

JAK NA ČISTÝ KÓD

a ŠKÁLOVATELNOU ARCHITEKTURU KÓDU

DOZVÍTE SE:

- ▶ Jak udržovat stabilní kvalitu kódu na projektu
- ▶ Jak efektivně vrstvit kód a oddělovat jednotlivé zodpovědnosti v kódu
- ▶ Praktické ukázky běžně používaných principů pro dosahování snadno pochopitelného a rozšiřitelného kódu



1. 3. 2023



SIT Port



od 18 hod



TechTower - SeedUp Space

Koterovská 2827/152, Slovany, Plzeň

RTsoft s.r.o.

- Plzeňská softwarová společnost působící na trhu již přes 20 let
- Vývoj webových a mobilních aplikací na míru
- Více než 300 dokončených projektů
- Mezi naše partnery patří MND, Knihy Dobrovský (3× e-shop roku), Sazka, Sportisimo, Darujme.cz a mnoho dalších
- Více než 40 vývojářů
- Přes e-shopy našich zákazníků proteklo za 2021 přes 2 mld. Kč

RTsoft s.r.o.



ČISTÝ KÓD A NÁVRH ARCHITEKTURY

Co to je?

- komunikační prostředek
- flexibilní guideline
- způsob unifikace řešení běžných úkolů

Přínosy

- úspora času při čtení a pochopení kódu (poměr 10:1)
- méně náchylný kód na chyby
- snadnější testovatelnost kódu

Přínosy vol. 2

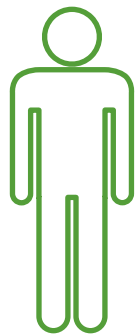
- lepší předvídatelnost kódu
- jednodušší onboarding vývojáře na projekt
- spokojenější vývojáři

Cost / benefit

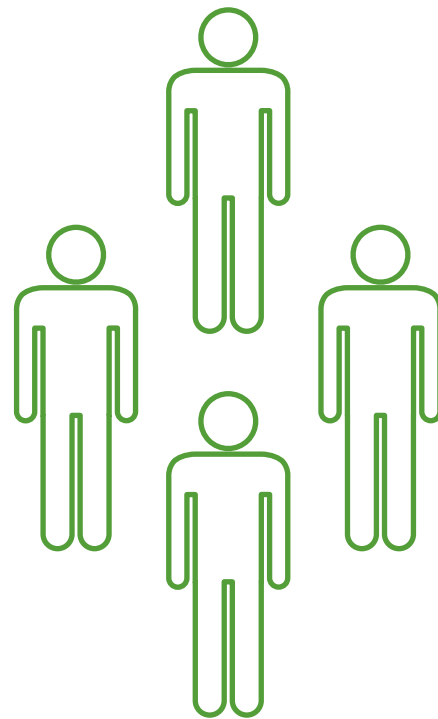
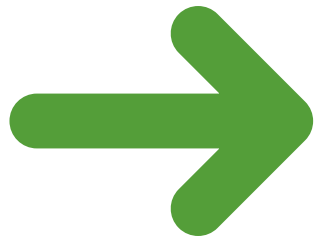
- psaní čistého kódu a dodržování pravidel by mělo řešit problémy
- krátkodobý x dlouhodobý pohled do budoucna
- business critical logic

Cost / benefit vol. 2

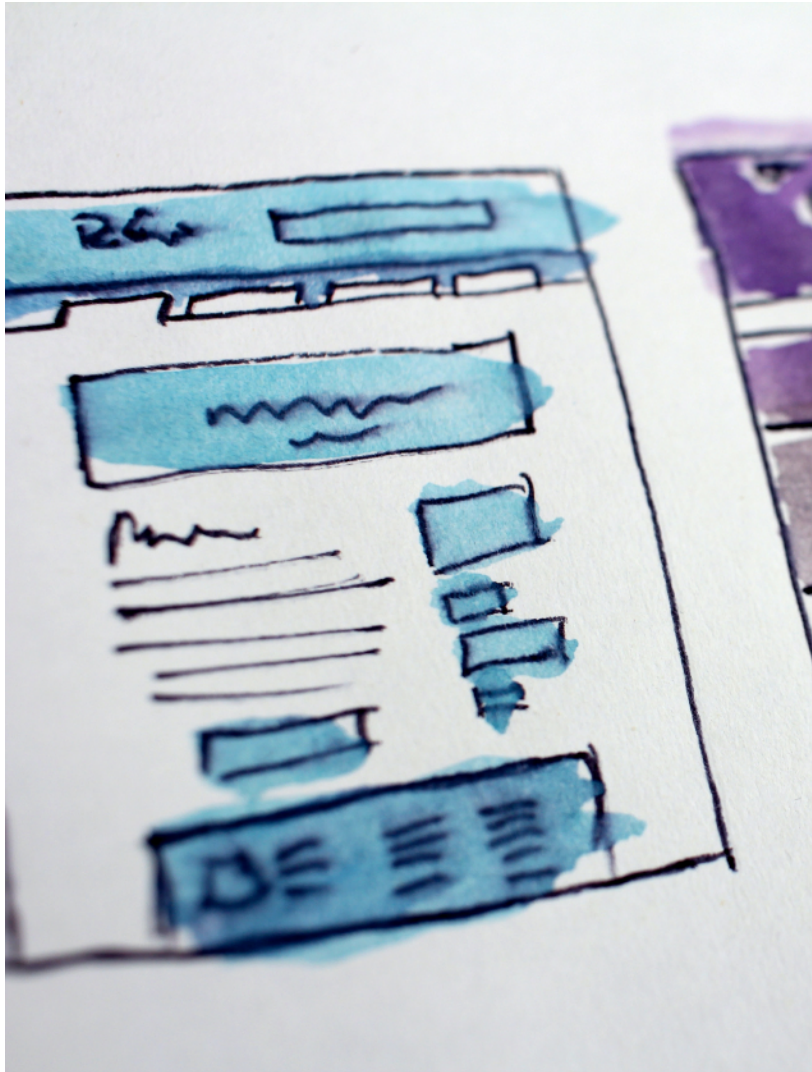
- benefits jsou zjevné zejména na větších projektech a týmech
- balanc mezi DX a businessovými požadavky
- pozor na technologický dluh



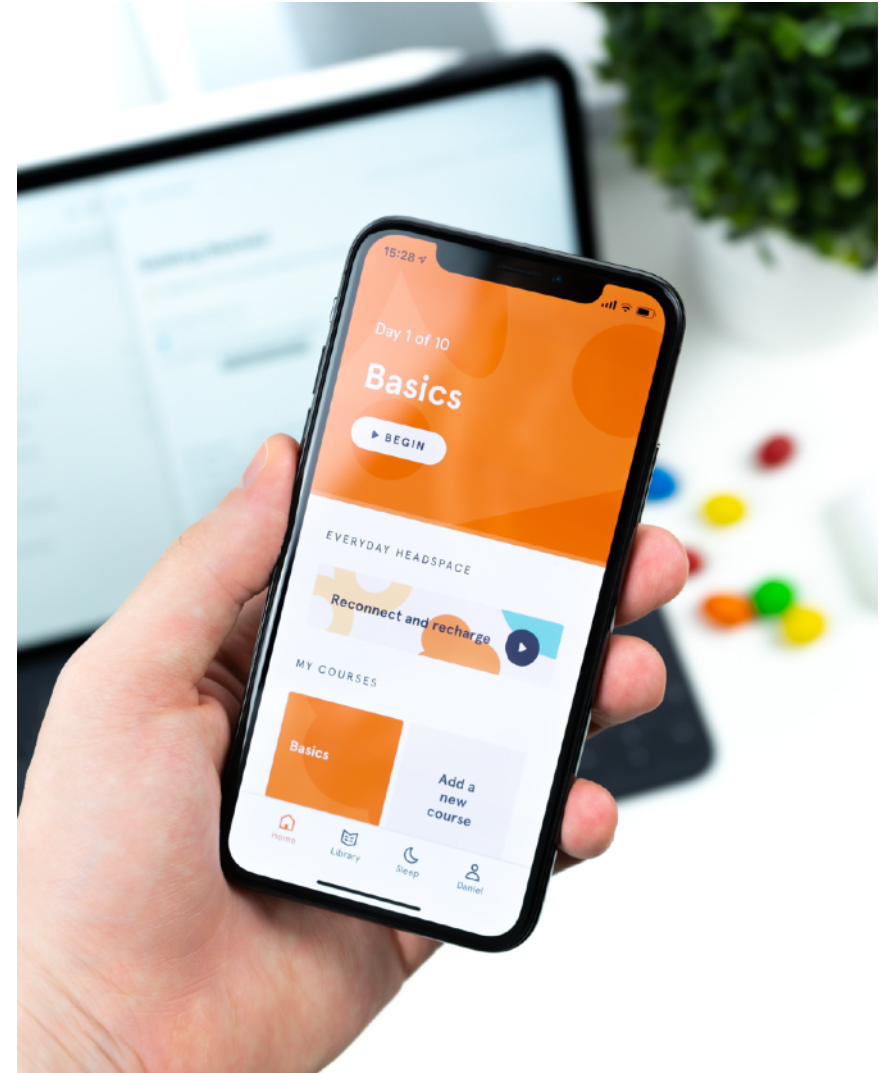
Freelancer



Vývojářský tým



VS



KDO PSAL TEN KÓD???



BUZZWORDS

DRY

Don't repeat yourself.

~~Don't repeat yourself.~~

~~Don't repeat yourself.~~

~~Don't repeat yourself.~~

YAGNI

You Ain't Gonna Need It

KISS

Keep It Simple Stupid

VRSTVENÍ KÓDU

Typy vrstev

- aplikační logika
- UI logika
- business logika
- DB logika

Oddělování vrstev

- framework agnostic
- zjednodušuje sdílení kódu mezi projekty
- zlepšuje testovatelnost

Separation of Concerns

- rozdělení kódu podle zodpovědností
- členění kódu tak, aby nevznikaly špagety a kód byl stále přehledný
- možné řešení - extrakce kódu
- v IDE extract variable nebo extract method

Single Responsibility Principle

- třída by měla dělat jen jednu specifickou věc (jednu zodpovědnost)
- rozdělování kódu do více tříd
- warning - spoustu obecných metod nesouvisející s danou třídou
- např. rozdělení tříd OrderFacade a OrderPriceCalculator

Dependency Inversion

- loose coupling / tight coupling
- injectování interfaces
- typické využití pro služby 3. stran

ARCHITEKTURA

Architektura projektu

- domény
- service vs value object
- rich model vs anemic model



Zdroj: knowledgehut



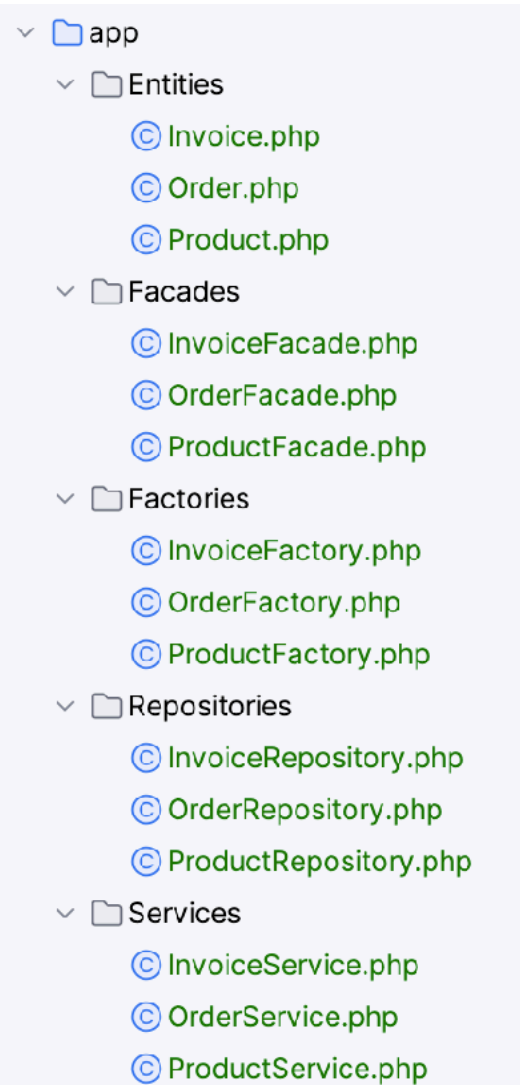
Facade

Repository

Service

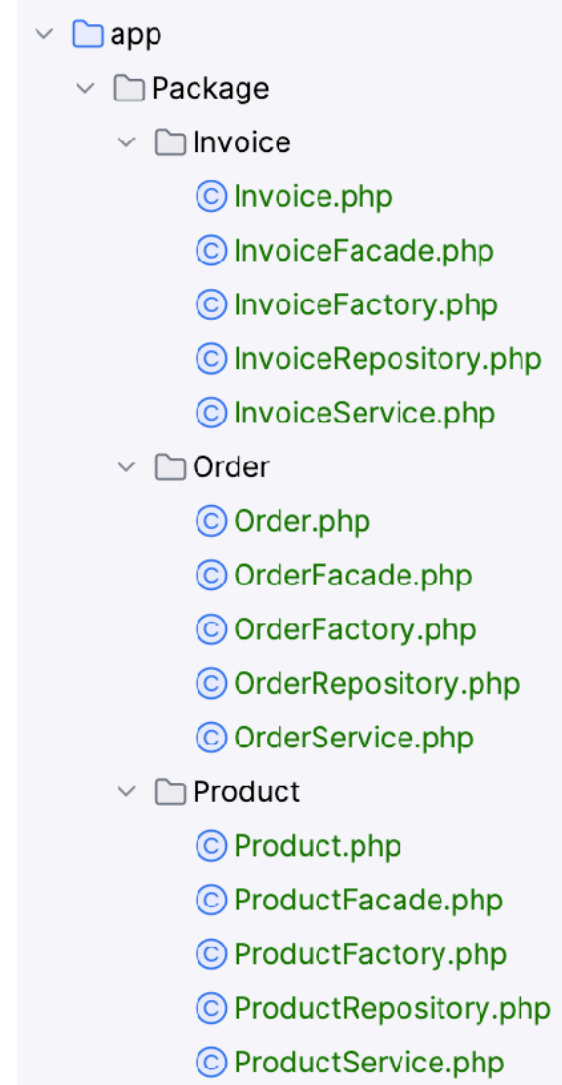
Entity

Horizontal

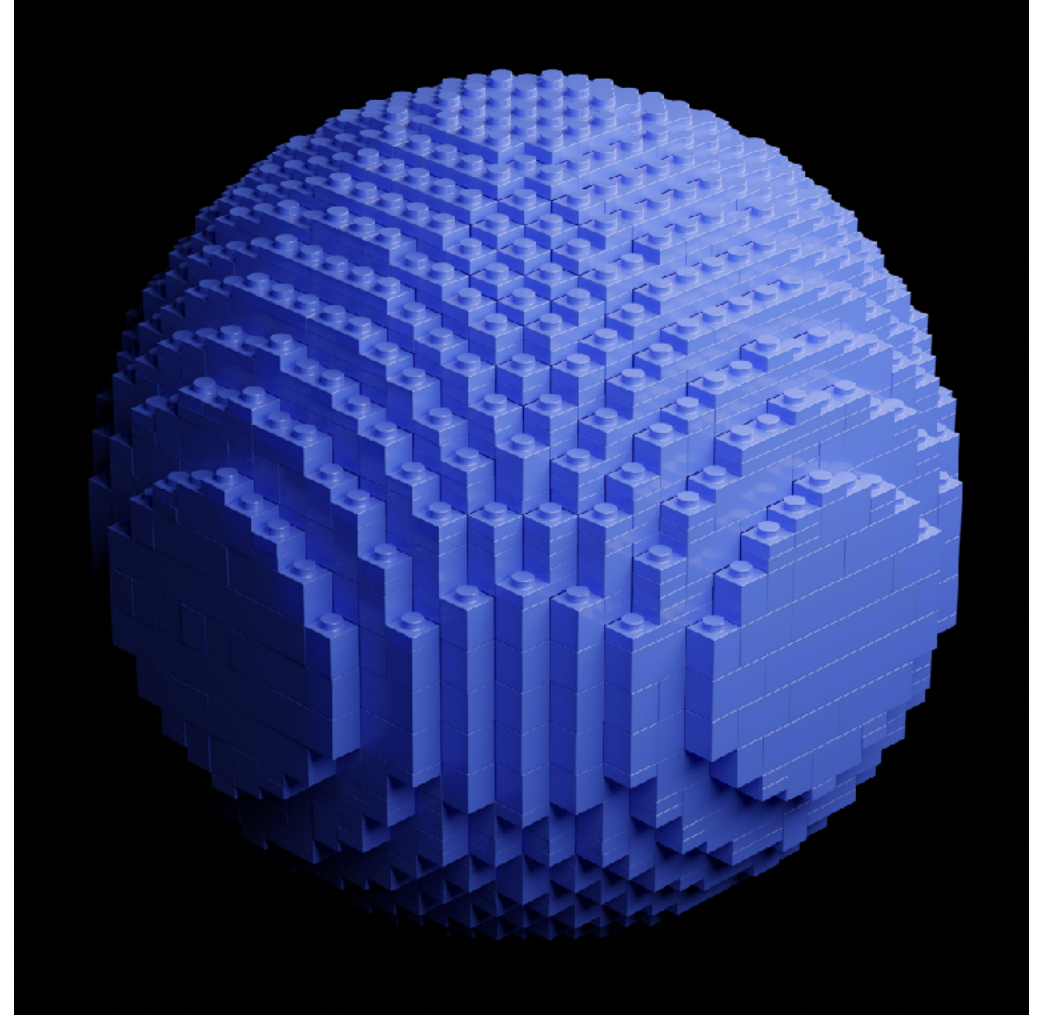


VS

Vertical



MODULARITA



Modularita

- dělení aplikace na skládatelné moduly
- projektové vs knihovní moduly
- důraz na dostatečnou abstrakci, pozor na polovičaté řešení

IMUTABILITA

Imutabilita

Kolik \$\$\$ se pošle do banky? Třída Cash je mutabilní.

```
public function transferCash()  
{  
    $cashToTransfer = new Cash(1000000000);  
    $this->logRecalculatedCashToDatabase($cashToTransfer);  
    $this->bankApi->sendCash($cashToTransfer);  
}
```


Imutabilita

Do banky se pošle 121 000 000 \$, tzn. o 21 000 000 \$ více, než chceme.

Proč?

```
public function logRecalculatedCashToDatabase(Cash $cash)
{
    $cash->setCash($cash->getValue() * 1.21);
    $this->saveToDatabase($cash);
}
```

Imutabilita

Možné řešení

```
public function setCash(int $cash)
{
    $this→cash = $cash;
}
```



```
public function withCash(int $cash)
{
    $newObject = clone $this;
    $newObject→cash = $cash;

    return $newObject;
}
```

Imutabilita

- neměnnost vnitřního stavu objektu
- místo změny vnitřního stavu se vrací nový objekt
- preventuje složitě odhalitelné business critical buggy
- nevytvářet zbytečné settery nebo používat “with” metody

BONUS



Bonus

- srozumitelný naming proměnných a funkcí
- správné zapouzdření `private` / `protected` / `public`
- preferování kompozice před dědičností v OOP

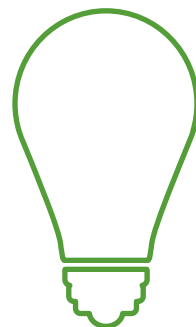
Bonus vol 2.

- nepoužívat boolean parametry měnící chování funkce
- psát použitelné komentáře
- používat linter

Dosahování čistého kódu

- postupný refaktoring a odbavování tech. dluhu
- code reviews
- postupná adaptace + inkubace

CO SI ZAPAMATO VAT



K zapamatování

- úspora času při čtení a pochopení kódu (poměr 10:1)
- méně náchylný kód na chyby
- snadnější testovatelnost kódu
- lepší předvídatelnost kódu
- jednodušší onboarding vývojáře na projekt
- spokojenější vývojáři





IT workshop

Novinky v PHP 8.*

The PHP logo, consisting of the lowercase letters "php" in a bold, black, sans-serif font with a white outline, set against a light blue oval background. The entire logo is framed by a large, dark blue, stylized swoosh that curves around the left and bottom of the oval.

php

29. 3. 2023 v 18:00

Restaurace U Salzmannů

Salonek v 1. patře

Vstup a občerstvení zdarma

Děkujeme a budeme rádi za feedback!



Stále nabíráme!

www.rtsoft.cz/kariera