OOP intro

уеб разработка с PHP2020/2021

Милена Томова
Vratsa Software
https://vratsasoftware.com/

Table of Contents


the essence of OOP

the essence of OOP

Web application, developed procedural way -

- info/data flow from a variable to a variable
- info/data flow from a function to function
- large pieces of code - hard to read and understand
- same pieces of code are repeated many times

the essence of OOP

Web application, developed OOP way -

- Different objects /users, customers, pages etc./ interact with each other
- Every objects has a pack of tasks and abilities in the App.
- Every object`s tasks and abilities are clearly predefined -
  - compare the users and pages in a Web App
  - compare the admin user, the customer user, the seller user

what are the tasks and abilities they have in a Web App?

the essence of OOP

Web app - the OOP way -

- Code reusability -
  - the admin user and the customer user both have **same** and **different** tasks and abilities in a Web App

    name the same and the different admin`s and customer`s tasks ...

- Modularity and extensibility -
  - The Code is separated in parts /modules/ and can be easily extended
- Encapsulation -
  - Developers use class methods, not interested in class methods definitions - the logic flow, just the result of the methods.

the essence of OOP

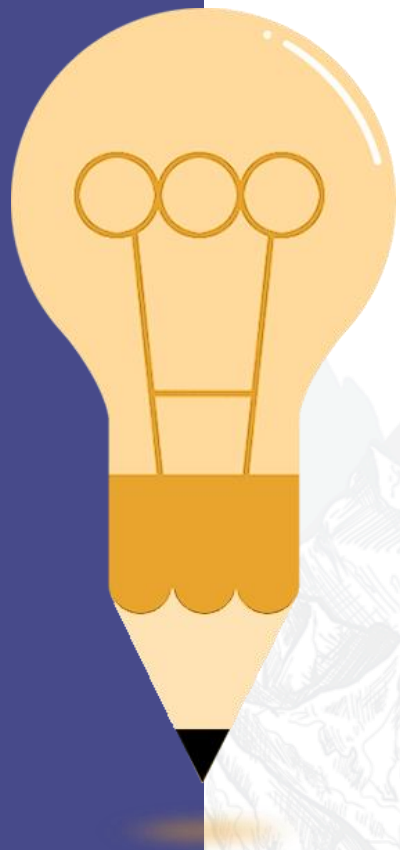
New functionalities are added without need to modify the existing ones.

After a Web App was built and even set in production we can add new features -

- *registration*
- *pages with access only after registration and login*
- *add new user types with new abilities /pages accessible only for this type of users/*
- *etc etc*

name such scenarios ...

to summarize

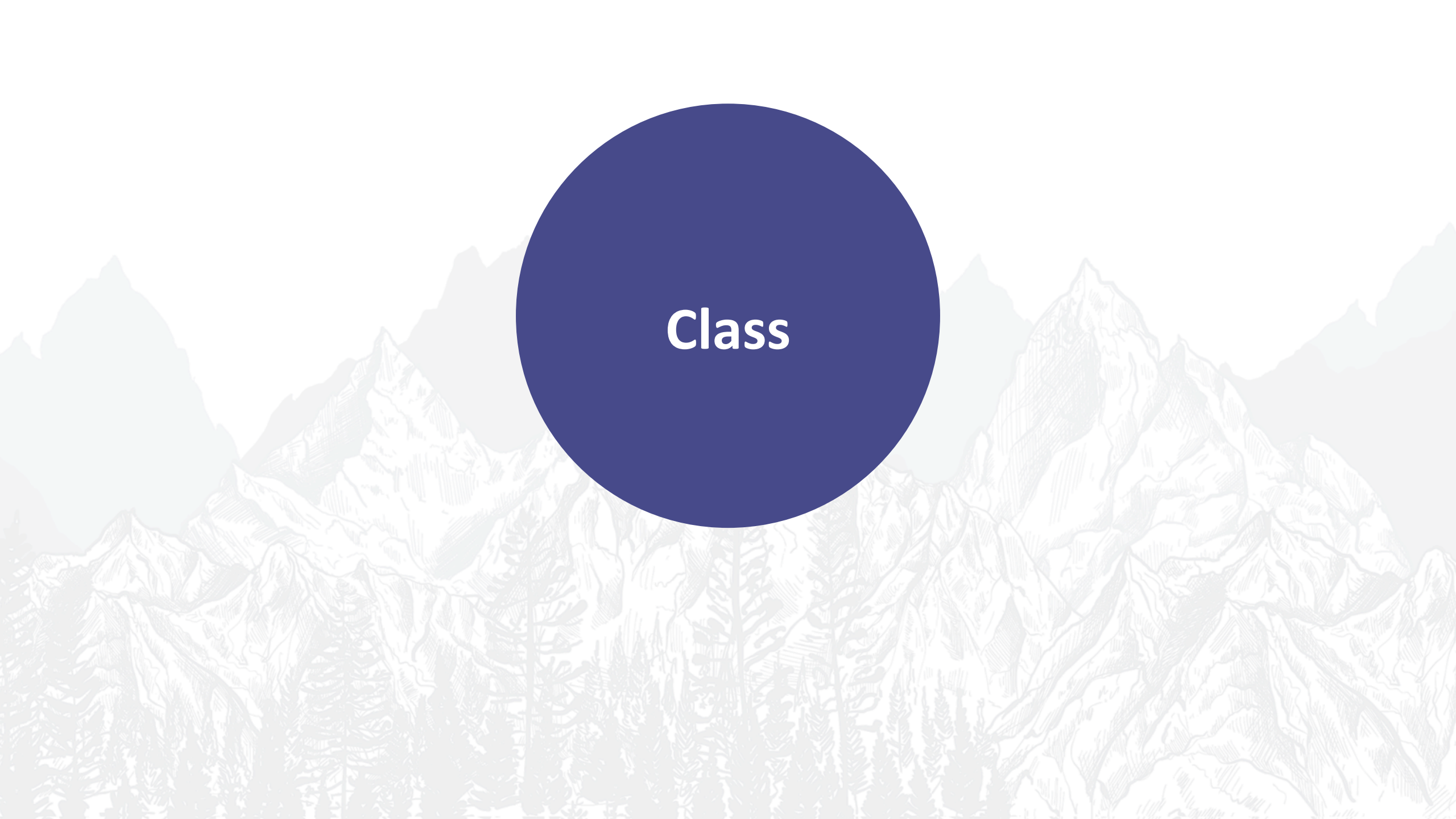


procedural programming

- small not complicated projects

OOP

- larger projects
- the project can be separated into parts
- every project`s part can be developed by different developers team
- every team works on its own - not interested in how the other team’s parts have been developed and what is the logic flow


Class

Class

The class is a way to describe an object.

What features and functionalities an object has that make us call it a car? A customer? A web page?

Class

The class is a blueprint -

a template that holds the object features /properties/

and abilities /methods/, without specifying the properties values!

Only the features and abilities that make us name that object an user, a car, a page ….

Class

Define

class Car

class User

class Page

Object

Object

After the blueprint - the class is defined

```
class Automobile {

}

class User {

}
```

Object

We can start producing - instantiating cars, users, pages.

Cars are of different colors, models etc.

Users have different usernames, passwords etc.

The properties` values can be different but the blueprint - the class stays unchanged.

Object

Using the class /the blueprint/ we can create objects of that class.

The objects have values for the properties defined in the class -

The first user object has an username ‘Pesho’ and a password ‘Secret’

The second user object has an username ‘Gosho’ and a password ‘MoreSecret’

etc.

Object

The first page object has a title ‘Home page’ and a headline ‘This is a home page ….’

The second page object has a title ‘Profile page’ and a headline ‘Here you can see and edit your personal data’

Object

Врача Софтуер Общество

```
class Car {
   public $color;
   public $model;
   public $doors = 4;
}

$red_car = new Car();
$red_car->color = ‘red’;
$red_car->model = ‘red car model’;
```

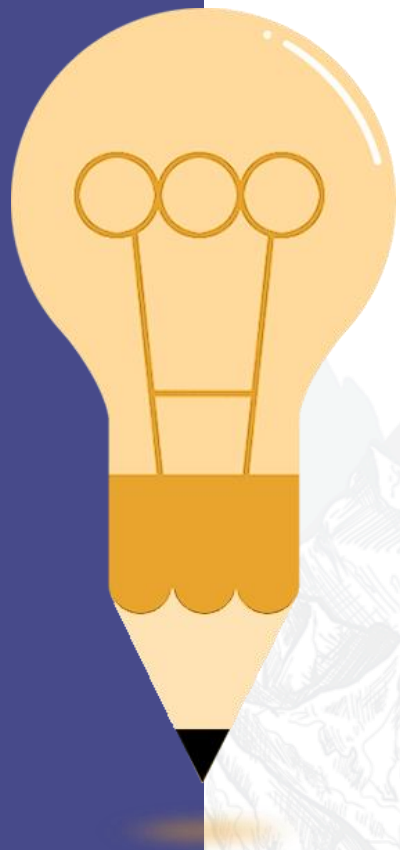
Object

Враца Софтуер Общество

```
class Car {
   public $color;
   public $model;
   public $doors = 4;
}

$green_car = new Car();
$green_car->color = ‘green’;
$green_car->model = ‘green car model’;
$green_car->doors = 2;
```

Good practice



Every class definition is saved in a file with a name as the class name.

```
class Car {
    public $color;
    public $model;
    public $doors = 4;
}
```

is stored in Car.php

Properties

Properties

Врача Софтуер Общество

The class properties are variables that can have default values. Their values can be set and/or change.

```
class Car {
    public $color;
    public $model;
    public $doors = 4;
}
```

public is an access specifier

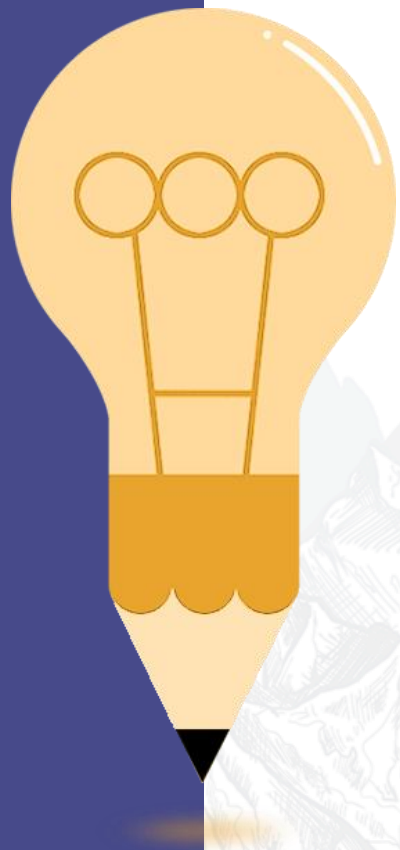
Properties

Врaца Софтуер Общество

How to access object properties

```
$green_car = new Car();
…
echo $green_car->doors;//4
$green_car->doors = 2;
echo $green_car->doors;//2
```

Be careful!



The object properties values in real projects are not modified directly.

Here we modify their values directly only for practice reason!

Properties

Define Page class and User class.

Define three properties per class.

Set default value for one property per class.

Instantiate two objects per class.

Set values for each property.

Print object properties.

Change object properties values.

Methods

Methods

The class methods are functions defined within the class.

They answer to the question - What objects of this class can/will do?

The are defined with access modifier also.

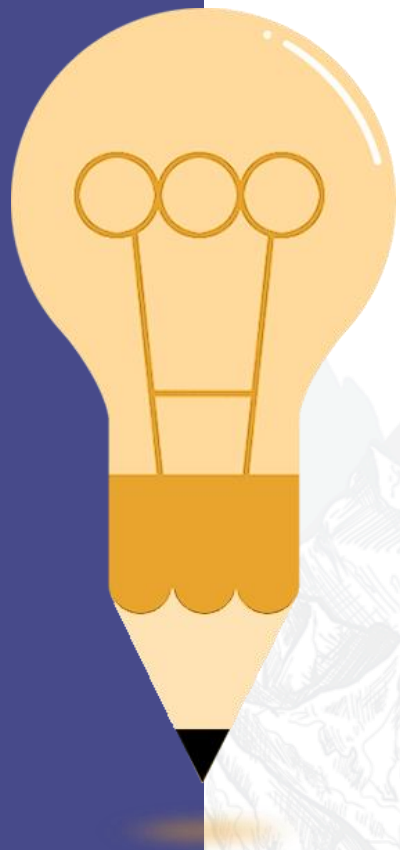
Methods

Враца Софтуер Общество

```
class Car {
      //properties
      public $color;
      ...

      public function set_car_color( $color ){
      $this->color = $color;
    }
      public function print_car_color(){
      echo  $this->color;
    }
}
```

$this



Within the class, $this keyword points to the object`s

- property
- method

It gives us possibility to access the class properties and methods in the blueprint of the class /in designing what feature the objects of that class have and what they can do/

Methods

Враца Софтуер Общество

```
$red_car = new Car();
$red_car->set_car_color('red');
$red_car->print_car_color();//red
```

Access specifiers

Access specifiers

The access specifiers limit the access to the class properties and methods outside of the class.

When using them

- on a class instances - the objects
- on classes that inherit the current class
- on classes that use current class properties or methods

Access specifiers

The access specifiers can be

public
private
protected

Access specifiers

Враца Софтуер Общество

```
class Specifier {
    public $a = ‘A specifier’;
    private $b = ‘B specifier’;
    protected $c = = ‘C specifier’;
}

$test_spec = new Specifier();

echo $test_spec->a;
echo $test_spec->b;
echo $test_spec->c;
```

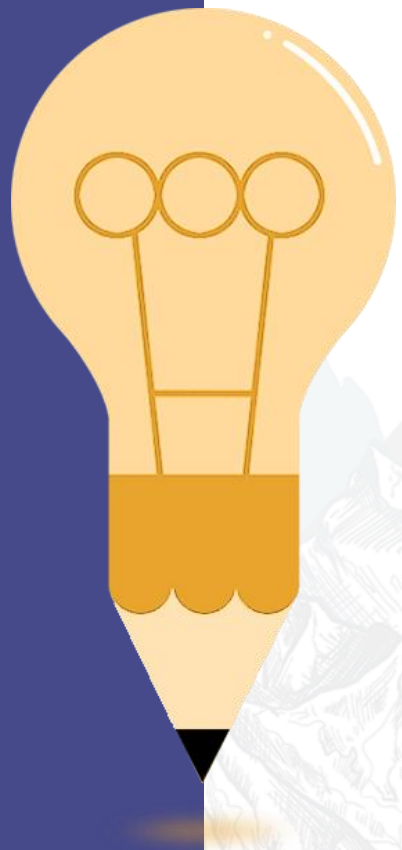
Access specifiers

Define public, private and protected methods in a class.
Instantiate an object of that class.
Call the three methods on that object.
Read the errors if any.

Use class properties with the three types of access modifiers in class methods.

Call that methods on a class object.
Read the errors if any.

access specifiers



They are used to avoid accidental or intended method definition or property value change.

By using access specifiers we “hide” /encapsulate/ the class definitions /implementation/.

We instantiate objects and can call their methods, use their properties without need to know how these methods are implemented.

конструктор
__construct()

__construct()

A method that is called and executed **always**
when a new object is instantiated.

We are not obliged to have a __construct() method in each class definition

Access specifiers

Враца Софтуер Общество

```
class Page {
      public function __construct(){
      echo ‘A new page has been created!’;
   }
}

$home_page = new Page();
      //A new page has been created!
```

Access specifiers

Врача Софтуер Общество

We can use the constructor to set initial properties values.

```
class Page {
	public $title;
	public function __construct( $title ){
		$this->title = $title;
	echo “A new page - {$this->title} has been created!”;
    }
}
$home_page = new Page(‘Home’);
	//A new page - Home has been created!
```

Questions?


Враца Софтуер Общество
Гнездото Coworking
Цялостен курс по програми ране
Дизайн курс
Курс по дигит. маркети нг
MindHub
CodeWeek Враца
HACK VRATSA
#кодаправиш

Partners

Враца Софтуер Общество

AMERICA FOR BULGARIA FOUNDATION

ВРАЦА

Telerik Academy

НАЦИОНАЛНА МРЕЖА ЗА ДЕЦАТА

MindHub

ПРОМЯНАТА

Trainings @ Vratsa Software

- Vratsa Software – High-Quality Education, Profession and Jobs
  - www.vratsasoftware.com
- The Nest Coworking
  - www.nest.bg
- Vratsa Software @ Facebook
  - www.fb.com/VratsaSoftware
- Slack Channel
  - www.vso.slack.com