPort 1050". The command prompt shows the command "C:\>orbd.exe -ORBInitialPort 1050" being entered and executed. A small horizontal line is visible below the command, indicating the start of the program's output.

13. Wróć do środowiska NetBeans IDE 5.0. Kliknij prawym przyciskiem myszy gdziekolwiek w edytorze zawierającym otwarty plik `Server.java`. Z menu kontekstowego wybierz opcję `Run File`.



14. Ostatnia część ćwiczenia to przygotowanie programu klienta, który połączy się z usługą i pobierze aktualny czas. Kliknij prawym przyciskiem myszy na ikonie `Source Packages` i z menu kontekstowego wybierz `New→Java Class...` Wprowadź nazwę klasy `Client`. Kliknij przycisk `Finish`. Plik `Client.java` zostanie otwarty w edytorze kodu. Wprowadź do pliku następujący kod.

```
import Timer.*;
import org.omg.CosNaming.*;
import org.omg.CORBA.*;
import java.util.Properties;

public class Client {

    public static void main(String args[] ) {
        try {
            Properties props = new Properties();
            props.put("org.omg.CORBA.ORBInitialPort", "1050");
            props.put("org.omg.CORBA.ORBInitialHost", "localhost");
            ORB orb = ORB.init(args, props);

            org.omg.CORBA.Object objRef =
                orb.resolve_initial_references("NameService");
            NamingContext ncRef = NamingContextHelper.narrow(objRef);

            NameComponent nc = new NameComponent("TimeServant", "");
        }
    }
}
```

```
NameComponent path[] = {nc};
TimeService timeRef = TimeServiceHelper.narrow(ncRef.resolve(path));

String time = "Time on the server: " + timeRef.getTime();
System.out.println(time);

} catch (Exception e) { e.printStackTrace(); }
}
```

15. Uruchom program klienta przez kliknięcie prawym przyciskiem myszy w oknie edycji pliku Client.java i wybranie z menu kontekstowego opcji Run File.

