

CodeWeek. 



Išlipus iš 1 PHP failo

Apie mane



Aurelijus Banelis

PHP programuotojas

aurelijus.banelis.lt

aurelijus@banelis.lt



Sužinosite



Išlipus iš 1 PHP failo

»»» Pabandydams užtenka 1 failo,
kokie iššūkiai,
kai programos kodo daugėja

»»» 3 praktiniai patarimai apie
naudingiausią tobulėjimo kelią

1 PHP failas



```
$today = date('Y-m-d');  
if (date('z') == 0xFF) {  
    echo "Programuotojų diena";  
  
} else if ('2015-10-10' <= $today &&  
    $today <= '2015-10-10') {  
    echo "Programuotojų savaitė";  
  
} else {  
    echo "Paprasta diena";  
}
```

Bet kas toliau?



php | programmer

php programmer salary
php programming
php programmer job description
freelance php programmer

About 14,200,000 results (0.53 seconds)

php programavimas |

php programavimas
php programavimo kursai
php programavimo kalba
php programavimas pradedantiesiems

About 171,000 results (0.49 seconds)

GitHub search results for 'php' showing repository statistics and language usage.

Category	Count
Repositories	147,366
Code	267,076,428
Issues	1,034,956
Users	2,829

Language	Count
PHP	179,695,302
HTML	11,661,346
JavaScript	6,717,105

Bet kas toliau?



Dirbančių

Hacker'ių

Akademikų

Bet kas toliau?



Dirbančių

Hacker'ių

Naudok karkasą!

Imk kas want bangos!

Akademikų

Suprask principus!

Bet ką toliau?



Dirbančių

- Imk Symfony2
- Kodėl
- Visi jį naudoja
- O pats?
- Neseniai dirbau su OXID'u
- Tai jis geresnis?
- Jis viską daro savaip
- O didelės kompanijos?
- Viena naudoja Zend
- Tai jis geresnis?
- Jis irgi viską savaip
- O namie?
- Namie tai ImpressPages
- O jis skiriasi nuo Zend?
- Aišku, jis irgi savaip daro

Magento

Phalcon

ImpressPages

Drupal

Zend

Typo3

OXID

Symfony2

Cake

Wordpress

Naudok karkasą!

Akademikų

- Karkasus išmoksi vėliau
- Tai ką mokytis dabar
- PHP objektinė, tai OOP
- O kas op op?
- Na visi tie IoC, FLORIC, SOLID
- Tai, tai D?
- O tu nori TDD ar BDD
- Ne man tik D žinoti reikia
- Galima pradėti ir nuo DDD

Suprask principus!

Hacker'ių

- PHP žinau mažai, bet va su Scala
- O ką su ja?
- Visur matau monadus
- Čia tas pats, kas li...
- Su listais irgi ten daug galima
- Ką galima?
- Na foldo nėra, bet yra reduce
- O kam jis?
- Na šalia map funkcijų
- O kažka paprasto
- PHP turi Closure'us
- Nenoriu šito
- Tai tau gal nori su Aspectais

Reduce

Covariance

Monad

Compile

Currying

Async

Map

Folds

Imk kas want bangos!!

DDD

SOLID

TDD

IODA

MVC

DSL

KISS

SQRS

BDD

OOP

SOLID

DI

IoC

P.S. pavyzdžiai paimti atsitiktine tvarka. Dydis, eiliškumas svarbos nenurodo

Paprastesnio!

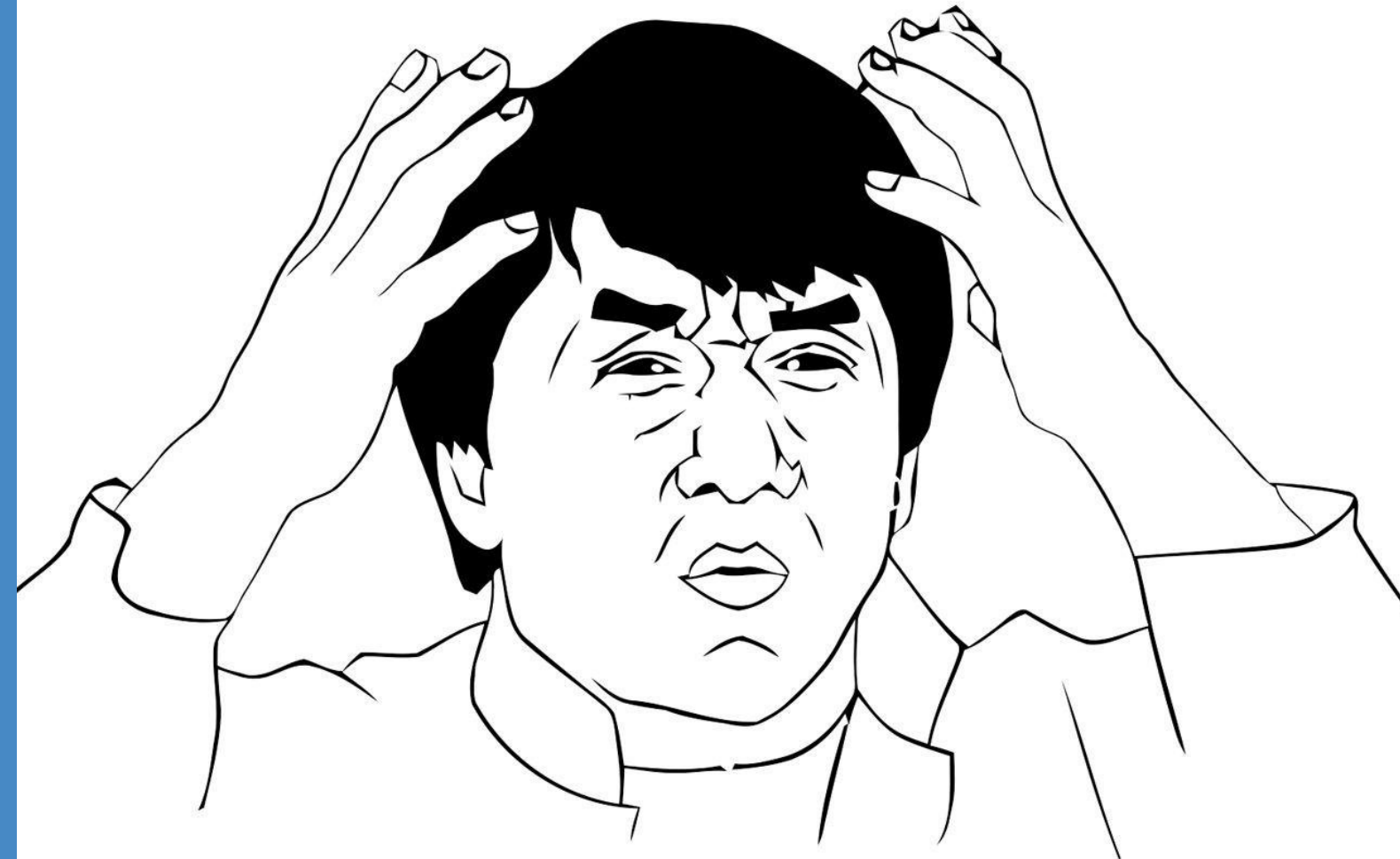


ultrad.com.br

**Negi reikia
visko?**

**Nuo ko
pradėti?**

**Gal yra
lengvesnis
kelias?**



Iš patirties



Daugiausia naudos davė tobulėjant/įkvėpė:



Modulių testavimas

Stefan Pribsch "Risk-Free Framework Usage"



Programa ne medis

Kevlin Henney "A System Is Not a Tree"



Idealus vs naudojamas

Silicon Valley Comes to Baltics

Iš patirties



Daugiausia naudos davė tobulėjant/įkvėpė:



Modulių testavimas

Stefan Pribsch "Risk-Free Framework Usage"



Programa ne medis

Kevlin Henney "A System Is Not a Tree"



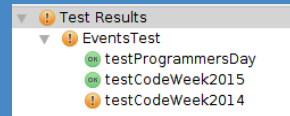
Idealus vs naudojamas

Silicon Valley Comes to Baltics

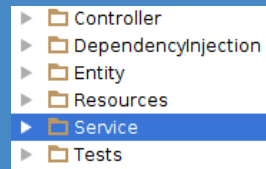
Testavimas



Pradėkite rašyti modulių testus



Blogai stuktūrai sunku rašyti testus



Pagerėja sistemos struktūra/architektūra

Teisingas sistemos suskaidymas
SOLID, MVC ar kiti principai net to nežinant

Kodo perpanaudojimas
Išsvajotas programų sistemų tikslas

Kita sistema
Kita sistema

Modulių testai



```
class EventTest extends PHPUnit_Framework_TestCase {  
    public function testCodeWeek2015() {  
        $s = new Events();  
        $this->assertTrue($s->isCodeWeek('2015-10-10'));  
        $this->assertFalse($s->isCodeWeek('2015-10-09'));  
    }  
  
    public function testCodeWeek2014() {  
        $s = new Events();  
        $this->assertTrue($s->isCodeWeek('2014-10-10'));  
        $this->assertFalse($s->isCodeWeek('2014-10-09'));  
    }  
}
```

Test Results

EventsTest

- OK testProgrammersDay
- OK testCodeWeek2015
- ! testCodeWeek2014

Pagal įvestus
duomenis tikimės
tam tikrų rezultatų

Buvo 1 PHP failas



```
$today = date('Y-m-d');  
if (date('z') == 0xFF) {  
    echo "Programuotojų diena";  
  
} else if ('2015-10-10' <= $today &&  
    $today <= '2015-10-10') {  
    echo "Programuotojų savaitė";  
  
} else {  
    echo "Paprasta diena";  
}
```

Pritaikėme



```
class Events {  
    public function isProgrammersDay($today) {  
        return date('z', strtotime($today)) == 0xFF;  
    }  
    public function isCodeWeek($today) {  
        $date = date('Y-m-d', strtotime($today));  
        return '2015-10-10' <= $date  
            && $date <= '2015-10-18';  
    }  
}
```

➤ Suskaidėme į funkcijas
➤ Iškelėme argumentus
➤ Tapo perpanaudojamas

- ▶ Controller
- ▶ DependencyInjection
- ▶ Entity
- ▶ Resources
- ▶ Service
- ▶ Tests

Kodėl geresnis?



Testai

Lankytojai

Modulis

» Ne tik kodo gabalas,
bet modulis - bent 2
skirtingi to paties
kodo panaudojimo
būdai

- » Vienas panaudojimas serveryje
- » Kitas panaudojimas testuojant

Kita sistema
Kita sistema

Iš patirties



Daugiausia naudos davė tobulėjant/įkvėpė:



Modulių testavimas

Stefan Pribsch "Risk-Free Framework Usage"



Programa ne medis

Kevlin Henney "A System Is Not a Tree"



Idealus vs naudojamas

Silicon Valley Comes to Baltics

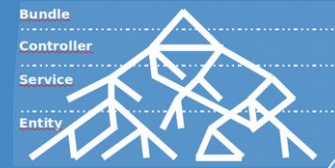
Programa - medis?



» Hierarchijomis lengva aiškinti



» Realybėje yra daugiau sąryšių

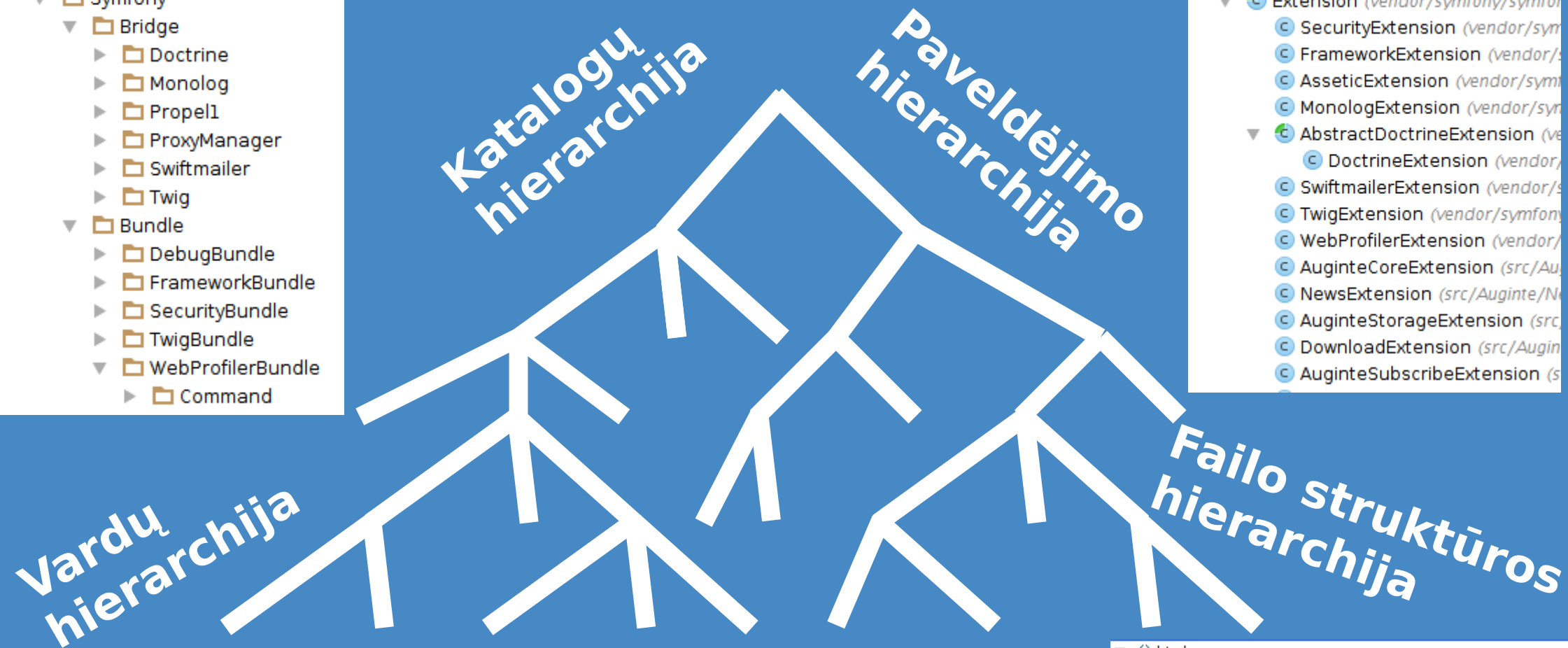
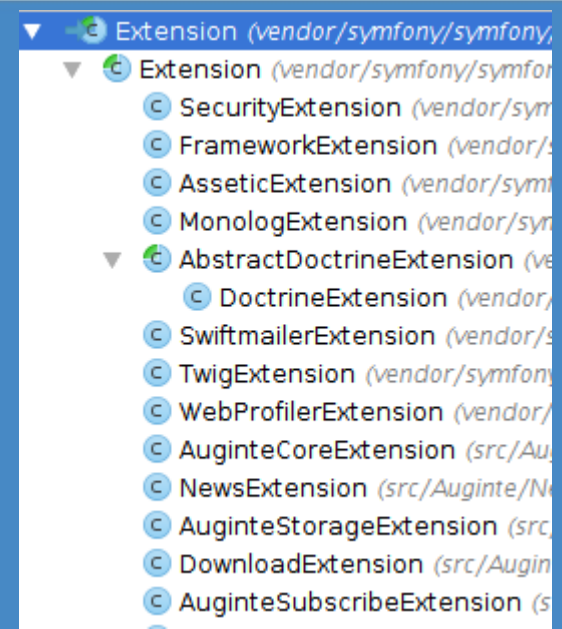
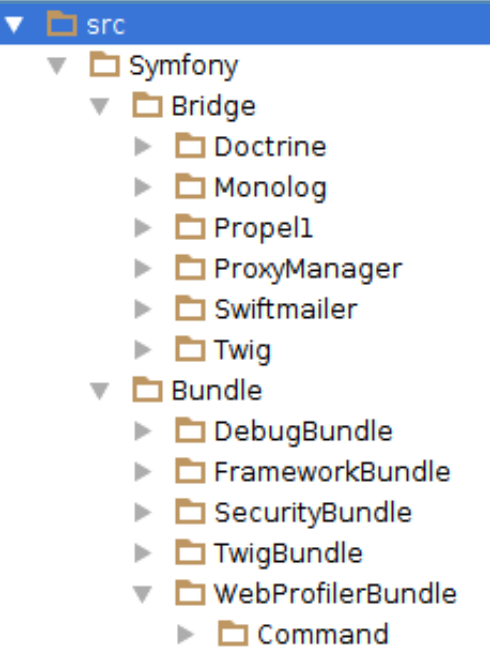


» Išnaudoti PHP kalbos ir įrankių galimybes

» Grupavimas pagal daiktus
class, extends, abstract, parent

» Grupavimas pagal funkcijas
Interface, implements, trait, use

Medžiai visur



```
use Symfony\Component\DependencyInjection\nuse Symfony\Component\Config\FileL\nuse Symfony\Component\HttpKernel\D\nuse Symfony\Component\DependencyInjection
```

```
<html\n  <head\n    <meta (http-equiv="Content-Type" content="text/html;\n    <robots:meta (content="noindex,nofollow")\n    <title\n    <link (rel="stylesheet" href="bundles/framework/css/st\n    <link (rel="stylesheet" href="bundles/framework/css/bc\n    <link (rel="stylesheet" href="bundles/sensiodistributio\n  <body\n    <content:div
```

Teorijoje

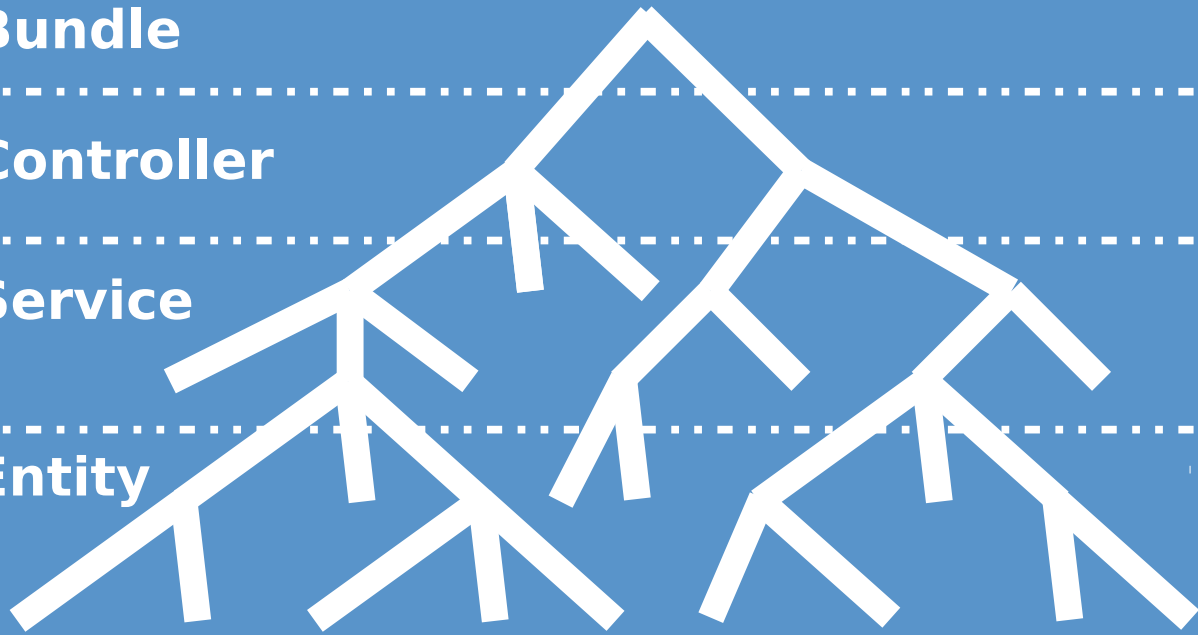


Bundle

Controller

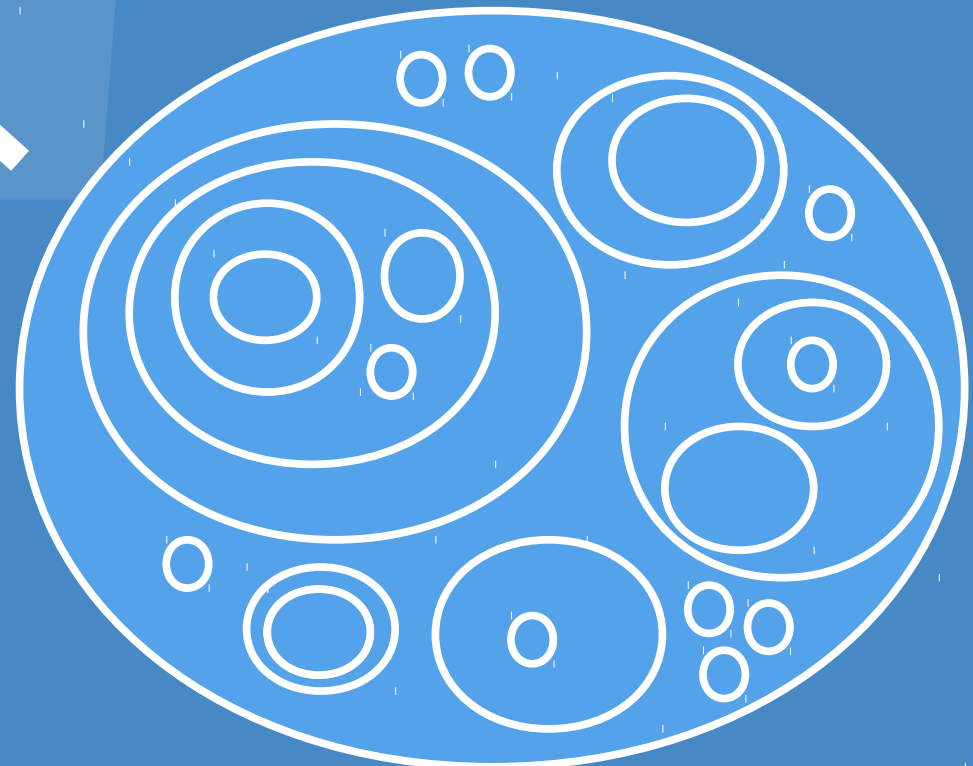
Service

Entity



Sudaryta iš
nepriklausomų
komponentų

Štai mūsų
sistemos
architektūra



Realybėje

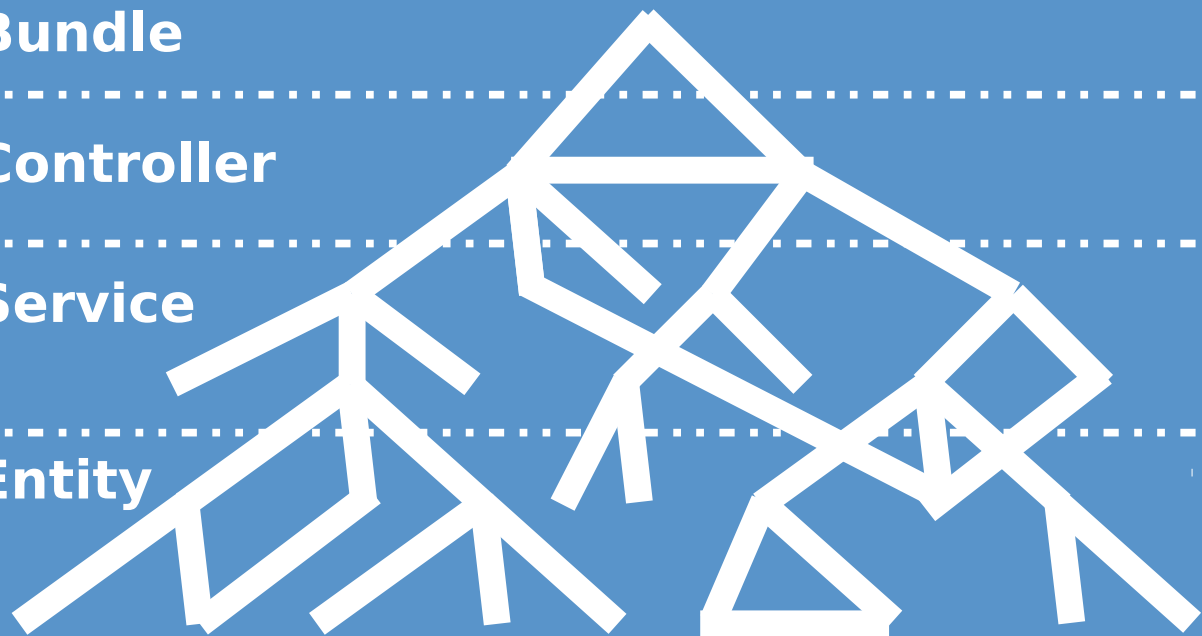


Bundle

Controller

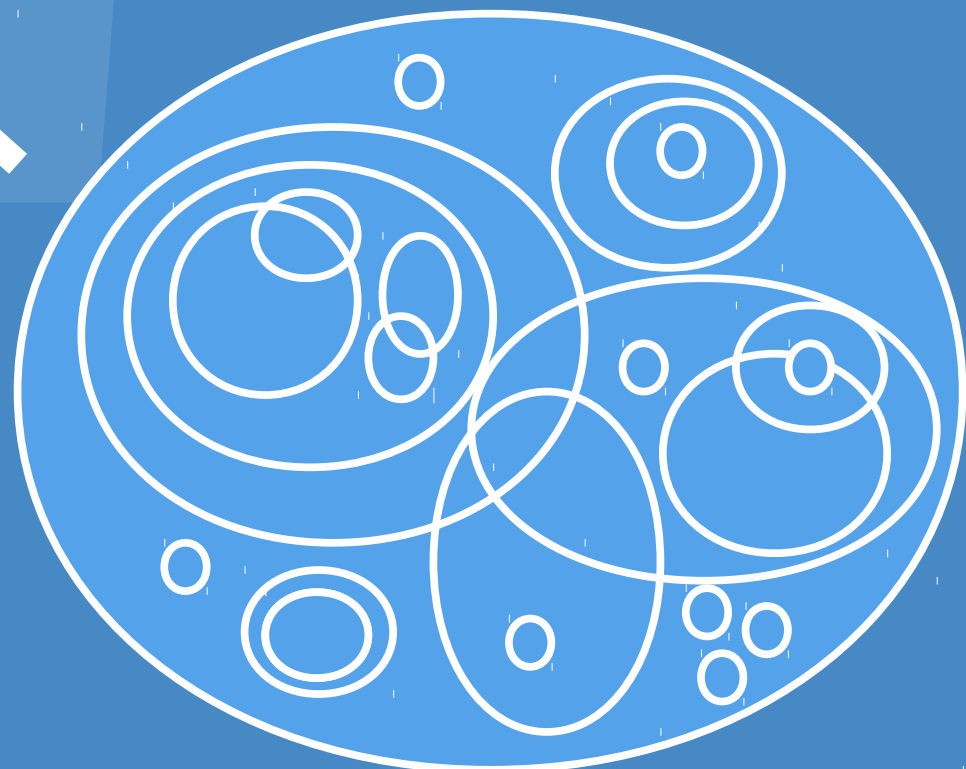
Service

Entity



**Beveik viskas
nuo visko
priklauso arba
ne savo vietoje**

**Kai reikia
taisyti/keisti:
pasimato tikroji
„architektūra“**



Kodėl kertasi?



Pagal operacijas

Pagal objektus

	Create	Activate	GetOne	GetAll	...
User					
Product					
Set					
Supplier					
...					

Naudotojas
Gamintojas
Prekė
Paprasta
Komplektas
Skaitymas
Rodyti vieną
Rodyti visus
Rašymas
Sukurti
Aktyvuoti

Skirtingas
grupavimas,
bet bendros
dalys

Skaitymas
Rodyti vieną
Rodyti visus
Rašymas
Sukurti
Aktyvuoti
Naudotoją
Gamintoją
Prekę
Paprastą
Komplektą

Programos kodas gali būti sudėliotas labai įvairiais būdais. Vien jau grupavimas pagal Objektus ir pagal operacija sukuria 2 skirtingas hierarchijas.
Abu grupavimai turi labai panašaus kodo, todėl norisi jį išskirti, o ne kopijuoti kiekvieną kartą.

PHP galimybės



```
<?php
abstract class Product {
    protected function save() {
        echo "Saugojame\n";
    }
}
class Simple extends Product {
    function create() {}
    function activate() {}
}
class Set extends Product {
    function create() {
        parent::save();
    }
    function activate() {}
}
(new Set())->create();
```

PHP turi visas galimybes įgyvendinti tiek grupavimą pagal objektus, tiek pagal operacijas nedubliuojant bendro kodo/logikos.

Grupavimui pagal objektus jau nuo PHP 5.0 versijos yra naudojamas paveldėjimas (*class, abstract, extend, parent* raktažodžiai).

Grupavimui taip pat galima naudoti ir vardų sritis (angl. *namespace*), bet jos neturi priemonių bendro kodo panaudojimui.

Grupavimui pagal operacijas irgi nuo PHP 5.0 yra aprašoma objekto sąsaja (*interface, implements* raktažodžiai).

Bendram operacijų kodui buvo naudojama kompozicija, bet nuo PHP 5.4 versijos daug patogiau tai atlikti su pažymit, kad objektas turi tam tikrų bruožų (trait, use raktažodžiai). Kas geriausia, kad taip įkeltus metodus galima pervadinti, taip pritaikant pagal aprašytą sąsają.

```
<?php
interface Persistable {
    function create();
    function activate();
}
trait Common {
    function save() {
        echo "Saugojame\n";
    }
}
class Simple implements Persistable {
    function create() {}
    function activate() {}
}
class Set implements Persistable {
    use Common { save as create;}
    function activate() {}
}
(new Set())->create();
```

Įrankiai padės



Choose subclass of Product

- Simple .../A.php
- Set .../A.php
- WishListProduct .../D.php



Choose implementation of Persistable

- Simple .../B.php
- Set .../B.php
- WishListProduct .../D.php

```
class WishListProduct
extends Product
implements Persistable, Searchable
```

```
{
    function create() {}
    function activate() {}
    function getOne() {}
    function getAll() {}
```

```
} https://gist.github.com/aurelijusb/e178424ea6ae7921ba2b
```

Na ir aišku abu grupavimo būdus galima sujungti. Šašajos (angl. *Interface*) būdas turi privalumą - objektas gali priklausyti daugumai hierarchijų, kai paveldėjimo atveju tik vienai.

Šaunu, kad įrankiai leidžia rasti elementus grupėse: tiek vaikus, tiek šašajos įgyvendinimą. Kartu su katalogų struktūra gauname pakankamai grupavimo būdų.

Išreikštos grupės vėliau leidžia lengviau daryti pakeitimus, testuoti ar tiesiog suprasti architektūrą.

Išreikšti
grupavimai
palengvina
architektūros
supratimą

Iš patirties



Daugiausia naudos davė tobulėjant/įkvėpė:



Modulių testavimas

Stefan Pribsch "Risk-Free Framework Usage"



Programa ne medis

Kevlin Henney "A System Is Not a Tree"



Idealus vs naudojamasis

Silicon Valley Comes to Baltics

Norėjau, bet išėjo...



» Išgirdom apie *trait'us*, dėsime visur, kur reikia ir kur nereikia?

```
class Related {  
    function __construct(  
        ProductReader $reader,  
        RelatedCreator $creator,  
        RelatedAggregator $aggregator,  
        RelatedFormatter $formatter) {}  
    /** @return RelatedProduct[] */  
    public function getRelated() {}  
}
```

» *Trait'ai* nepatiko, perrašome viską paveldėjimu?

```
HEAD develop Postpone release  
JSON-LD: Structured data: Company  
master v0.8.1 v0.8.1: Using boot  
Augintejs: Locally cached Facebook/  
Augintejs: Loading/saving state from  
Simple storage component: new, sa
```

» „Kol pabaigiau, pridygo konkurentų“

» Turėti platesnę struktūros viziją, bet įgyvendinti ir tikrinti mažais etapais

Persistengus



» „Kodėl neužtenka vieno failo?“

```
class Related {  
    function __construct(  
        ProductReader $reader,  
        RelatedCreator $creator,  
        RelatedAggregator $aggregator,  
        RelatedFormatter $formatter) {}  
    /** @return RelatedProduct[] */  
    public function getRelated() {}  
}
```

» Priskaičius teorijų reikėtų nepersistengti

Persistengus



» „Kodėl neužtenka vieno failo?“

```
class Related {  
    function __construct(  
        ProductReader $reader  
    ) {}  
    /** @return RelatedProduct[] */  
    public function getRelated() {}  
}
```

» Priskaičius teorijų reikėtų nepersistengti
Sėkti jomis, kai jos atneša realią naudą

Dažnai tikrinti



» „Kokie lankytojų atsiliepimai?“

» Versijų kontrolės sistemos (pvz. Git) tampa geriausiu tavo padėjėju

» Galimybė eksperimentuoti ir grįžti

» Nuolatinis tobulinimas, atsižvelgus į rezultatus

```
● HEAD develop Postpone release
● JSON-LD: Structured data: Company
● master v0.8.1 v0.8.1: Using boot
● Augintejs: Locally cached Facebook/
● Augintejs: Loading/saving state from
● Simple storage component: new, sa
```

Gausėjančio kodo suvaldymui pravers:



Testavimas

Nes bus suskaidyta nepriklausomomis dalimis



Grupavimas PHP sintakse

Nes kodas ir įrankiai pasakys kitiems, kas ir kodėl susiję



Veikainčios tarpinės versijos

Nes galėsite parodyti, patobulinti ir kurti naudingesnes programas

Nuorodos



- <https://phpunit.de/>
- http://symfony.com/doc/current/book/from_flat_php_to_symfony2.html
- http://symfony.com/doc/current/book/service_container.html
- https://en.wikipedia.org/wiki/Object-oriented_programming
- [https://en.wikipedia.org/wiki/SOLID_\(object-oriented_design\)](https://en.wikipedia.org/wiki/SOLID_(object-oriented_design))
- <https://webtechcon.de/2014/sessions/risk-free-framework-usage>
- <http://www.slideshare.net/Kevlin/a-system-is-not-a-tree>
- <https://www.coursera.org/course/proglang>
- <http://php.net/manual/en/language.oop5.inheritance.php>
- <http://php.net/manual/en/language.oop5.traits.php>
- <http://geekswithblogs.net/theArchitectsNapkin>
- <https://medium.com/@webseanhickey/the-evolution-of-a-software-en>
-

Ačiū. Klausimai?



Gausėjančio kodo suvaldymui pravers:



Testavimas

Nes bus suskaidyta nepriklausomomis dalimis



Grupavimas PHP sintakse

Nes kodas ir įrankiai pasakys kitiems, kas ir kodėl susiję



Veikainčios tarpinės versijos

Nes galėsite parodyti, patobulinti ir kurti naudingesnes programas