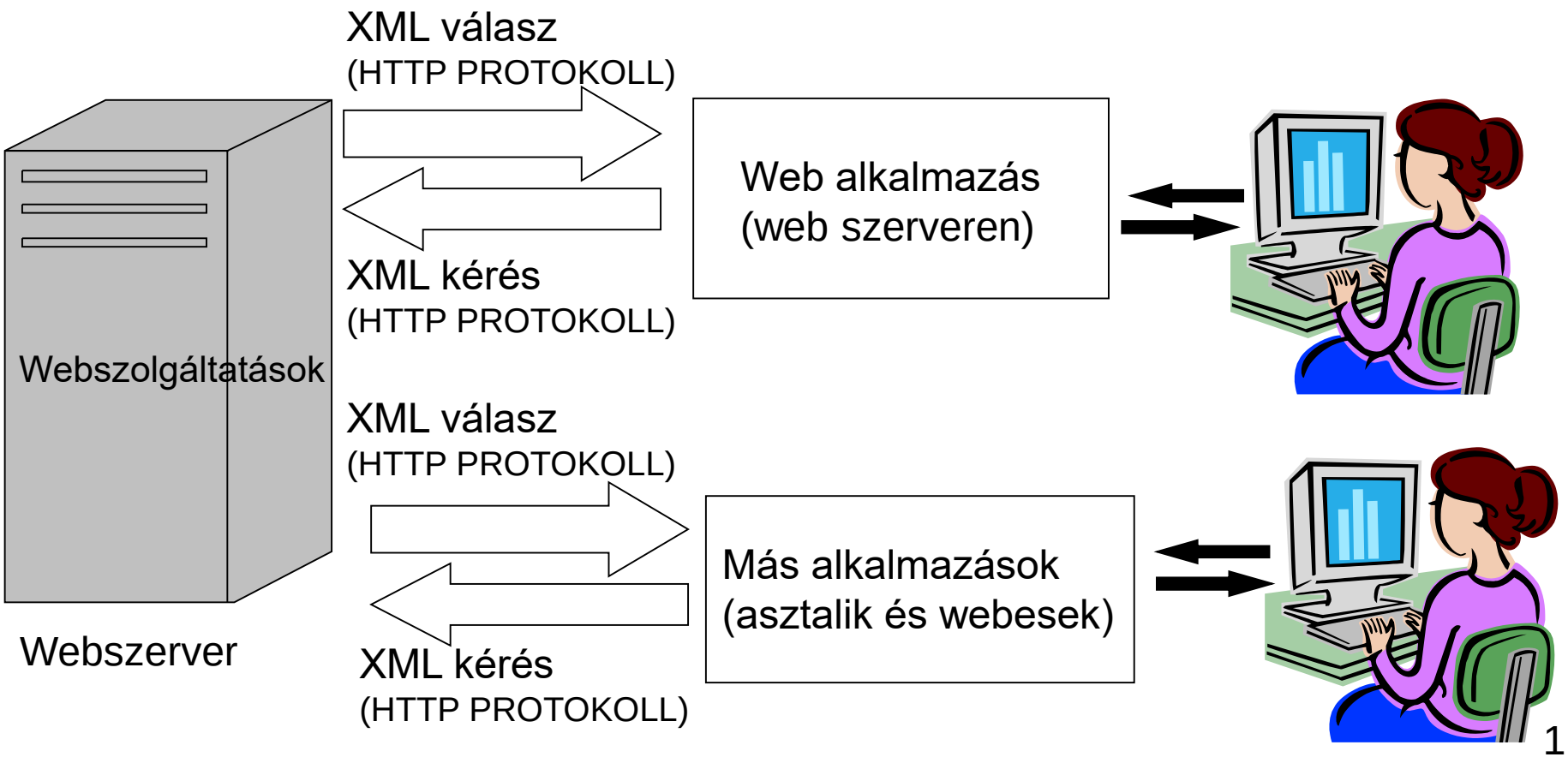
WEB-programozás II (3. előadás)

WEBSZOLGÁLTATÁSOK PHP-BEN

A **webszolgáltatás** (angolul **webservice**) alkalmazások közötti adatcserére szolgáló protokollok és szabványok gyűjteménye.

Különböző programnyelveken írt és különböző platformokon futó szoftveralkalmazások számítógép-hálózatokon (mint az Internet) keresztül történő adatcserére használják a webszolgáltatásokat.



WEB-programozás II (3. előadás)
WEBSZOLGÁLTATÁSOK PHP-BEN

Egy webszolgáltatás egy olyan szoftver, amely elérhető web-en keresztül és XML alapú szabványos üzeneteken keresztül kommunikál más szoftverekkel (kérelmeket fogad és válaszokat küld).

Tulajdonságok:	**Megvalósítás:**
• **Alapkoncepciója az RPC (Remote Procedure Call).** • **Új elem a platformfüggetlen szabványok használata, XML-re épülő technológia.** • **Megvalósítási elemek elrejtése a kliens alkalmazás elől.** • **Működésük az internetre jellemző heterogén környezetben is garantált.**	• **Szolgáltatás-átvitel (kommunikációs protokoll): HTTP protokoll.** • **XML alapú üzenetkezelés: SOAP (Simple Object Acces Protocoll).** • **Szolgáltatás XML alapú leírása: WSDL (Web Services Description Language).** • **Szolgáltatás felkutatása: UDDI (Universal Description, Discovery and Integration) http://uddi.xml.org/**

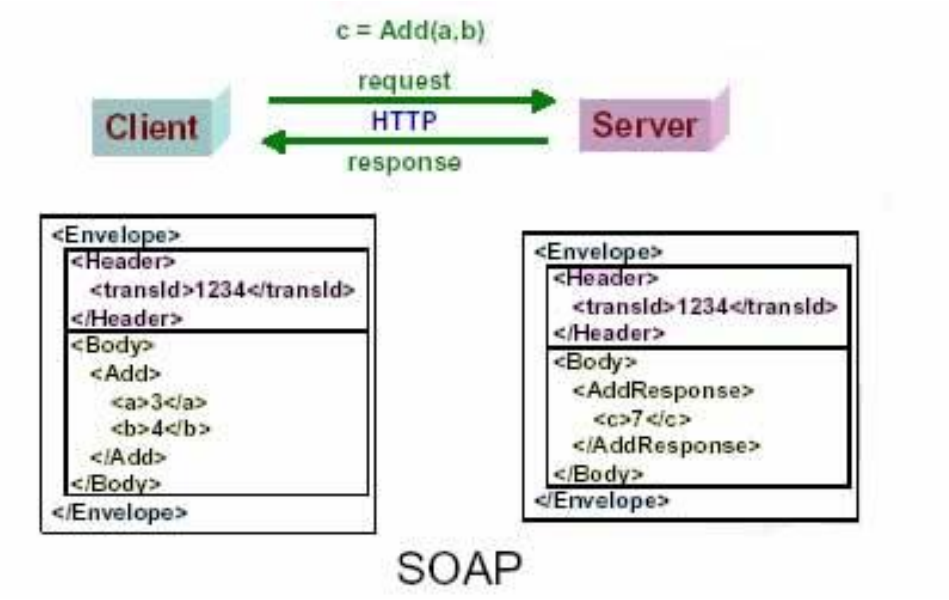
WEB-programozás II (3. előadás)
WEBSZOLGÁLTATÁSOK PHP-BEN

Előnyök:	Hátrányok:
•A webszolgáltatások együttműködést biztosítanak a különböző platformokon futó szoftver alkalmazások között. •A webszolgáltatás nyílt szabványokat és protokollokat használ. A protokollok és adatok minden lehetséges helyen szöveg alapúak, így egyszerűsítve a fejlesztők feladatát. •A HTTP használatával a webszolgáltatások keresztüljutnak a legtöbb tűzfalon a tűzfal paramétereinek megváltoztatása nélkül. •A webszolgáltatások egyszerű módon teszik lehetővé különböző gyártóktól származó szoftverek és szolgáltatások kombinálását új, integrált szolgáltatások létrehozására. •A webszolgáltatások lehetővé teszik a szolgáltatások és komponensek újrafelhasználását egy infrastruktúrán belül.	•A webszolgáltatás szabványok olyan téren, mint a tranzakciókezelés, vagy nem képesek megoldással szolgálni, vagy kényelmetlen a használatuk az olyan kiforrott elosztott rendszereket leíró nyílt szabványokhoz képest, mint CORBA. •A webszolgáltatás lassabb, mint az olyan elosztott modellek, mint a DCOM, a CORBA vagy az RMI. Ez a szöveg-alapú formátummal járó hátrány. Az XML tervezési céljai nem tartalmazzák a kódolás tömörségét vagy a feldolgozás hatékonyságát. •A HTTP protokoll használatának következtében a webszolgáltatások át tudnak jutni a tűzfalakon, melyeknek célja a programok közti kommunikáció blokkolása vagy ellenőrzése.

WEB-programozás II (3. előadás)
WEBSZOLGÁLTATÁSOK PHP-BEN

SOAP: XML alapú protokoll alkalmazások közötti üzenetváltásra.

A SOAP szabvány szerint szerkesztett üzenetek a HTTP protokoll segítségével POST kérelmekként továbbíthatók és fogadhatók.



WEB-programozás II (3. előadás)
WEBSZOLGÁLTATÁSOK PHP-BEN

Webszolgáltatások a PHP SOAP könyvtárával (PHP 5.3+):

Távoli webszolgáltatások használata PHP alkalmazásokban

```
<?php
   $client = new SoapClient('http://api.radioreference.com/soap2/?wsdl&v=latest');
   $countries = $client->getCountryList();
   echo "<pre>";
   $types = $client->__getFunctions();
   var_dump($types);
   echo "<br>";
   var_dump($countries);
   echo "</pre>";
?>
```

WEB-programozás II (3. előadás)
WEBSZOLGÁLTATÁSOK PHP-BEN

Webszolgáltatások megvalósítása a PHP SOAP könyvtárával:

a) Webszogáltatások WSDL nélkül (non WSDL)

Példa helye: http://localhost/ws/php-soap/non-wsdl

Szerver: soap-server.php

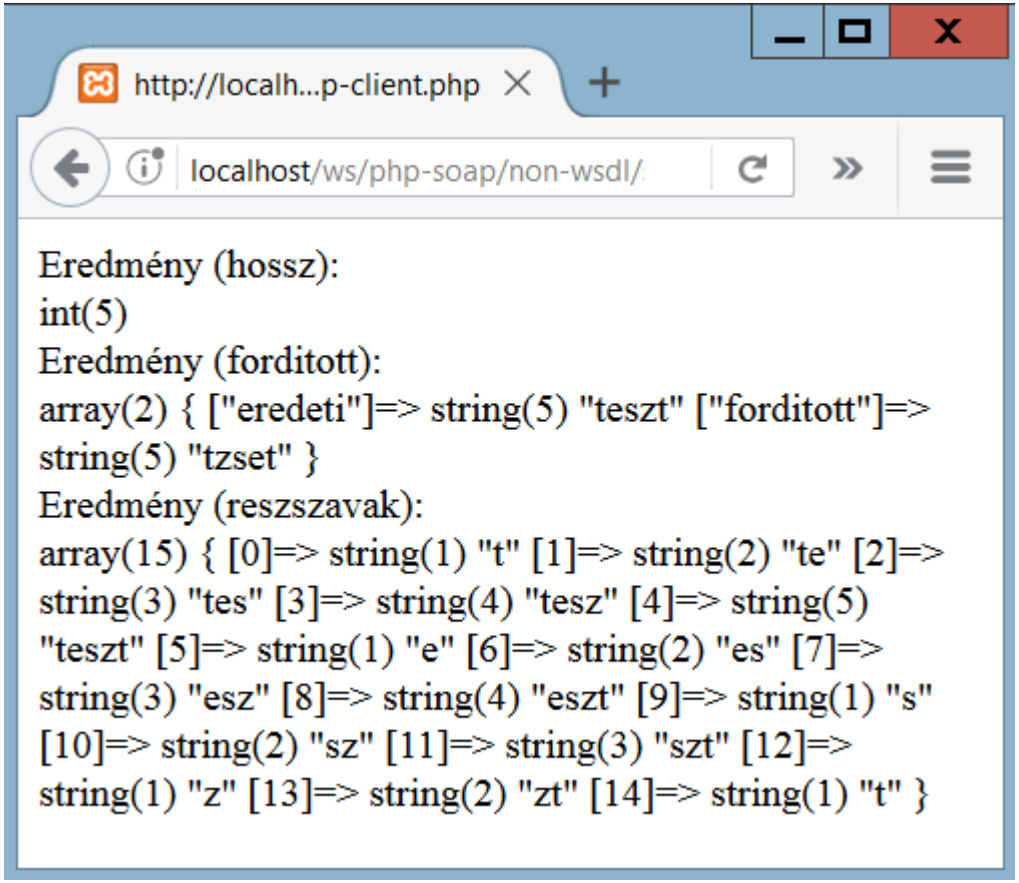
```
<?php

class Szavak {
   public function hossz($szo) {
      $eredmeny = strlen($szo);
      return $eredmeny;
   }

   public function forditott($szo) {
      $forditott = "";
      for($i=strlen($szo)-1; $i>=0; $i--)
         $forditott .= $szo[$i];
      $eredmeny = Array(
                        "eredeti"=>$szo,
                        "forditott"=>$forditott
                      );
      return $eredmeny;
   }

   public function reszszavak($szo) {
      $eredmeny = Array();
      for($i=0; $i<strlen($szo); $i++)
         for($j=1; $j<=strlen($szo)-$i; $j++)
            $eredmeny[ ] = substr($szo, $i, $j);
      return $eredmeny;
   }
}

$options = array(
            "location" => "http://localhost/ws/
php-soap/non-wsdl/soap-server.php",
            "uri" => "http://localhost");
$server = new SoapServer(null, $options);
$server->setClass('Szavak');
$server->handle();

?>
```

WEB-programozás II (3. előadás)
WEBSZOLGÁLTATÁSOK PHP-BEN

Webszolgáltatások megvalósítása a PHP SOAP könyvtárával:

Kliens: soap-client.php

```
<?php
  $options = array("location" => "http://localhost/ws/php-soap/non-wsdl/soap-server.php",
                   "uri" => "http://localhost");

  try {

      $client = new SoapClient(null, $options);
      $szo = "teszt";

      $hossz = $client->hossz($szo);
      echo "Eredmény (hossz):<br>";
      var_dump($hossz);
      echo "<br>";

      $forditott = $client->forditott($szo);
      echo "Eredmény (forditott):<br>";
      var_dump($forditott);
      echo "<br>";

      $reszszavak = $client->reszszavak($szo);
      echo "Eredmény (reszszavak):<br>";
      var_dump($reszszavak);
      echo "<br>";
  } catch (SoapFault $e) {
      var_dump($e);
  }
?>
```



WEB-programozás II (3. előadás)

WEBSZOLGÁLTATÁSOK PHP-BEN

Webszolgáltatások megvalósítása a PHP SOAP könyvtárával:

b) Webszogáltatások WSDL-lel (a műveleteket függvények valósítják meg)

Példa helye: http://localhost/ws/php-soap/wsdl

Szerver: soap-server.php

```
<?php

function hossz($szo) {
        echo $szo;
        $eredmeny = strlen($szo);
        return $eredmeny;
}

function forditott($szo) {
        $forditott = "";
        for($i=strlen($szo)-1; $i>=0; $i--)
            $forditott .= $szo[$i];
        $eredmeny = Array(
                              "eredeti"=>$szo,
                              "forditott"=>$forditott
                            );
        return $eredmeny;
}
```

```
function reszszavak($szo) {
        $eredmeny = Array();
        for($i=0; $i<strlen($szo); $i++)
            for($j=1; $j<=strlen($szo)-$i; $j++)
                $eredmeny[ ] = substr($szo, $i, $j);
        return $eredmeny;
}

$server = new SoapServer("words.wsdl");
$server->addFunction('hossz');
$server->addFunction('fordito');
$server->addFunction('reszszavak');
$server->handle();

?>
```

WEB-programozás II (3. előadás)

WEBSZOLGÁLTATÁSOK PHP-BEN

Webszolgáltatások megvalósítása a PHP SOAP könyvtárával:

WSDL: Web-szolgáltatásokat leíró nyelv

Egy WSDL leírás egy XML dokumentum. A dokumentum leírja a szolgáltatást, megadja a helyét és a szolgáltatás által kínált műveleteket, metódusokat.

Egy **WSDL** fájl struktúrája:

<definitions>
    < types>
        adattípusok definíciói
    < /types>
    < message>
        üzenetek definíciói (mindenegyes üzenethez egy ilyen szekció)
    < /message>
    < portType>
        műveletek megadása
    < /portType>
    < binding>
        protokoll és adatformátumok specifikációi
    < /binding>
    < service>
        a szolgáltatás elérhetőségének a megadása
    < /service>
< /definitions>

WEB-programozás II (3. előadás)
WEBSZOLGÁLTATÁSOK PHP-BEN

WSDL fájl (words.wsdl):

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    targetNamespace="http://localhost/ws/php-soap/wsdl/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://localhost/ws/php-soap/wsdl/"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
```

WEB-programozás II (3. előadás)
WEBSZOLGÁLTATÁSOK PHP-BEN

Az első szekció a típusdefiníciókat tartalmazza. Ehhez nézzük át a három művelet bemenő és kimenő értékeit:

•Ebben az esetben mind a 3 műveletnek a bemenő paramétere egy karaktersorozat, amely egyszerű típus, és nem kell definiálni.

•A ***hossz*** művelet kimenete egy egész szám, amely egyszerű típus, és szintén nem kell definiálni.

•A ***forditott*** művelet kimenete struktúra (több névvel rendelkező mezőből álló típus), amely az ***eredeti*** és a ***forditott*** karaktersorozat típusú mezőkből áll. Ez egy összetett típus („complex type”), amelyet definiálnunk kell. Definiálni fogjuk a a ***szopar*** nevű struktúra típusú összetett típust.

•A ***reszszavak*** művelet kimenete egy tömb, amelynek elemeinek száma előre nem meghatározott, és a típusuk karaktersorozat. Ez ís egy összetett típus, amelyet definiálnunk kell. Definiálni fogjuk a ***stringek*** nevű tömb típusú összetett típust.

```
<types>
  <xsd:schema targetNamespace="http://localhost/ws/php-soap/wsdl/">
    <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
    <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/"/>
    <xsd:complexType name="szopar">
      <xsd:all>
      <xsd:element name="eredeti" type="xsd:string"/>
      <xsd:element name="forditott" type="xsd:string"/>
      </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="stringek">
        <xsd:complexContent>
          <xsd:restriction base="SOAP-ENC:Array">
            <xsd:attribute wsdl:arrayType="xsd:string[]" ref="SOAP-ENC:arrayType"/>
          </xsd:restriction>
        </xsd:complexContent>
    </xsd:complexType>
  </xsd:schema>
</types>
```

WEB-programozás II (3. előadás)
WEBSZOLGÁLTATÁSOK PHP-BEN

Mindenegyes művelethez rendelünk egy input és egy output üzenetet.

Ezeket az üzeneteket definiálni kell a művelethez való rendelés előtt.

Célszerű az üzeneteket úgy nevezni, hogy könnyen azonosítható legyen, hogy melyik művelethez tartozik, és milyen típusú (input vagy output).

Több művelet ugyanazt az üzenetet használhatja, pl. ebben a példában lehetett volna definiálni egyetlen egy üzenetet mindem művelet input üzenetének:

```
<message name="mindRequest">
    <part name="szo" type="xsd:string"/>
</message>
```

```
<message name="hosszRequest">
    <part name="szo" type="xsd:string"/>
</message>
<message name="hosszResponse">
    <part name="eredmeny" type="xsd:int"/>
</message>
<message name="forditottRequest">
    <part name="szo" type="xsd:string"/>
</message>
<message name="forditottResponse">
    <part name="eredmeny" type="tns:szopar"/>
</message>
<message name="reszszavakRequest">
    <part name="szo" type="xsd:string"/>
</message>
<message name="reszszavakResponse">
    <part name="eredmeny" type="tns:stringek"/>
</message>
```

WEB-programozás II (3. előadás)
WEBSZOLGÁLTATÁSOK PHP-BEN

A ***portType*** szekcióban definiáljuk a műveleteket. Itt megadjuk a műveletek neveit és rendeljük hozzá a már definiált üzeneteket a műveletekhez:

```
<portType name="WordOperationsPortType">
    <operation name="hossz">
        <input message="tns:hosszRequest"/>
        <output message="tns:hosszResponse"/>
    </operation>
    <operation name="forditott">
        <input message="tns:forditottRequest"/>
        <output message="tns:forditottResponse"/>
    </operation>
    <operation name="reszszavak">
        <input message="tns:reszszavakRequest"/>
        <output message="tns:reszszavakResponse"/>
    </operation>
</portType>
```

WEB-programozás II (3. előadás)

WEBSZOLGÁLTATÁSOK PHP-BEN

A ***binding*** (kötések) szekció: (a ***binding type*** attribútumának az értéke a korábban definiált ***portType*** neve)

```
<binding name="WordOperationsBinding" type="tns:WordOperationsPortType">
   <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
   <operation name="hossz">
      <soap:operation style="rpc" soapAction="http://localhost/ws/php-soap/wsdl/soap-server.php/hossz"/>
      <input>
         <soap:body namespace="http://localhost/ws/php-soap/wsdl/" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
      </input>
      <output>
         <soap:body namespace="http://localhost/ws/php-soap/wsdl/" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
      </output>
   </operation>
   <operation name="forditott">
      <soap:operation style="rpc" soapAction="http://localhost/ws/php-soap/wsdl/soap-server.php/forditott"/>
      <input>
         <soap:body namespace="http://localhost/ws/php-soap/wsdl/" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
      </input>
      <output>
         <soap:body namespace="http://localhost/ws/php-soap/wsdl/" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
      </output>
   </operation>
   <operation name="reszszavak">
      <soap:operation style="rpc" soapAction="http://localhost/ws/php-soap/wsdl/soap-server.php/reszszavak"/>
      <input>
         <soap:body namespace="http://localhost/ws/php-soap/wsdl/" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
      </input>
      <output>
         <soap:body namespace="http://localhost/ws/php-soap/wsdl/" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
      </output>
   </operation>
</binding>
```

Ezután már csak a szolgáltatás elérését kell megadni a ***service*** szekcióban (a ***port binding*** a korábban definiált ***binding*** neve):

```
<service name="WordOperations">
   <port name="WordOperationsPort" binding="tns:WordOperationsBinding">
      <soap:address location="http://localhost/ws/php-soap/wsdl/soap-server.php"/>
   </port>
</service>
```

Miután befejeztük a definíciókat, le kell zárni a ***definitions*** xml elemet:

```
</definitions>
```

WEB-programozás II (3. előadás)
WEBSZOLGÁLTATÁSOK PHP-BEN

Webszolgáltatások megvalósítása a PHP SOAP könyvtárával:

Kliens: soap-client.php

```
<?php
  try {
        $client = new SoapClient("http://localhost/ws/php-soap/wsdl/words.wsdl");

        $szo = "teszt";

        $hossz = $client->hossz($szo);
        echo "Eredmény (hossz):<br>";
        var_dump($hossz);
        echo "<br>";

        $forditott = $client->forditott($szo);
        echo "Eredmény (forditott):<br>";
        var_dump($forditott);
        echo "<br>";

        $reszszavak = $client->reszszavak($szo);
        echo "Eredmény (reszszavak):<br>";
        var_dump($reszszavak);
        echo "<br>";

  } catch (SoapFault $e) {
        var_dump($e);
  }
?>
```

WEB-programozás II (3. előadás)

WEBSZOLGÁLTATÁSOK PHP-BEN

Webszolgáltatások megvalósítása a PHP SOAP könyvtárával:

c) Webszogáltatások WSDL-lel (a műveleteket egy osztály metódusai valósítják meg)

Példa helye: http://localhost/ws/php-soap/wsdl-class

Szerver: soap-server.php

```
<?php

Class Szavak {
    public function hossz($szo) {
      echo $szo;
      $eredmeny = strlen($szo);
      return $eredmeny;
    }

    public function forditott($szo) {
      $forditott = "";
      for($i=strlen($szo)-1; $i>=0; $i--)
         $forditott .= $szo[$i];
      $eredmeny = Array(
                          "eredeti"=>$szo,
                          "forditott"=>$forditott
                        );
      return $eredmeny;
    }
    public function reszszavak($szo) {
      $eredmeny = Array();
      for($i=0; $i<strlen($szo); $i++)
         for($j=1; $j<=strlen($szo)-$i; $j++)
            $eredmeny[ ] = substr($szo, $i, $j);
      return $eredmeny;
    }
}

$server = new SoapServer("words.wsdl");
$server->setClass('Szavak');
$server->handle();

?>
```

WEB-programozás II (3. előadás)

WEBSZOLGÁLTATÁSOK PHP-BEN

Webszolgáltatások megvalósítása a PHP SOAP könyvtárával:

A WSDL fájl ugyanaz lehet, mint az előző esetben, amikor a web-szolgáltatás műveleteit függvényekkel valósítottuk meg. Természeten a **http://localhost/ws/php-soap/wsdl** címet cseréljük ki a **http://localhost/ws/php-soap/wsdl-class** címre.

Még egy módosítást csináltunk a words.wsdl fájlban, a műveletek definíciójában (<operation> elemek):

soapAction="http://localhost/ws/php-soap/wsdl-class/soap-server.php/*művelet_név*" helyett a

soapAction="http://localhost/ws/php-soap/wsdl-class/soap-server.php/method=*művelet_név*"

megadást alkalmazzuk.

A végrehajtandó metódust megadása a művelet definíciójában:

soapAction="http://localhost/ws/php-soap/wsdl-class/soap-server.php/method=hossz"

soapAction="http://localhost/ws/php-soap/wsdl-class/soap-server.php/method=forditott"

soapAction="http://localhost/ws/php-soap/wsdl-class/soap-server.php/method=reszszavak"

A soap-client.php fájlban csak a wsdl elérési címét kell módosítani a http://localhost/ws/php-soap/wsdl-class/words.wsdl címre. A kliens számára a műveletek megvalósításának a módja érdektelen.

WEB-programozás II (3. előadás)

WEBSZOLGÁLTATÁSOK PHP-BEN

Webszolgáltatások megvalósítása a PHP SOAP könyvtárával:

d) Webszogáltatások a WSDL fájl generálásával

Példa helye: http://localhost/ws/php-soap/gen-wsdl

Léteznek eszközök a wsdl generálására a webszolgáltatást definiáló php osztályból. Ilyen eszköz a WSDLDocument php osztály (https://code.google.com/archive/p/wsdldocument/). A szolgáltatást definiáló osztály megfelelő kommentekkel kell ellátni. A **words.php** fájl:

```
<?php

class Szopar {
    /**
     * @var string
     */
    public $eredeti;
    /**
     * @var string
     */
    public $forditott;
}

class Szavak {
    /**
     *  @param string $szo
     *  @return integer
     */
    public function hossz($szo)
    {
        … // Itt a művelet megvalósítása
    }

    /**
     *  @param string $szo
     *  @return Szopar
     */
    public function forditott($szo) {
        … // Itt a művelet megvalósítása
    }

    /**
     *  @param string $szo
     *  @return string[ ]
     */
    public function reszszavak($szo) {
        … // Itt a művelet megvalósítása
    }
}

?>
```

WEB-programozás II (3. előadás)

WEBSZOLGÁLTATÁSOK PHP-BEN

Webszolgáltatások megvalósítása a PHP SOAP könyvtárával:

A wsdl fájl generálását végrehajtó szkript (gen-WSDLDocument.php):

```
<?php

error_reporting(0);

require 'words.php';
require './WSDLDocument/WSDLDocument.php';
$wsdl = new WSDLDocument('Szavak',
                                        "http://localhost/ws/php-soap/gen-wsdl/soap-server.php",
                                        "http://localhost/ws/php-soap/gen-wsdl/");
$wsdlfile = $wsdl->saveXML();
echo $wsdlfile;
file_put_contents ("words.wsdl" , $wsdlfile);

?>
```

A szkript létrehozza a words.wsdl fájlt abban a könyvtárban, ahol a szkript helyezkedik. A WSDLDocument konstruktorának a paraméterei:

1. A webszolgáltatást definiáló osztály neve (Szavak).
2. A soap szerver url-je.
3. A webszogáltatás névterének az url-je.

WEB-programozás II (3. előadás)
WEBSZOLGÁLTATÁSOK PHP-BEN

Webszolgáltatások megvalósítása a PHP SOAP könyvtárával:

A soap-server.php fájl:

```
<?php
require("words.php");
$server = new SoapServer("words.wsdl");
$server->setClass('Szavak');
$server->handle();
?>
```

A soap-client.php fájlban csak a wsdl elérési címét kell módosítani a http://localhost/ws/php-soap/gen-wsdl/words.wsdl címre.