

# PHPStan の力で Algebraic Data Types を 実現する

nsfisis (いまむら)

第 160 回 PHP 勉強会@東京

# 自己紹介

---

nsfisis (いまむら)



@ デジタルサーカス株式会社

Algebraic Data Type

代数的データ型

すごい enum 型

# ADT の例

---

```
// 注: これは架空の文法です
enum JsonValue {
    case Null;
    case Boolean(bool);
    case Number(int|float);
    case String(string);
    case Array(array);
    case Object(array);
}
```

# ADT の例 (今回扱うもの)

---

```
// 注: これは架空の文法です
enum OptionalInt {
    case None;
    case Some(int);
}
```

# 実現したいもの

---

```
/** @return TODO */  
function f(): TODO {  
    return /* Some(int) か None を返す */;  
}  
  
$x = f();  
if (/* $x が Some かどうか判定する */) {  
    echo /* $x が内部に持っている int の値を「型安全に」取り出す */;  
}
```

# どうやって実現するか？

---

Array shape 編

Object shape 編

Type assertion 編

PHPStan 魔改造編



# Array shape

---

```
$none = [  
  'has_value' => false,  
];  
$some = [  
  'has_value' => true,  
  'value' => 42,  
];
```

# Array shape

---

```
/** @var array{has_value: false} $none */  
$none = [  
    'has_value' => false,  
];  
/** @var array{has_value: true, value: int} $some */  
$some = [  
    'has_value' => true,  
    'value' => 42,  
];
```

# Array shape

---

```
/** @return (array{has_value: false}  
        |array{has_value: true, value: int}) */  
function f(): array { /* 略 */ }
```

# Array shape

---

```
/** @return (array{has_value: false}
           |array{has_value: true, value: int}) */
function f(): array { /* 略 */ }

$x = f();
if ($x['has_value']) {
    // PHPStan は、$x['value'] が int であることを認識できる
    echo $x['value'];
}
```

# Object shape

---

```
/** @var object{has_value: false} $none */  
$none = (object)[  
    'has_value' => false,  
];  
/** @var object{has_value: true, value: int} $some */  
$some = (object)[  
    'has_value' => true,  
    'value' => 42,  
];
```

# Object shape

---

```
/** @return (object{has_value: false}
           |object{has_value: true, value: int}) */
function f(): stdClass { /* 略 */; }

$x = f();
if ($x->has_value) {
    // 型が絞りこまれない！
    echo $x->value;
}
```

# Type assertion

---

```
/** @phpstan-assert-if-true int $a */  
/** @phpstan-assert-if-false !int $a */  
function check_is_int(mixed $a): bool {  
    return is_int($a);  
}  
  
$a = hogehoge();  
if (check_is_int($a)) {  
    PHPStan\dumpType($a);  
    // => int  
}
```

# Type assertion

---

```
abstract class OptionalInt {  
    /**  
     * @phpstan-assert-if-true OptionalIntSome $this  
     * @phpstan-assert-if-false OptionalIntNone $this  
     */  
    public function hasValue(): bool {  
        return $this instanceof OptionalIntSome;  
    }  
}  
class OptionalIntNone extends OptionalInt {}  
class OptionalIntSome extends OptionalInt {  
    public function __construct(public int $value) {}  
}
```



# Type assertion

---

```
$x = f();  
if ($x->hasValue()) {  
    echo $x->value;  
}
```

# Type assertion

---

```
abstract class OptionalInt {  
    /**  
     * @phpstan-assert-if-true OptionalIntSome $this  
     * @phpstan-assert-if-false OptionalIntNone $this  
     */  
    public function hasValue(): bool {  
        return $this instanceof OptionalIntSome;  
    }  
}  
class OptionalIntNone extends OptionalInt {}  
class OptionalIntSome extends OptionalInt {  
    public function __construct(public int $value) {}  
}
```

# ここまでのまとめ

---

Array ではうまく動く

Object では動かない / 課題アリ

Object でも型を絞り込みたい

# PHPStan 魔改造

---

```
if ($x['has_value']) {  
    // $x['has_value'] の型が true に確定したことで、  
    // $x に HasOffsetValueType('has_value', true) の型が付く  
    echo $x['value'];  
}
```

# PHPStan 魔改造

```
if ($x['has_value']) {  
    // $x['has_value'] の型が true に確定したことで、  
    // $x に HasOffsetValueType('has_value', true) の型が付く  
    echo $x['value'];  
}
```

```
// (array{has_value: false}  
//   |array{has_value: true, value: int})  
// と  
// HasOffsetValueType('has_value', true)  
// とを合成する
```

HasPropertyValue

HasPropertyValue と object shape の合成

内から外への型の伝播

# まとめ

---

PHPStan と array shape を使った ADT もどき

Object の型の絞り込みは未実装

現在鋭意実装中