

Обучение «Применение паттернов и структура модулей»

Докладчик: Иванов Н.Ю.

Проектный отдел: 3

Должность: Team Lead

План

- Вводная по паттернам;
- Структура модулей и как не создать спагетти код;
- Что такое DI и как его можно использовать в bitrix
- Автоматическая проверка вашего кода

Вводная по паттернам и структуре модулей

Курс на youtube



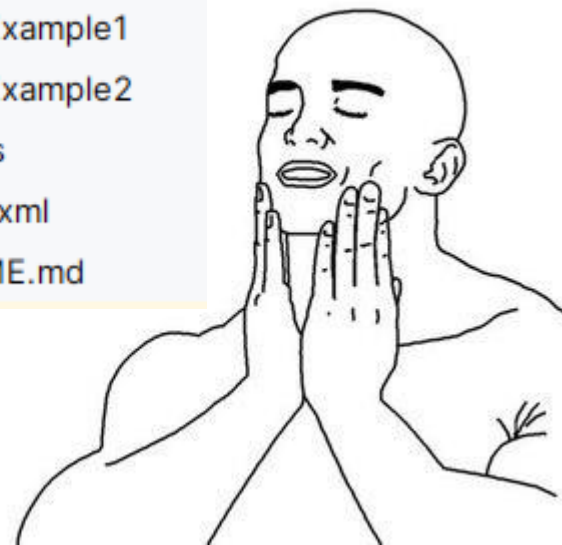
Сайт с паттернами



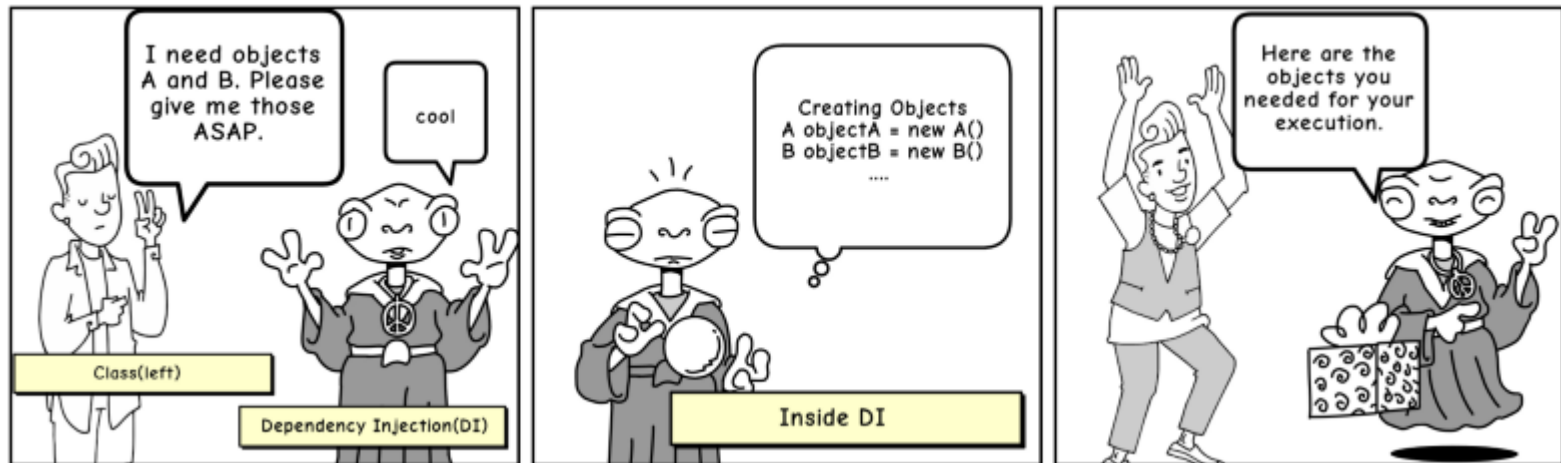
Структура модуля



```
> install
> lang
▼ lib
  Agents
  Controllers
  Events
  ▼ Services
    Example1
    Example2
  Utils
  phpcs.xml
  README.md
```



DI - Dependency injection



DI в bitrix



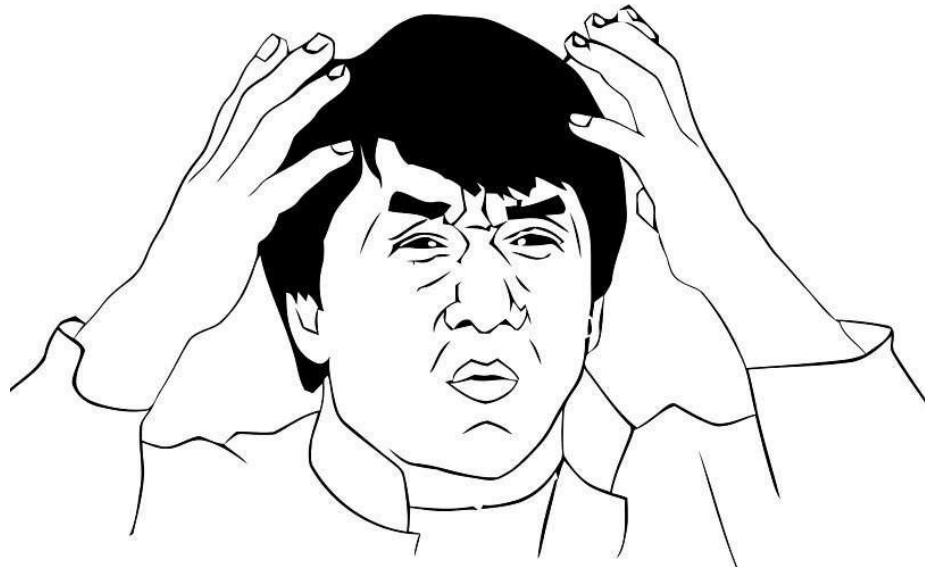
```
$serviceLocator = \Bitrix\Main\DI\ServiceLocator::getInstance();  
if ($serviceLocator->has('someService'))  
{  
    $someService = $serviceLocator->get('someService');  
    //...$someService использование сервиса  
}
```

```
'someModule.someServiceName' => [  
    'className' => \VendorName\SomeModule\Services\SomeService::class,  
    'constructorParams' => ['foo', 'bar'],  
]  
'someModule.someServiceName' => [  
    'className' => \VendorName\SomeModule\Services\SomeService::class,  
    'constructorParams' => static function () {  
        return ['foo', 'bar'];  
    },  
]
```

```
$serviceLocator = \Bitrix\Main\DI\ServiceLocator::getInstance();  
$someGoodServiceName = $serviceLocator->get('someGoodServiceName');  
$someServiceName = $serviceLocator->get('someServiceName');
```

Запутанность кода

obfuscate — делать неочевидным, запутанным или **запутывание кода** — приведение исходного кода или исполняемого кода программы к виду, сохраняющему её функциональность, но затрудняющему анализ, понимание алгоритмов работы



PSR



описание psr



Плохой код

```
1 <?php
2
3
4 namespace INY\EmployeeTraining\Services\Example1;
5
6
7 use Bitrix\Main\UserTable;
8 use Bitrix\Main\ORM\Query\Query;
9 use Bitrix\Main\Engine\CurrentUser;
10
11
12 class Example
13 {
14     1 usage
15     const FIELD_DATA_CREATED = 'DATA_CREATED';
16     1 usage
17     function getQueryFilter(){
18         $nameList = ['Андрей', 'Леша', 'Коля', 'Маша'];
19         return Query::filter()→logic(Query::filter()::LOGIC_OR)→where( ...filter: self::FIELD_DATA_CREATED, ">", '12.12.2022')→whereIn( column: 'UF_NAME', $nameList)
20         →whereNotNull( column: 'GROUPS');}
21     function execute()
22     {
23         $result = [];
24         $resultUser = UserTable::query()→setSelect(['*'])→where( ...filter: 'ID', '=', CurrentUser::get()→getId())→where($this→getQueryFilter())
25         →setCacheTtl( ttl: 3600)→exec();
26         while ($user = $resultUser→fetch()) if( $user['ACTIVE'] === 'Y' ) $result[] = $user;
27         return $result;
28     }
29 }
```



Отформатированный код по psr12

```
1 <?php
2
3 namespace INY\EmployeeTraining\Services\Example1;
4
5 use Bitrix\Main\UserTable;
6 use Bitrix\Main\ORM\Query\Query;
7 use Bitrix\Main\ORM\Query\Filter\ConditionTree;
8
9 /**
10  * class Example
11  *
12  * @package INY\EmployeeTraining\Services\Example1
13  */
14 class Example
15 {
16     1 usage
17     ... public const FIELD_DATA_CREATED = 'DATA_CREATED';
18
19     ... /**
20     ... * @return array
21     ... * @throws \Bitrix\Main\ArgumentException
22     ... * @throws \Bitrix\Main\ObjectPropertyException
23     ... * @throws \Bitrix\Main\SystemException
24     ... */
25     ... public function execute(): array
26     ... {
27         ... $result = [];
28         ... $resultUser = UserTable::query()
29         ...     →setSelect(['*'])
30         ...     →where( ...filter: 'ID', '=', \Bitrix\Main\Engine\CurrentUser::get()→getId())
31         ...     →where(
32         ...         ... $this→getQueryFilter()
33         ...     )
34         ...     →setCacheTtl( ttl: 3600)
35         ...     →exec();
36         ... while ($user = $resultUser→fetch()) {
37         ...     if ($user['ACTIVE'] === 'Y') {
38         ...         $result[] = $user;
39         ...     }
40         ... }
41
42     ... return $result;
43     ... }
44
45     ... /**
46     ... * @return ConditionTree
47     ... * @throws \Bitrix\Main\ArgumentException
48     ... */
49     1 usage
50     ... protected function getQueryFilter(): ConditionTree
51     ... {
52         ... $nameList = ['Андрей', 'Леша', 'Коля', 'Маша'];
53
54         ... return Query::filter()
55         ...     →logic(Query::filter()::LOGIC_OR)
56         ...     →where(
57         ...         ... _filter: self::FIELD_DATA_CREATED,
58         ...         ... ">",
59         ...         ... '12.12.2022'
60         ...     )
61         ...     →whereIn( column: 'UF_NAME', $nameList)
62         ...     →whereNotNull( column: 'GROUPS');
63     ... }
```

Конфигурация phpcs

Конфиг

Команда без конфига

```
✓ bash  
php phpcs.phar -s -p -colors --standard=PSR12 \  
--extensions=php --ignore=install/index.php \  
--exclude=Generic.ControlStructures.InlineControlStructure \  
--runtime-set testVersion 8.1 \  
./local/modules/
```

```
''' xml  
<?xml version="1.0"?>  
<ruleset name="PHP_CodeSniffer">  
  <description>PHP_CodeSniffer configuration</description>  
  <arg name="colors" />  
  <arg value="p" />  
  <arg value="s" />  
  <arg name="report" value="code" />  
  <arg name="extensions" value="php" />  
  
  <rule ref="PSR12" />  
  
  <rule ref="Generic.NamingConventions.UpperCaseConstantName.ClassConstantNotUpperCase">  
    <exclude-pattern>bootstrap.php</exclude-pattern>  
  </rule>  
  
  <rule ref="PSR1.Classes.ClassDeclaration.MissingNamespace">  
    <exclude-pattern>install/index.php</exclude-pattern>  
  </rule>  
  
  <rule ref="Squiz.Classes.ValidClassName.NotCamelCaps">  
    <exclude-pattern>install/index.php</exclude-pattern>  
  </rule>  
  
  <rule ref="PSR1.Methods.CamelCapsMethodName.NotCamelCaps">  
    <exclude-pattern>install/index.php</exclude-pattern>  
  </rule>  
  
  <rule ref="PSR1.Files.SideEffects.FoundWithSymbols">  
    <exclude-pattern>install/index.php</exclude-pattern>  
  </rule>  
</ruleset>
```

Команда с конфигом

```
''' bash  
php phpcs.phar --standard=/home/nivanov/ni.core/local/modules/iny.core/phpcs.xml --runtime-set testVersion 8.1 /home/nivanov/ni.core/local/modules
```

Спасибо за внимание