



Informační systémy
na míru



Business
analytika



Mobilní
aplikace



Weby
a E-shopy

Funkcionální programování v PHP



Martin Hora
hora@rtsoft.cz



1. MÍSTO

IBM
FIRMA ROKU 2021
PLZEŇSKÉHO KRAJE



Informační systémy
na míru



Business
analytika



Mobilní
aplikace



Weby
a E-shopy

Co to je funkcionální programování

- technika deklarativního zápisu kódu
- pure functions
- eliminace side effectů
- imutabilní architektura kódu
- vhodné pro paralelní operace



Deklarativní / imperativní způsob zápisu kódu

- deklarativní způsob - definujeme, čeho chceme dosáhnout, zbytek se děje na pozadí (expresivnější sémantika)
- imperativní způsob zápisu - explicitně říkáme, jakým způsobem se má kód provést



Imperativní způsob - mapování

```
$productId = [];  
foreach ($products as $product)  
{  
    $productId[] = $product->getId();  
}
```



Deklarativní způsob - mapování

```
$productIds = $products->map(fn (Product $product) => $product->getId());
```

```
$productIds = array_map(  
    fn (Product $product) => $product->getId(),  
    $products,  
);
```



Imperativní způsob - filtrace + mapování

```
$productIds = [];  
foreach ($products as $product)  
{  
    if ($product->isActive())  
    {  
        $productIds[] = $product->getId();  
    }  
}
```

Deklarativní způsob - filtrace + mapování

```
$productId = $products
```

```
->filter(fn (Product $product) => $product->isActive())
```

```
->map(fn (Product $product) => $product->getId());
```

```
$productId = array_map(
```

```
fn (Product $product) => $product->getId(),
```

```
array_filter($products, fn (Product $product) => $product->isActive()),
```

```
);
```



Generika v PHP

- řešena pomocí PHPStanu
- zápis pomocí anotace `@template T`
- podporuje expresivitu a předvídatelnost kódu, odhaluje bugy ještě před interpretací kódu
- <https://phpstan.org/blog/generics-in-php-using-phpdocs>

Generika v PHP

```
/**  
 * @param ArrayList<Product>  
 * @return ArrayList<OrderProduct>  
 */  
function saveProductsToOrder(ArrayList $products): ArrayList
```



Generika v PHP

```
// indexMap vrátí ArrayMap<int, Product>
```

```
$indexedProducts = $products->indexMap(fn (Product $product) => $product->getId());
```

```
// PHPStan zde detekuje, že getNonExistingProperty neexistuje
```

```
$indexedProducts[4]->getNonExistingProperty();
```

Pure functions

Co bude výsledkem těchto volání? Obě funkce vrátí součin 2 čísel.

Co bude výsledkem těchto volání?

Pure function

\$calculator->multiply(3, 4);

Non pure function

\$calculator->multiply();

Pure functions

- pure function je funkce, která vždy vrátí ten samý výstup na ten samý vstup
- preferováním pure functions se podporuje předvídatelnost a čitelnost kódu (10:1)
- místu, které porušuje princip pure function se nazývá side effect
- OOP změna vnitřního stavu třídy
- načtení dat z DB nebo API



Informační systémy
na míru



Business
analytika



Mobilní
aplikace



Weby
a E-shopy

Pure functions - jak jich dosáhnout

- eliminace závislostí metod na vnitřním
- předávání dat přes parametry
- výhodou např. cachování výsledků funkce

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

- neměnnost vnitřního stavu objektu
- při jeho změně se nemění objekt samotný, ale vrací nová instance
- preventuje složitě odhalitelné business critical buggy
- jednoduché řešení - nevytvářet zbytečné settery nebo používat “with” metody



Informační systémy
na míru



Business
analytika



Mobilní
aplikace



Weby
a E-shopy

Imutabilita

Jak to bude vypadat imutabilně?



Imutabilita

```
public function withCash(float $value)
```

```
{
```

```
    $newCash = clone $this;
```

```
    $newCash->value = $value;
```

```
    return $newCash;
```

```
}
```



Imutabilita

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



Výhody

- předvídatelnější, expresivnější kód
- žádná překvapení a side effecty
- eliminace bugů v kódu
- vyšší jistota v kód, rychlejší vývoj
- lepší testovatelnost

Nevýhody

- občas je třeba dát si pozor na performance
- něco nového k naučení



Informační systémy
na míru



Business
analytika



Mobilní
aplikace



Weby
a E-shopy

Díky za pozornost!

Budeme se těšit na příště.



Informační systémy
na míru



Business
analytika



Mobilní
aplikace



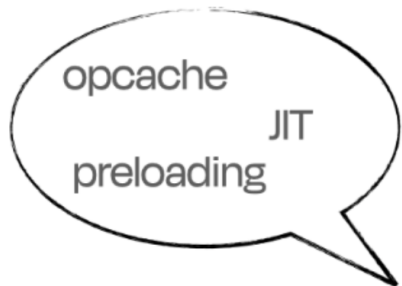
Weby
a E-shopy

Stále nabíráme!

www.rtsoft.cz/kariera

Ladíme běh PHP

pro vysoký výkon



24. 11. 2021

18:00

U Salzmannů, salonek 1. patro



PeoplePath