



Politechnika Łódzka

Katedra Mikroelektroniki i Technik Informatycznych

Programowanie internetowe

Wykład 6 – Obiektowość w PHP, mechanizm sesji, baza MySQL i współpraca z poziomem PHP

mgr inż. Michał Wojtera

email: mwojtera@dmcs.pl



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Plan wykładu

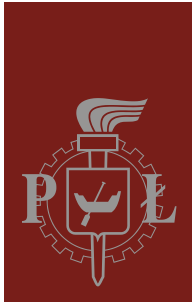
- ⇒ **Obiektywność w PHP**
 - ⇒ koncepcje programowania obiektowego (przypomnienie)
 - ⇒ klasy, właściwości, metody oraz różnice między wersją 4 i 5 PHP
 - ⇒ dziedziczenie
- ⇒ **Podstawy MySQL**
 - ⇒ podstawowe informacje związane z relacyjnym systemem zarządzania bazą danych MySQL
 - ⇒ dostęp z poziomu PHP
- ⇒ **Mechanizm sesji oraz cookies**
- ⇒ **Podsumowanie**



Katedra Mikroelektroniki i Technik Informatycznych

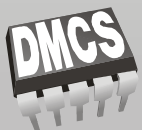
90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Obiektość w PHP

- ⇒ Koncepcje programowania obiektowego
 - ⇒ Obiektywny sposób tworzenia kodu, a nowoczesne języki programowania
 - ⇒ Oprogramowanie tworzone w sposób obiektowy:
 - ⇒ obiekty
 - ⇒ właściwości
 - ⇒ metody
 - ⇒ klasa
 - ⇒ hermetyzacja
 - ⇒ Polimorfizm
 - ⇒ Dziedziczenie



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Klasy, właściwości metody

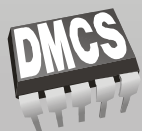
⇒ Tworzenie klas:

⇒ struktura klasy:

```
class classname  
{  
}
```

⇒ właściwości:

```
class classname  
{  
    var $attribut1;  
    var $attribute2;  
}
```





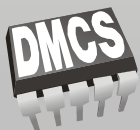
Klasy, właściwości metody

⇒ Tworzenie klas cd.:

⇒ metody:

```
class classname
{
    function operation1()
    {
    }
    function operation2($param1, $param2)
    {
    }
}
```

⇒ w PHP 4 dozwolone są tylko stałe inicjalizatory zmiennych *var*



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Klasy, właściwości metody

⇒ Tworzenie klas cd.:

⇒ konstruktory:

```
class classname
{
    function classname($param)
    {
        echo "Constructor called with
            parameter $param <br />";
    }
}
```

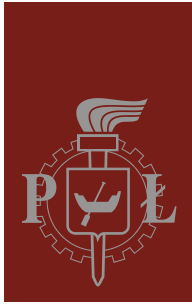
⇒ konstruktory nie zwracają żadnej wartości!

Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



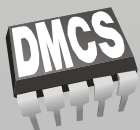


Klasy, właściwości metody

⇒ Tworzenie egzemplarza klasy

```
class classname
{
    function classname($param)
    {
        echo "Constructor called with parameter
            $param <br />";
    }
}

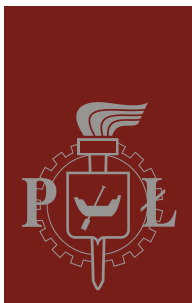
$a = new classname('First');
$b = new classname('Second');
$c = new classname();
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Klasy, właściwości metody

⇒ Korzystanie z właściwości klasy:

⇒ wskaźnik/zmienna *\$this*

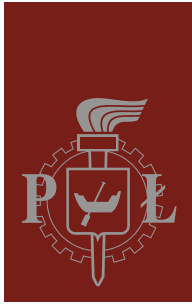
```
class classname
{
    var $attribute;
    function operation($param)
    {
        $this->attribute = $param
        echo $this->attribute;
    }
}
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl

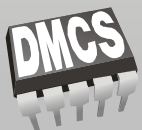


Klasy, właściwości metody

- ⇒ Dostęp do właściwości klasy:
 - ⇒ dostęp spoza klasy

```
class classname
{
    var $attribute;
}

$a = new classname();
$a->attribute = 'value';
echo $a->attribute;
```



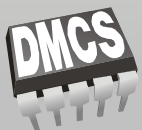


Klasy, właściwości metody

⇒ Dostęp do właściwości klasy:

⇒ funkcje udostępniające:

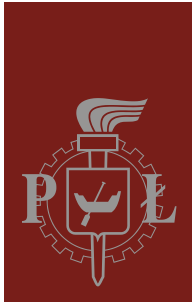
```
class classname
{
    var $attribute;
    function get_attribute()
    {
        return $this->attribute;
    }
    function set_attribute($new_value)
    {
        $this->attribute = $new_value;
    }
}
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

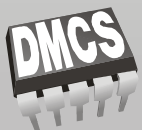
tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Klasy, właściwości metody

- ⇒ Dostęp do właściwości klasy:
 - ⇒ funkcje udostępniające - przykład:

```
function set_attribute($new_value)
{
    if( $new_value >= 0 && $new_value <= 100 )
        $this->attribute = $new_value;
}
```





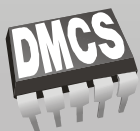
Klasy, właściwości metody

⇒ Wywoływanie metod:

```
class classname
{
    function operation1()
    {
    }
    function operation2($param1, $param2)
    {
    }
}

$a = new classname();

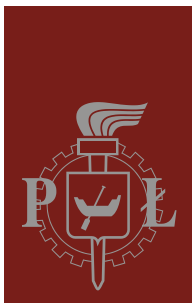
$a->operation1();
$a->operation2(12, 'test');
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Dziedziczenie

⇒ Implementacja dziedziczenia:

```
class B extends A
{
    var $attribute2;
    function operation2()
    {
    }
}

class A
{
    var $attribute1;
    function operation1()
    {
    }
}
```

```
$b = new B();
$b->operation1();
$b->attribute1 = 10;
$b->operation2();
$b->attribute2 = 10;
```

```
$a = new A();
$a->operation1();
$a->attribute1 = 10;
$a->operation2();
$a->attribute2 = 10;
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Dziedziczenie

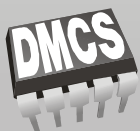
➔ Przesłanie:

```
class A
{
    var $attribute = 'default value';
    function operation()
    {
        echo 'Something<br />';
        echo "The value of \$attribute is \$this->attribute<br />";
    }
}

class B extends A
{
    var $attribute = 'different value';
    function operation()
    {
        echo 'Something else<br />';
        echo "The value of \$attribute is \$this->attribute<br />";
    }
}
```

```
$a = new A();
$a -> operation();

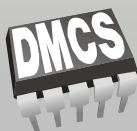
$b = new B();
$b -> operation();
```





Dziedziczenie

- ⇒ Konstruktory – UWAGA!
 - ⇒ PHP 4 nie wywołuje automatycznie konstruktora klasy bazowej z konstruktora klasy pochodnej. Należy zapewnić propagację wywołań konstruktorów klas nadrzędnych, jeśli to konieczne.
- ⇒ Dziedziczenie wielokrotne
- ⇒ Operator rozróżnienia zasięgu (::)
- ⇒ Nazwa specjalna *parent*





Dziedziczenie

⇒ *parent* – przykład:

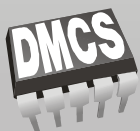
```
class A {
    function przyklad() {
        echo "Jestem A::przyklad() I dostarczam
                podstawową funkcjonalność.<br />";
    }
}

class B extends A {
    function przyklad() {
        echo "Jestem B::przyklad() i dostarczam
                dodatkową funkcjonalność.<br />";
        parent::przyklad();
    }
}

$b = new B;

// Wywoła to B::przyklad(), który z kolei wywoła
// A::przyklad().

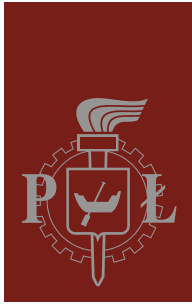
$b->przyklad();
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



OOP w PHP5

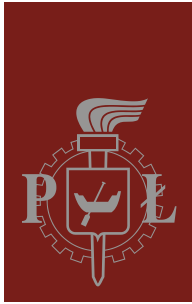
- ⇒ Moduł napisany od podstaw – znaczące różnice w odniesieniu do wersji 4
- ⇒ Lepsza wydajność i większa funkcjonalność
- ⇒ Tak jak w wersji 4, wartości domyślne właściwości muszą być stałymi (lub tablicami stałych)



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



OOP w PHP5

⇒ Automatyczne ładowanie obiektów – funkcja `__autoload()`:

```
<?php
    function __autoload($class_name) {
        require_once $class_name . '.php';
    }

    $obj  = new MyClass1();
    $obj2 = new MyClass2();
?>
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



OOP w PHP5

⇒ Konstruktor - `__construct()`

```
class BaseClass {  
    function __construct() {  
        print "In BaseClass constructor\n";  
    }  
}
```

```
class SubClass extends BaseClass {  
    function __construct() {  
        parent::__construct();  
        print "In SubClass constructor\n";  
    }  
}
```

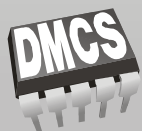
```
$obj = new BaseClass();
```

```
$obj = new SubClass();
```

Katedra Mikroelektroniki i Techniki Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl





OOP w PHP5

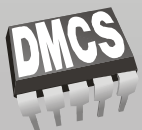
⇒ Destruktor - `__destruct()`

```
<?php
    class MyDestructableClass {
        function __construct() {
            print "In constructor\n";
            $this->name = "MyDestructableClass";
        }

        function __destruct() {
            print "Destroying " . $this->name . "\n";
        }
    }

    $obj = new MyDestructableClass();

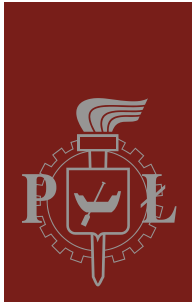
?>
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

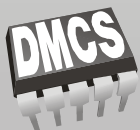
tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



OOP w PHP5

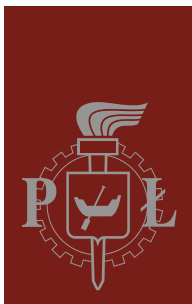
⇒ Hermetyzacja – specyfikatory dostępu: *public*, *protected*, *private*

```
class MyClass
{
    public $public = 'Public';
    protected $protected = 'Protected';
    private $private = 'Private';
    function printHello()
    {
        echo $this->public;
        echo $this->protected;
        echo $this->private;
    }
}
$obj = new MyClass();
echo $obj->public; // Works
echo $obj->protected; // Fatal Error
echo $obj->private; // Fatal Error
$obj->printHello(); // Shows Public, Protected and Private
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3
tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



OOP w PHP5

⇒ właściwości i metody statyczne – słowo *static*

⇒ stałe:

```
<?php
class MyClass
{
    const constant = 'constant value';

    function showConstant() {
        echo self::constant . "\n";
    }
}

echo MyClass::constant . "\n";

$class = new MyClass();
$class->showConstant();
// echo $class::constant; is not allowed
```

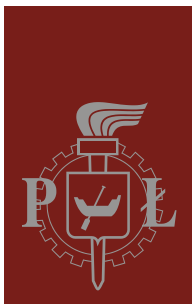
?>

Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



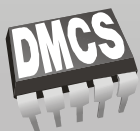


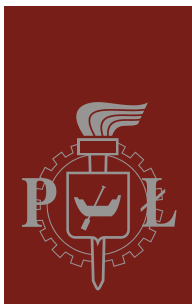
Czym jest MySQL?

- ➔ Czym jest baza danych?
 - ➔ uporządkowany zbiór danych, zazwyczaj przechowywany w jednym lub wielu skojarzonych ze sobą plikach
 - ➔ czasem mylnie używane pojęcie *baza danych*
- ➔ Co to jest relacyjna baza danych?
 - ➔ zakładamy, iż dane są przechowywane w tabelach, dla których możliwe są odniesienia między ich polami; istnienie tych relacji między tabelami pozwala wprowadzić pojęcie relacyjnej bazy danych
 - ➔ przykłady: Oracle, MySQL, IBM DB2, Microsoft SQL Server
 - ➔ inne modele
- ➔ System zarządzania relacyjną bazą danych

Katedra Mikroelektroniki i Technik Informatycznych

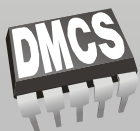
90-924 Łódź, al. Politechniki 11, budynek C3
tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl





Pojęcia

- ⇒ Tabele
- ⇒ Rekordy
- ⇒ Pola
- ⇒ Zapytania
- ⇒ SQL
- ⇒ Indeksy
- ⇒ Klucze



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



MySQL

⇒ Cechy cd.:

- ⇒ replikacja
- ⇒ transakcje
- ⇒ ograniczenia kluczy obcych
- ⇒ funkcje GIS
- ⇒ liczne języki programowania współpracującego
- ⇒ ODBC
- ⇒ niezależność sprzętowa
- ⇒ szybkość

⇒ Aktualna wersja stabilna:

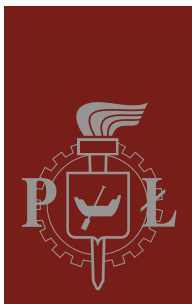
⇒ 5.0

Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl





Podstawy pracy z MySQL

⇒ Korzystanie z Monitora MySQL:

- ⇒ uruchamiany poleceniem *mysql*;
- ⇒ polecenia monitora (*;*, *quit*);
- ⇒ logowanie do monitora

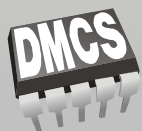
```
> mysql -h hostname -u username -p
```

⇒ Tworzenie bazy danych:

- ⇒ polecenie `mysql> create database dbname;`

⇒ Użytkownicy i uprawnienia:

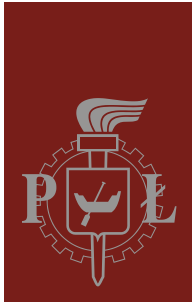
- ⇒ konta użytkowników;
- ⇒ system uprawnień – zasada minimalnych uprawnień



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Podstawy pracy z MySQL

⇒ przydzielanie uprawnień – polecenie GRANT:

⇒ służy do tworzenia nowego użytkownika i nadawania mu uprawnień

⇒ przykład:

```
GRANT privileges [columns]
```

```
ON item
```

```
TO user_name [IDENTIFIED BY 'pass']
```

```
[WITH GRANT OPTION]
```

⇒ item: *dbname.tablename*, wszystko – ‘*’

⇒ opcja WITH GRANT OPTION



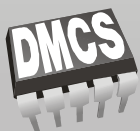


Podstawy pracy z MySQL

- ➔ uprawnienia:
 - ➔ użytkowników: **SELECT, INSERT, UPDATE, DELETE, INDEX, ALTER, CREATE, DROP;**
 - ➔ specjalne: **ALL, USAGE**
- ➔ przykłady:

```
mysql> GRANT all
-> ON *
-> TO fred identified by 'abc123'
-> WITH GRANT OPTION;
```

```
mysql> REVOKE alter, create, drop
-> ON dbname
-> FROM fred;
```





Podstawy pracy z MySQL

- ⇒ korzystanie z właściwej bazy - polecenia:

```
mysql > use dbname;  
> mysql dbname -h hostname -u username -p
```

- ⇒ tworzenie tabel bazy danych:

```
CREATE TABLE tablename (columns);
```

- ⇒ wykonanie polecenia SQL z pliku:

```
> mysql -h host -u username dbname -p <file.sql
```

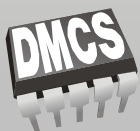
- ⇒ przykład:

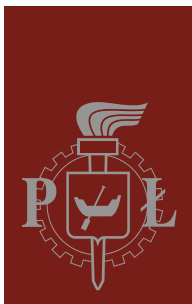
```
CREATE TABLE customers  
( customerid int UNSIGNED NOT NULL AUTO_INCREMENT  
                                     PRIMARY KEY,  
  name char(30) NOT NULL,  
  address char(40) NOT NULL,
```

Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl

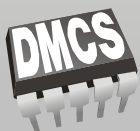




Podstawy pracy z MySQL

- ⇒ Słowa kluczowe: `NOT NULL`, `AUTO_INCREMENT`, `PRIMARY KEY`, `UNSIGNED`;
- ⇒ przeglądanie struktury bazy danych: polecenia `SHOW` i `DESCRIBE`;

```
mysql> SHOW tables;  
mysql> SHOW databases;  
mysql> DESCRIBE tablename;
```
- ⇒ identyfikatory MySQL;





Podstawy pracy z MySQL

⇒ typy danych:

źródło: www.w3schools.com

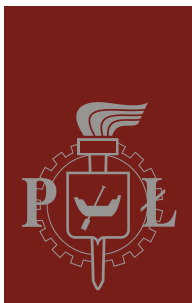
Numeric Data Types	Description
int(size) smallint(size) tinyint(size) mediumint(size) bigint(size)	Hold integers only. The maximum number of digits can be specified in the size parameter
decimal(size,d) double(size,d) float(size,d)	Hold numbers with fractions. The maximum number of digits can be specified in the size parameter. The maximum number of digits to the right of the decimal is specified in the d parameter



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Podstawy pracy z MySQL

⇒ typy danych:

źródło: www.w3schools.com

Textual Data Types	Description
char(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis
varchar(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis
tinytext	Holds a variable string with a maximum length of 255 characters
text blob	Holds a variable string with a maximum length of 65535 characters
mediumtext mediumblob	Holds a variable string with a maximum length of 16777215 characters
longtext longblob	Holds a variable string with a maximum length of 4294967295 characters



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Podstawy pracy z MySQL

⇒ typy danych:

źródło: www.w3schools.com

Date Data Types	Description
<code>date(yyyy-mm-dd)</code> <code>datetime(yyyy-mm-dd hh:mm:ss)</code> <code>timestamp(yyyymmddhhmmss)</code> <code>time(hh:mm:ss)</code>	Holds date and/or time

Misc. Data Types	Description
<code>enum(value1,value2,ect)</code>	ENUM is short for ENUMERATED list. Can store one of up to 65535 values listed within the () brackets. If a value is inserted that is not in the list, a blank value will be inserted
<code>set</code>	SET is similar to ENUM. However, SET can have up to 64 list items and can store more than one choice



Katedra Mikroelektroniki i Technik Informatycznych

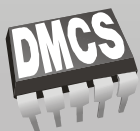
90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Praca z MySQL

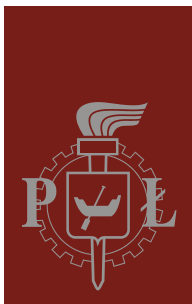
- ⇒ Co to jest SQL?
- ⇒ Podstawowe polecenia:
 - ⇒ **CREATE ;**
 - ⇒ **ALTER ;**
 - ⇒ **SELECT ;**
 - ⇒ **DELETE ;**
 - ⇒ **INSERT INTO *tablename* VALUES**
 - ⇒ **UPDATE**
 - ⇒ **DROP**



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Praca z MySQL

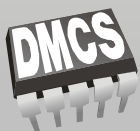
⇒ Wstawianie danych do bazy:

```
INSERT [INTO] table [(column1, column2,  
column3,...)] VALUES (value1, value2, value3,...);
```

```
insert into klienci values  
(NULL, „Nowak”, „ul. Nieznana 10”, „Łódź”);
```

```
insert into klienci (nazwisko, miasto) values  
(„Nowak”, „Łódź”);
```

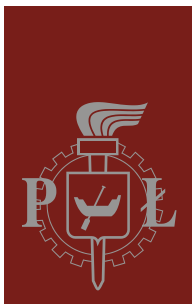
```
insert into klienci  
set nazwisko=„Nowak”,  
adres=„Nieznana 10”,  
miasto=„Łódź”;
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl

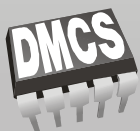


Praca z MySQL

⇒ Pobieranie informacji z bazy:

```
SELECT items
FROM tables
[ WHERE condition ]
[ GROUP BY group_type ]
[ HAVING where_definition ]
[ ORDER BY order_type ]
[ LIMIT limit_criteria ] ;
```

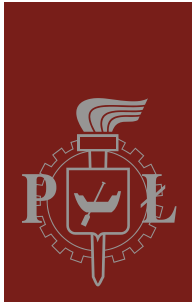
```
select *
from order_items;
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



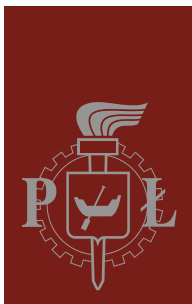
Praca z MySQL

- ⇒ Pobieranie danych spełniających określone kryteria:

```
select *  
from zamowienia  
where idklienta = 3 or idklienta = 4;
```

- ⇒ Operatory porównania używane w klauzuli **WHERE**:
 - ⇒ =, >, <, >=, <=, != lub <>, IS NOT NULL, IS NULL, BETWEEN, IN, NOT IN, LIKE, NOT LIKE;
 - ⇒ łączenie kryteriów: **and**, **or**.

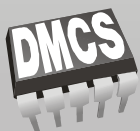




Praca z MySQL

➔ Przykłady:

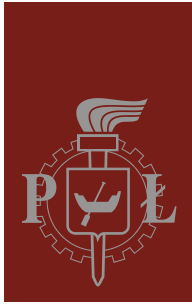
Operator	Przykład
IS NOT NULL	Address is not null
IS NULL	Address is null
BETWEEN	Amount between 0 and 10
IN	City in („warsaw”, „lodz”)
NOT IN	City not in („warsaw”, „lodz”)
LIKE	Name like („fred%”)
NOT LIKE	Name not like („fred%”)
REGEXP	Name regexp



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

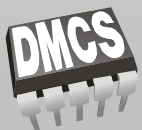
tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Praca z MySQL

⇒ Proste złączenie:

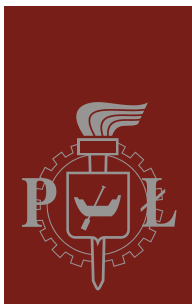
```
SELECT zamowienia.idzamowienia,  
zamowienia.kwota, zamowienia.data  
FROM klienci, zamowienia  
WHERE klienci.nazwisko = 'Jan Kowalski'  
and klienci.idklienta = zamowienia.idklienta;
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl

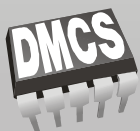


Dostęp z poziomu PHP

⇒ Łączenie się z bazą danych:

```
mysql_connect(servername,username,password);
```

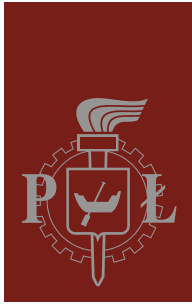
```
<?php
    $con = mysql_connect("localhost",
                        "peter","abc123");
    if (!$con)
        die('Could not connect: ' . mysql_error());
    // some code
?>
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

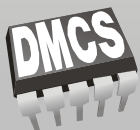
tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Dostęp z poziomu PHP

⇒ Wykonywanie zapytania:

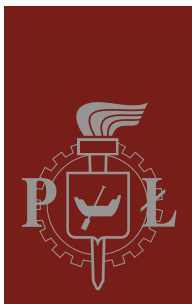
```
<?php
    $con = mysql_connect("localhost","peter","abc123");
    if (!$con)
        {
            die('Could not connect: ' . mysql_error());
        }
    if (mysql_query("CREATE DATABASE my_db", $con))
        {
            echo "Database created";
        }
    else
        {
            echo "Error creating database: " . mysql_error();
        }
    mysql_close($con);
?>
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



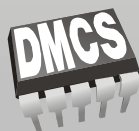
Dostęp z poziomu PHP

⇒ Wstawienie danych do bazy:

```
<?php
    $con = mysql_connect("localhost","peter","abc123");
    if (!$con)
        {
            die('Could not connect: ' . mysql_error());
        }

    mysql_select_db("my_db", $con);
    mysql_query("INSERT INTO person (FirstName,
    LastName, Age) VALUES ('Peter', 'Griffin', '35')");
    mysql_query("INSERT INTO person (FirstName,
    LastName, Age) VALUES ('Glenn', 'Quagmire', '33')");

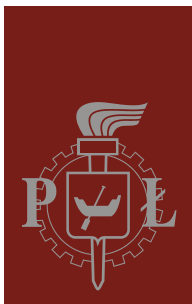
    mysql_close($con);
?>
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl

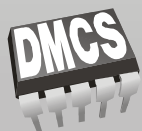


Dostęp z poziomu PHP

⇒ Pobieranie wiersza danych z wyniku zapytania:

- ⇒ `mysql_fetch_row($zmienna_wynikow);`
- ⇒ `mysql_fetch_array($zmienna_wynikow);`

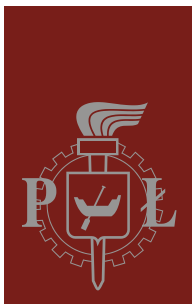
```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
    { die('Could not connect: ' . mysql_error
()); } mysql_select_db("my_db", $con);
$result = mysql_query("SELECT * FROM person");
while ($row = mysql_fetch_array($result))
{
    echo $row['FirstName'] . " " . $row['LastName'];
    echo "<br />";
}
mysql_close($con);
?>
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

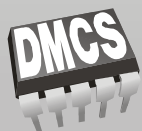
tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Dostęp z poziomu PHP

- ➔ Wyświetlenie pobranych danych w formie tabeli HTML:

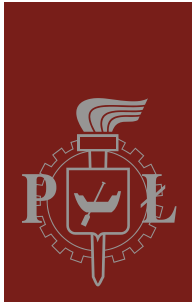
```
<?php
    $con = mysql_connect("localhost","peter","abc123");
    if(!$con)
        { die('Could not connect: ' . mysql_error
    ()); }
    mysql_select_db("my_db", $con);
    $result = mysql_query("SELECT * FROM person");
    echo "<table border='1'> <tr> <th>Firstname</th>
        <th>Lastname</th> </tr>";
    while($row = mysql_fetch_array($result))
        {
        echo "<tr>";
        echo "<td>" . $row['FirstName'] . "</td>";
        echo "<td>" . $row['LastName'] . "</td>";
        echo "</tr>";
        }
    echo "</table>";
    mysql_close($con);
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Cookies

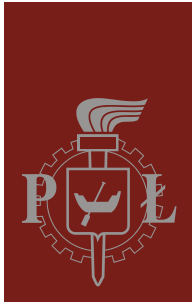
- ⇒ HTTP – stateless protocol
- ⇒ Wynalazek Netscape
- ⇒ Pierwsze zastosowania
- ⇒ Rozwój cookies
- ⇒ Przechowywanie identyfikatora



Katedra Mikroelektroniki i Technik Informatycznych

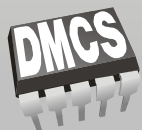
90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Cookies i sesje

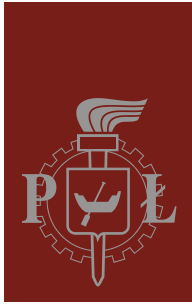
- ⇒ Oba mechanizmy dostępne dla programisty PHP
- ⇒ Oba mogą spełniać zadanie przechowywania danych pomiędzy stronami w naszym serwisie
- ⇒ Różnice:
 - ⇒ Cookies mogą być ustawiane na bardzo długi czas
 - ⇒ Miejsce przechowywania danych sesji i cookies
 - ⇒ Można manipulować przy cookies po stronie klienta używając JavaScript
 - ⇒ Sesje mogą przechowywać duże ilości danych
 - ⇒ Dane sesji nie muszą być przesyłane z każdym wywołaniem podstrony



Katedra Mikroelektroniki i Technik Informatycznych

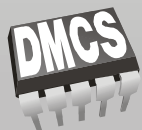
90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Cookies i sesje

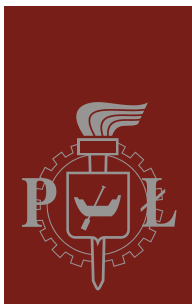
- ⇒ Sesje używają cookies do przechowywania informacji o identyfikatorze sesji
- ⇒ Jeżeli mamy do dyspozycji tylko mechanizm sesji – identyfikator przesyłany przy użyciu URL
- ⇒ Jeżeli używamy cookie do przechowywania id, to automatycznie wygasa w momencie zamknięcia okna przeglądarki
- ⇒ Najlepsze rozwiązanie – połączenie obu mechanizmów:
 - ⇒ dane przechowywać w bazie
 - ⇒ używać cookie do przechowywania numeru ID by dostać się do naszych danych w bazie



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Cookies

- ⇒ Tworzenie ciasteczka – funkcja `setcookie()`:
 - ⇒ musi być wywołana przed przesłaniem danych formularza
 - ⇒ HTTP przesyła informacje nagłówka przed ciałem
 - ⇒ Cookies są przesyłane w nagłówku – kiedy błąd?
 - ⇒ Dwa rozwiązania:
 - ⇒ umieszczenie cookies na górze strony
 - ⇒ użycie buforowania wyjścia
- ⇒ Przykład wywołania:
 - ⇒ `setcookie("Name", $_POST['Name'], time() + 31536000) ;`



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



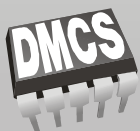
Cookies

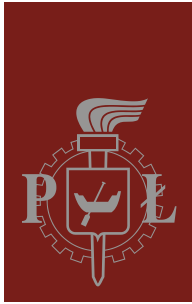
- ⇒ Parametry:
 - ⇒ Nazwa
 - ⇒ Wartość
 - ⇒ Czas wygaśnięcia
 - ⇒ Ścieżka – katalog, w którym cookie jest aktywne (domyślnie /)
 - ⇒ Domena
 - ⇒ Secure

```
bool setcookie ( string name [, string value [,  
                int expire [, string path [, string domain [,  
                bool secure [, bool httponly]]]] ] )
```

Katedra Mikroelektroniki i Technik Informatycznych

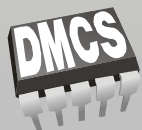
90-924 Łódź, al. Politechniki 11, budynek C3
tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl

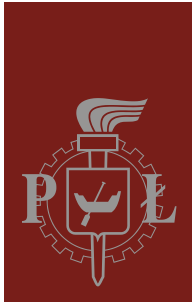




Cookies

- ⇒ Jeżeli cookie zostało ustawione, możemy dostać się do niego na następnej stronie przy użyciu `$_COOKIE`
- ⇒ Pułapki:
 - ⇒ Cookie nie będzie widoczne przed załadowaniem strony, dla której ma być widoczne (`print_r($_COOKIE);`)
 - ⇒ Cookie muszą być usuwane z takimi parametrami z jakimi zostały utworzone. Jeżeli parametr wartość jest pustym łańcuchem lub `FALSE`, a wszystkie parametry takie jak dla ustawienia cookie, wtedy cookie z określoną nazwą będzie usunięte z komputera klienta





Cookies

⇒ Przykłady:

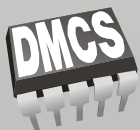
⇒ ustawienie cookie:

```
$value = 'something from somewhere';
```

```
setcookie("TestCookie", $value);  
setcookie("TestCookie", $value, time()+3600);  
/* expire in 1 hour */  
setcookie("TestCookie", $value, time()+3600,  
          "/~rasmus/", ".example.com", 1);
```

⇒ usunięcie cookie:

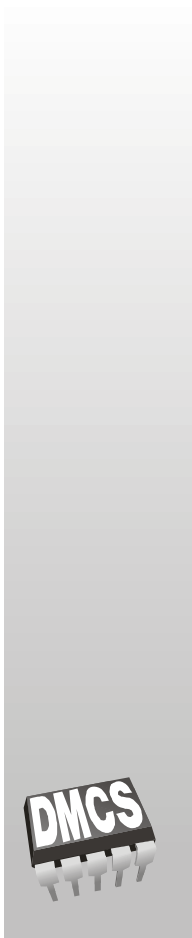
```
// set the expiration date to one hour ago  
setcookie ("TestCookie", "", time() - 3600);  
setcookie ("TestCookie", "", time() - 3600,  
          "/~rasmus/", ".example.com", 1);
```





Sesja

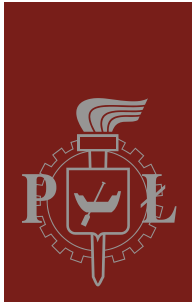
⇒ Czym jest sesja?



Katedra Mikroelektroniki i Technik Informatycznych

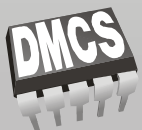
90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



Sesja

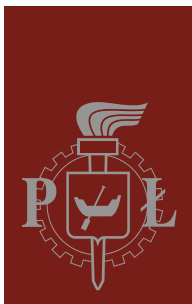
- ⇒ Mechanizm, który daje możliwość zachowania pewnych danych podczas następujących po sobie wywołań strony
- ⇒ Użytkownik otrzymuje unikalny identyfikator, tzw. id sesji
- ⇒ Identyfikator jest przechowywany albo jako ciasteczko po stronie klienta lub propagowany jest w url
- ⇒ Sesja jest kombinacją informacji przechowywanych w pliku po stronie serwera i cookie po stronie klienta zawierającego referencję do danych na serwerze



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



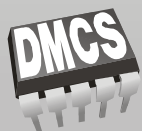
Sesja

- ⇒ Tworzenie nowej sesji:
 - ⇒ jawnie - `session_start()`
 - ⇒ niejawnie – `session_register()` (jeżeli `register_globals = on`)
- ⇒ Tablica superglobalna `$_SESSION`
- ⇒ Ustawianie zmiennych sesji:

```
$_SESSION['var'] = $val;  
$_SESSION['FirstName'] = "Jim";
```

 - ⇒ dawniej `session_register()`
- ⇒ Czytanie zmiennych sesji

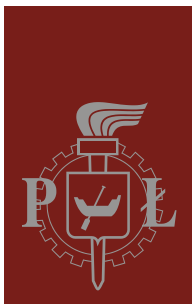
```
$_SESSION['foo'] = 'bar';  
print $_SESSION['foo'];
```



Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl



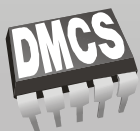
Sesja

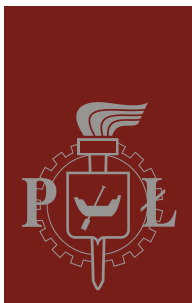
⇒ Usuwanie informacji z sesji

```
$_SESSION['foo'] = 'bar';  
print $_SESSION['foo'];  
unset($_SESSION['foo']);
```

⇒ Kończenie sesji:

```
session_start( );  
$_SESSION = array( );  
session_destroy( );
```





Sesja

⇒ Sprawdzanie informacji sesji

```
session_start( );  
if (isset($_SESSION['FirstName'])) {  
    /// your code here  
}
```

⇒ Sesja umieszcza swoje dane w katalogu */tmp* domyślnie, więc trzeba mieć to na uwadze ze względów bezpieczeństwa. Można zmienić katalog na inny lub użyć bazy danych do przechowywania danych





Sesja

- ⇒ Autentykacja przy użyciu sesji i danych z bazy:
 - ⇒ skrypt logowania używający formularza do pobrania danych użytkownika
 - ⇒ porównanie wprowadzonych danych z tymi zawartymi w bazie, w tabeli *users*
 - ⇒ ustawienie zmiennych sesji w taki sposób, by posłużyły do autentykacji sesji
 - ⇒ funkcja zabezpieczająca każdą stronę która wymaga uwierzytelnienia
 - ⇒ funkcja do realizacji wylogowywania, niszcząca sesję
 - ⇒ zmiana hasła
 - ⇒ rejestrowanie IP -> `$_SERVER [„REMOTE_ADDR”]`

Katedra Mikroelektroniki i Technik Informatycznych

90-924 Łódź, al. Politechniki 11, budynek C3

tel. 042 631 26 45, fax 042 042 636 03 27, e-mail: secretary@dmcs.p.lodz.pl

