REFACTORING

Avagy a forráskód minőségének javítása.

Szabó Gábor
nexum Magyarország Kft.

Témák

A PROBLÉMA

A MEGOLDÁS

(PHP példák)

AZ EREDMÉNY

A PROBLÉMA

- *nehezen* érthető algoritmus,
- az egyszerűnek tűnő feladatok *sok időt* igényelnek,
- *nehezen* tesztelhető működés,
- a forrás *nem újrahasznosítható*,
- stb.

BONYOLULT FORRÁSKÓD

MI A MEGOLDÁS?

Megoldás keresése, ami

- *könnyen* elsajátítható,
- *rutin-átalakítások*kal,
- *javítja* a kód belső szerkezetét.

REFACTORING

EGYSZERŰBB FORRÁSKÓD

Mi a refactoring?

Folyamat, melynek során **javul a kód belső szerkezete**, de **nem módosul a kifelé mutatott viselkedése**.

Nyelv-független módszerek.
(A példákban PHP alapú minták.)

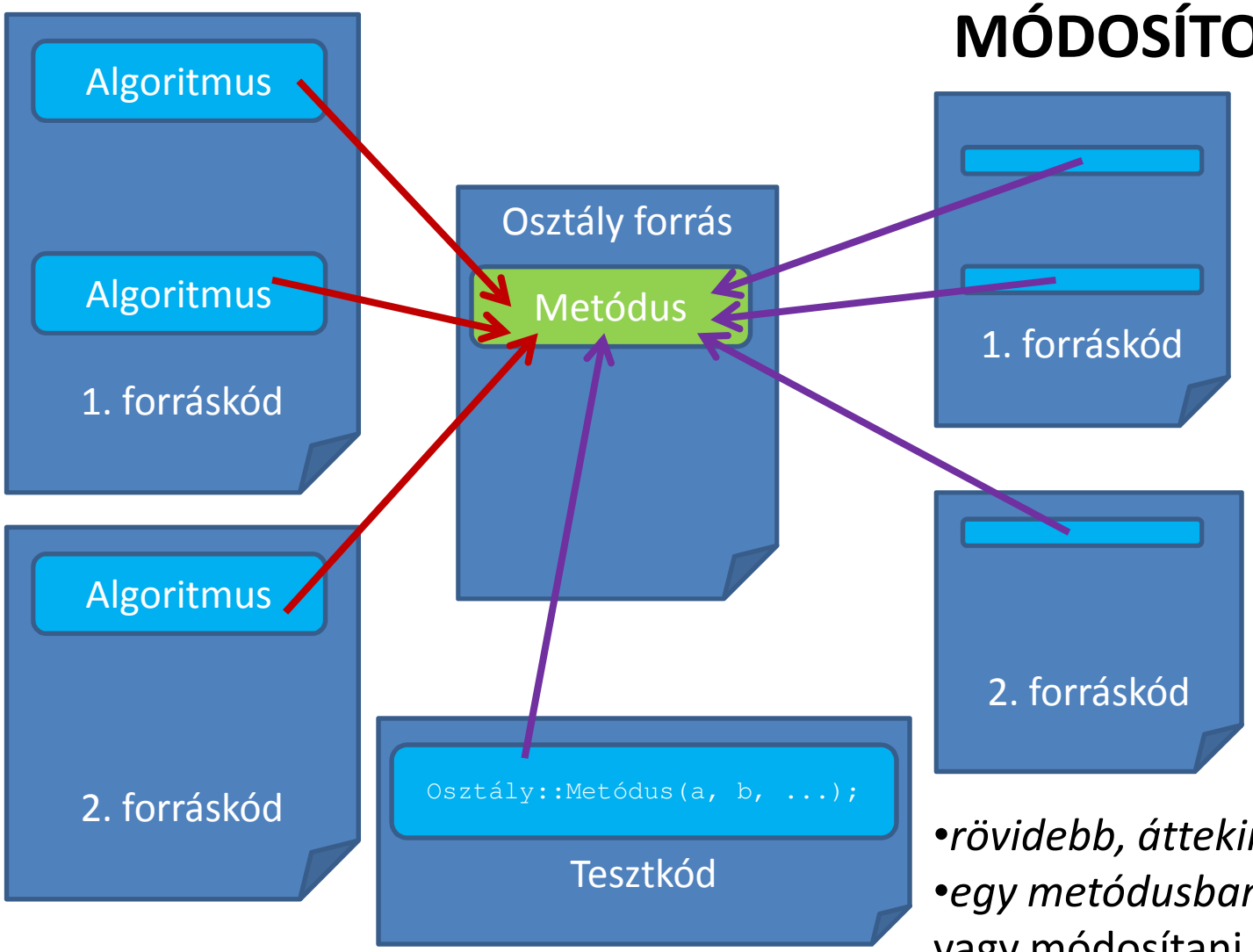
A probléma felismerése

- *kódismétlés*,
- *nagyméretű* és/vagy *több feladatot ellátó* algoritmusrészlet (eltérő elvonatkoztatási szint),
- *ugyanolyan jellegű feladatok* más-más osztályokban vagy, egy osztály *több másik osztály feladatát* is végzi,
- stb.

Martin Fowler:

„GYANÚS SZAGÚ” KÓD

A kódismétlés megszüntetése

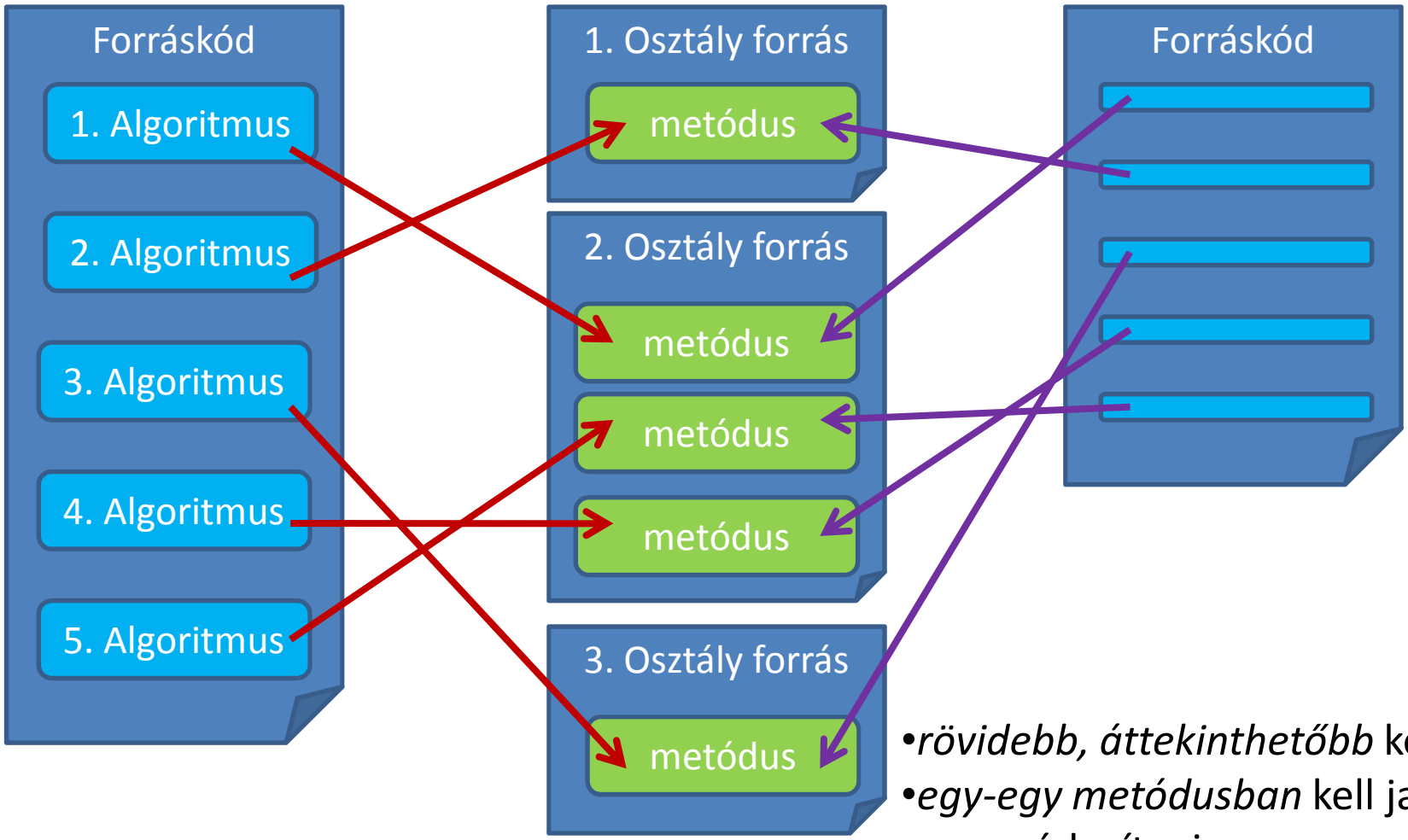


- *rövidebb, áttekinthetőbb* forrás,
- *egy metódusban* kell javítani, vagy módosítani,
- *elég egy* metódust tesztelni.

PÉLDA

Bonyolult, több feladatot ellátó kód

MÓDOSÍTOTT KÓD



- *rövidebb, áttekinthetőbb* kód,
- *egy-egy metódusban* kell javítani, vagy módosítani,
- *elég egy-egy* metódust tesztelni,
- *új szolgáltatások*.

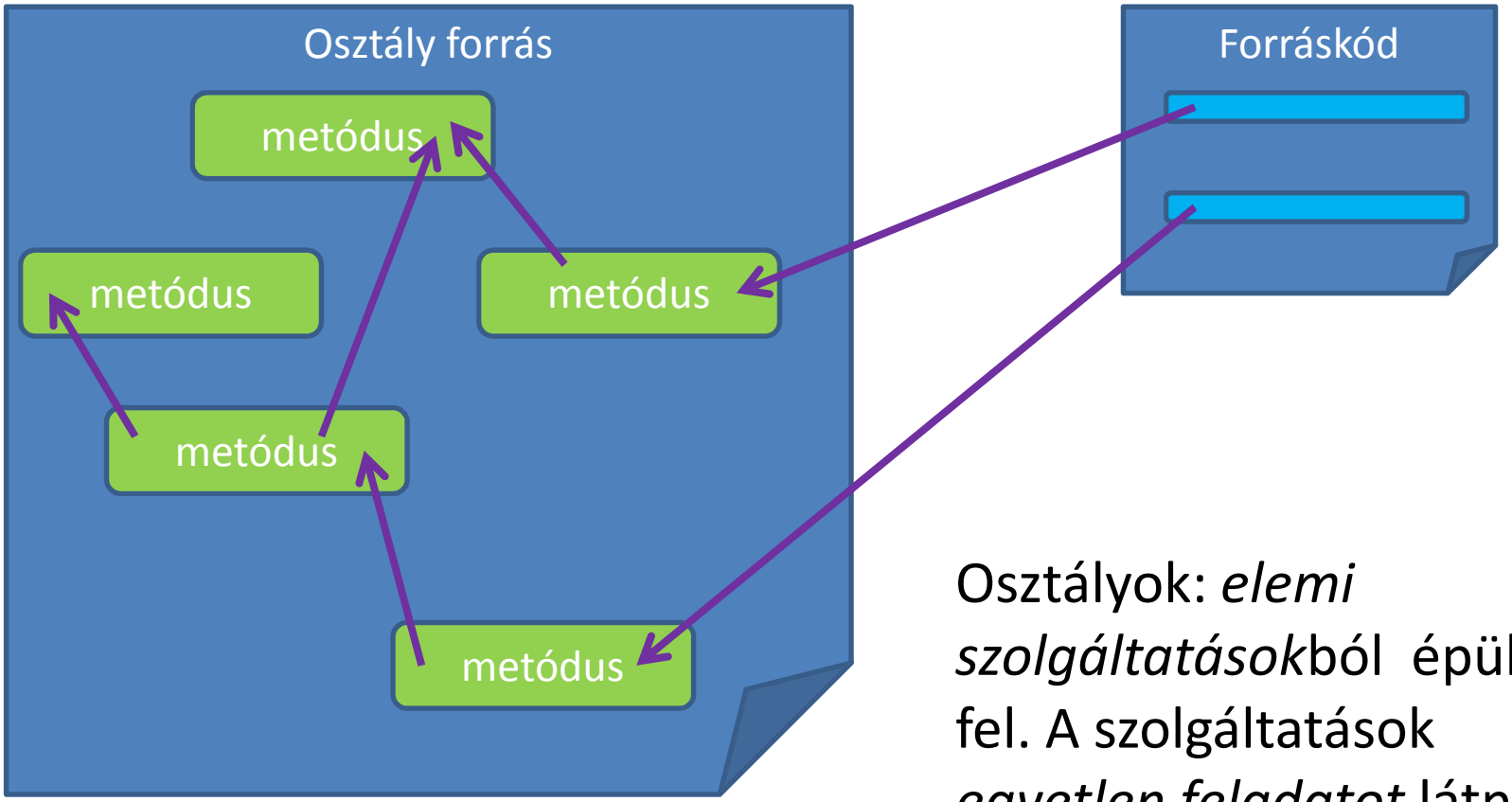
PÉLDA

A refactoring hátrányai

- *nagyobb méretű* osztályok (sok kis metódus),
- *összetettebb* hívási sorok (kis szolgáltatások),
- kezdetben *több erőforrást* igényel a fejlesztők részéről,
- *gyakorlat* kell az alkalmazásához.

Egymásra épülő funkciók

(A szolgáltatások bővítése a metódusok egymásra építésével.)



„metódus = szolgáltatás”

Osztályok: *elemi szolgáltatások*ból épülnek fel. A szolgáltatások *egyetlen feladatot* látnak el, a lehető *legjobban*.

PÉLDA

Egymásra épülő funkciók

Webáruház

http://develop.blog.hu/2010/07/30/egymasra_epulo_funkciok_egyszerusitese

PÉLDA

Termékár feltüntetése (több oldalon / pozícióban)

(Nettó ár + 25%-os ÁFA = Bruttó ár)

”12 345 + 3 086 = 15 431”

```php
$priceNet = 12345;
$priceNetFormat = number_format( $priceNet, 0, ',', ' ' );

$priceVat = $priceNet * 0.25;
$priceVatFormat = number_format( $priceVat, 0, ',', ' ' );

$priceTotal = $priceNet * 1.25;
$priceTotalFormat = number_format( $priceTotal, 0, ',', ' ' );

// A továbbiakban ezt elhagyom
echo $priceNetFormat.' + '.$priceVatFormat.' = '.$priceTotalFormat;
```

PÉLDA

```php
// Előző állapot

$priceNetFormat = number_format( $priceNet, 0, ',', ' ' );

$priceVat = $priceNet * 0.25;
$priceVatFormat = number_format( $priceVat, 0, ',', ' ' );

$priceTotal = $priceNet * 1.25;
$priceTotalFormat = number_format( $priceTotal, 0, ',', ' ' );
```

```php
function numberFormat( $value ) {
  return number_format( $value, 0, ',', ' ' );
}

$tax = 0.25;
$priceNet = 12345;
$priceNetFormat = numberFormat( $priceNet );

$priceVat = $priceNet * $tax;
$priceVatFormat = numberFormat( $priceVat );

$priceTotal = $priceNet * ( 1 + $tax );
$priceTotalFormat = numberFormat( $priceTotal );
```

PÉLDA

```
$priceNetFormat = numberFormat( $priceNet );

$priceVat = $priceNet * $tax;
$priceVatFormat = numberFormat( $priceVat );

$priceTotal = $priceNet * ( 1 + $tax );
$priceTotalFormat = numberFormat( $priceTotal );
```

```
function getPriceVat( $priceNet, $tax ) {
  return $priceNet * $tax;
}

function getPriceTotal( $priceNet, $tax ) {
  return $priceNet * ( 1 + $tax );
}

$tax = 0.25;
$priceNet = 12345;
$priceNetFormat = numberFormat( $priceNet, $tax );

$priceVat = getPriceVat( $priceNet, $tax );
$priceVatFormat = numberFormat( $priceVat );

$priceTotal = getPriceTotal( $priceNet, $tax );
$priceTotalFormat = numberFormat( $priceTotal );
```

PÉLDA

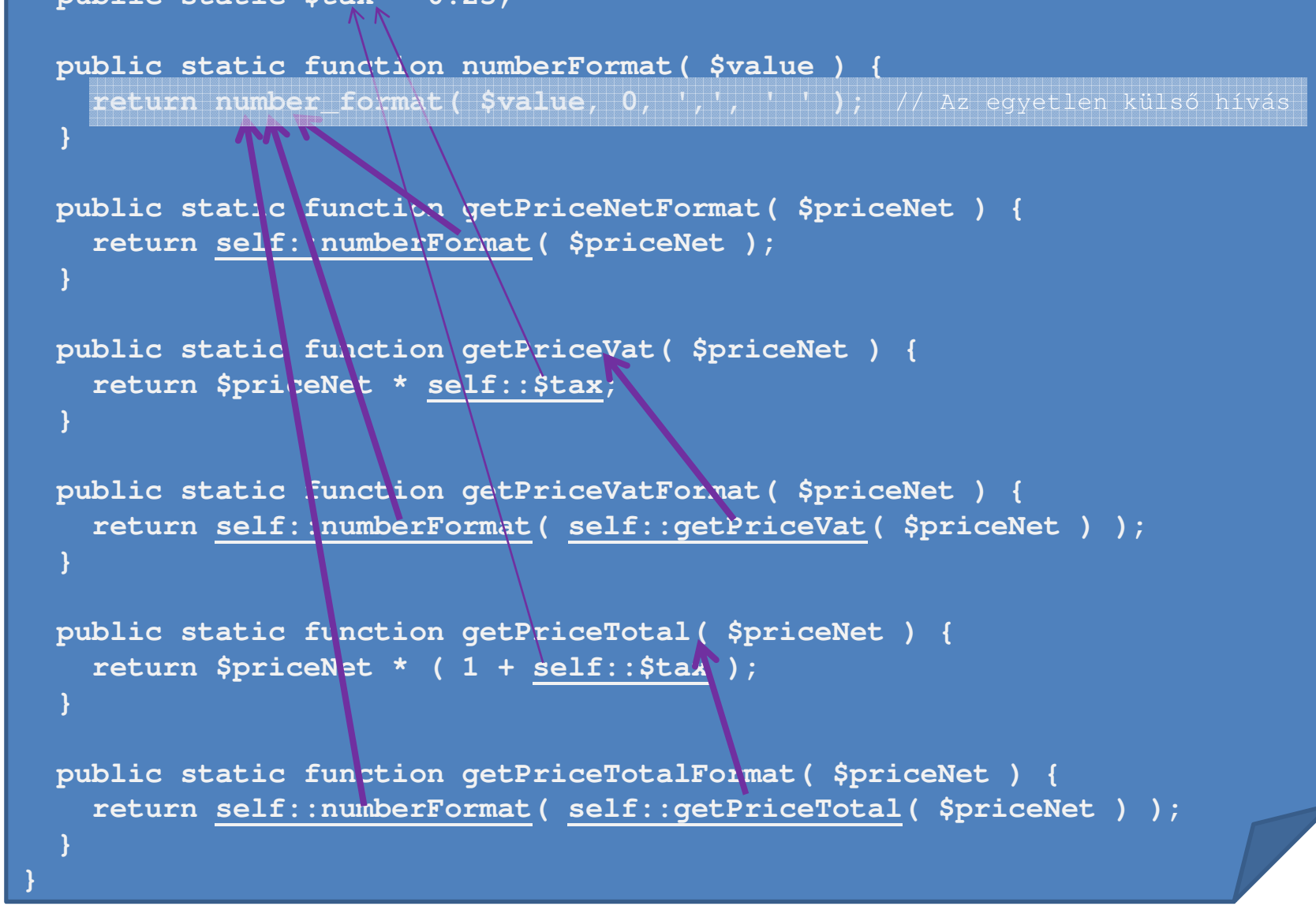
```php
function getPriceVat( $priceNet, $tax ) {
  return $priceNet * $tax;
}

function getPriceTotal( $priceNet, $tax ) {
  return $priceNet * ( 1 + $tax );
}
```

```php
function getPriceNetFormat( $priceNet, $tax ) {
  return numberFormat( $priceNet );
}

function getPriceVatFormat( $priceNet, $tax ) {
  return numberFormat( getPriceVat( $priceNet, $tax ) );
}

function getPriceTotalFormat( $priceNet, $tax ) {
  return numberFormat( getPriceTotal( $priceNet, $tax ) );
}

$tax = 0.25;
$priceNet = 12345;
$priceNetFormat   = getPriceNetFormat( $priceNet, $tax );
$priceVatFormat   = getPriceVatFormat( $priceNet, $tax );
$priceTotalFormat = getPriceTotalFormat( $priceNet, $tax );
```

PÉLDA

```php
$priceNet = 12345;
$priceNetFormat   = price::getPriceNetFormat( $priceNet );
$priceVatFormat   = price::getPriceVatFormat( $priceNet );
$priceTotalFormat = price::getPriceTotalFormat( $priceNet );
```

```php
class price {
  public static $tax = 0.25;

  public static function numberFormat( $value ) {
    return number_format( $value, 0, ',', ' ' );  // Az egyetlen külső hívás
  }

  public static function getPriceNetFormat( $priceNet ) {
    return self::numberFormat( $priceNet );
  }

  public static function getPriceVat( $priceNet ) {
    return $priceNet * self::$tax;
  }

  public static function getPriceVatFormat( $priceNet ) {
    return self::numberFormat( self::getPriceVat( $priceNet ) );
  }

  public static function getPriceTotal( $priceNet ) {
    return $priceNet * ( 1 + self::$tax );
  }

  public static function getPriceTotalFormat( $priceNet ) {
    return self::numberFormat( self::getPriceTotal( $priceNet ) );
  }
}
```



PÉLDA

Konklúzió

```
class price {
  public static $tax = 0.25;

  public static function numberFormat( $value ) {
    return number_format( $value, 0, ',', ' ' );
  }

  public static function getPriceNetFormat( $priceNet ) {
    return self::numberFormat( $priceNet );
  }

  public static function getPriceVat( $priceNet ) {
    return $priceNet * self::$tax;
  }

  public static function getPriceVatFormat( $priceNet ) {
    return self::numberFormat( self::getPriceVat( $priceNet ) );
  }

  public static function getPriceTotal( $priceNet ) {
    return $priceNet * ( 1 + self::$tax );
  }

  public static function getPriceTotalFormat( $priceNet ) {
    return self::numberFormat( self::getPriceTotal( $priceNet ) );
  }
}
```

```
$priceNet = 12345;
$priceNetFormat   = price::getPriceNetFormat( $priceNet );
$priceVatFormat   = price::getPriceVatFormat( $priceNet );
$priceTotalFormat = price::getPriceTotalFormat( $priceNet );
```

Az utolsó változat

- áttekinthető,
- könnyen módosítható,
- könnyen tesztelhető,
- nincs függőség a külvilágtól,
- beszédes hívások,
- egységes ár-megjelenés.

PÉLDA

Módosuló igények

Eredeti megjelenítés

”12 345 + 3 086 = 15 431”

(Nettó ár + 25%-os ÁFA = Bruttó ár)

- ÁFA-változás,
- 1000-es elválasztó módosítása,
- tizedesek számának módosítása,
- tizedes elválasztó módosítása.

```
number_format( $value, 0, ',', ' ' )
```

Továbbra is egységes az ár megjelenítése.

```
class price {
  public static $tax = 0.25;
  private static $decimals = 0;
  private static $decimalPoint = ',';
  private static $thousandSeparator = ' ';

  public static function setTax( $value ) {
    self::$tax = $value;
  }

  public static function setDecimals( $value ) {
    self::$decimals = $value;
  }

  public static function setDecimalPoint( $value ) {
    self::$decimalPoint = $value;
  }

  public static function setThousandSeparator( $value ) {
    self::$thousandSeparator = $value;
  }
  ...
  public static function numberFormat( $value ) {
    return number_format( $value, self::$decimals,
      self::$decimalPoint,
      self::$thousandSeparator );
  }
}
```

```
public static function numberFormat( $value ) {
    return number_format( $value, 0, ',', ' ' );
}
```

```
price::setTax( 0.2 );
price::setDecimals( 2 );

$priceNetFormat   = price::getPriceNetFormat( $priceNet );
$priceVatFormat   = price::getPriceVatFormat( $priceNet );
$priceTotalFormat = price::getPriceTotalFormat( $priceNet );
```

Feltételes kifejezések egyszerűsítése

Feladat-kezelő alkalmazás

http://develop.blog.hu/2010/07/28/felteteles_kifejezesek_egyszerusitese

PÉLDA

```
...
// Van nyitott feladat
if( task::getNumberOfTaskOpen() != 0 ) { ... }
...
```

```
...
// Nincs nyitott feladat
if( task::getNumberOfTaskOpen() == 0 ) { ... }
...
```

PÉLDA

```
...
// Van nyitott feladat
if( task::getNumberOfTaskOpen() != 0 ) { ... }

...

// Nincs nyitott feladat
if( task::getNumberOfTaskOpen() == 0 ) { ... }
```

```
...
$numberOfTaskOpen = task::getNumberOfTaskOpen();

// Van nyitott feladat
if( $numberOfTaskOpen != 0 ) { ... }

...

// Nincs nyitott feladat
if( $numberOfTaskOpen == 0 ) { ... }
```

PÉLDA

```
...
$numberOfTaskOpen = task::getNumberOfTaskOpen();

// Van nyitott feladat
if( $numberOfTaskOpen != 0 ) { ... }

...

// Nincs nyitott feladat
if( $numberOfTaskOpen == 0 ) { ... }
```

```
...
$numberOfTaskOpen = task::getNumberOfTaskOpen();

// Van nyitott feladat
if( 0 < $numberOfTaskOpen ) { ... }

...

// Nincs nyitott feladat
if( !( 0 < $numberOfTaskOpen ) ) { ... }
```

PÉLDA

```
$numberOfTaskOpen = task::getNumberOfTaskOpen();

// Van nyitott feladat
if( 0 < $numberOfTaskOpen ) { ... }

...

// Nincs nyitott feladat
if( !( 0 < $numberOfTaskOpen ) ) { ... }
```

```
function isTaskOpen() {
  if( 0 < task::getNumberOfTaskOpen() ) {
    return true;
  }
  return false;
}

// Van nyitott feladat
if( isTaskOpen() ) { ... }

...

// Nincs nyitott feladat
if( !isTaskOpen() ) { ... }
```

PÉLDA

```
if( isTaskOpen() ) { ... }
...
if( !isTaskOpen() ) { ... }
```

```
class task {
  public static function isTaskOpen() {
    if( 0 < self::getNumberOfTaskOpen() ) {
      return true;
    }
    return false;
  }

  public static function getNumberOfTaskOpen() { ... }
}

// Van nyitott feladat
if( task::isTaskOpen() ) { ... }

...

// Nincs nyitott feladat
if( !task::isTaskOpen() ) { ... }
```

PÉLDA

```
if( task::isTaskOpen() ) { ... }
...
if( !task::isTaskOpen() ) { ... }
```

```
class task {
  public static function isTaskOpen() {
    if( 0 < self::getNumberOfTaskOpen() ) {
      return true;
    }
    return false;
  }

  public static function isNotOpenTask() {
    return !self::isOpenTask();
  }

  public static function getNumberOfTaskOpen() { ... }

}

if( task::isTaskOpen() ) { ... }
...
if( task::isNotTaskOpen() ) { ... }
```

PÉLDA

Összetettebb eset

```
...
if( A && B || !C  && D || E ) { ... }
...
```

```
...
if( A && B || !C  && D || E ) { ... }
...
```

```
function feltetel() {
        return A && B || !C  && D || E;
}

if( feltetel() ) { ... }
```

```
...
if( isPost() && isPostValid() && post( „data” ) == „test” )
{
...
} elseif( isPost() && !isPostValid() ) {
  // Nem valid a POST
} elseif( !isPost() && isGet() && isGetValid() ...
```

PÉLDA

A paraméterátadás egyszerűsítése

Képfeliratozó osztály

(a paraméterek számának csökkentése)

http://develop.blog.hu/2011/02/05/parameteratadas_egyszerusitese_a_parameterek_szamanak_csokkentesevel

```
$image = imagecreatefromjpeg( 'c:\mintakep.jpg' );
$colorWhite = imagecolorallocate( $image, 255, 255, 255 );

$font = 'c:\arial.ttf‘;
$text = 'árvíztűrő tükörfúrógép ÁRVÍZTŰRŐ TÜKÖRFÚRÓGÉP‘;

imagettftext( $image, 12, 0, 4, 16, $colorWhite, $font, $text );

imagejpeg( $image,'c:\mintakep_output.jpg' );
```

PÉLDA

```
$image = imagecreatefromjpeg( 'c:\mintakep.jpg' );
$colorWhite = imagecolorallocate( $image, 255, 255, 255 );

$font = 'c:\arial.ttf‘;
$text = 'árvíztűrő tükörfúrógép ÁRVÍZTŰRŐ TÜKÖRFÚRÓGÉP‘;

imagettftext( $image, 12, 0, 4, 16, $colorWhite, $font, $text );

imagejpeg( $image,'c:\mintakep_output.jpg' );
```

```
class myImage {
  public function write( &$image, $size, $angle, $x, $y, $color, $fontfile, $text ) {
    imagettftext( $image , $size, $angle, $x, $y, $color, $fontfile, $text );
  }
}

$myImage = new myImage();

$image = imagecreatefromjpeg( 'c:\mintakep.jpg' );
$colorWhite = imagecolorallocate( $image, 255, 255, 255 );

$fontfile = 'c:\arial.ttf‘;
$text = 'árvíztűrő tükörfúrógép ÁRVÍZTŰRŐ TÜKÖRFÚRÓGÉP‘;

$myImage -> write( $image, 12, 0, 4, 16, $colorWhite, $fontfile, $text );

imagejpeg( $image, 'c:\mintakep_output.jpg' );
```

PÉLDA

```
class myImage {
  public function write( &$image, $size, $angle, $x, $y, $color, $fontfile, $text ) {
    imagettftext( $image , $size, $angle, $x, $y, $color, $fontfile, $text );
  }
}

$fontfile = ‘c:\arial.ttf’;

...
$myImage -> write( $image, 12, 0, 4, 16, $colorWhite, $fontfile, $text );
...
```

```
class myImage {
  const fontfile = 'c:\arial.ttf';
  public function write( &$image, $size, $angle, $x, $y, $color, $text ) {
    imagettftext( $image , $size, $angle, $x, $y, $color, self::fontfile, $text );
  }
}

$myImage = new myImage();

$image = imagecreatefromjpeg( 'c:\mintakep.jpg' );
$colorWhite = imagecolorallocate( $image, 255, 255, 255 );

$text = 'árvíztűrő tükörfúrógép ÁRVÍZTŰRŐ TÜKÖRFÚRÓGÉP‘;

$myImage -> write( $image, 12, 0, 4, 16, $colorWhite, $text );

imagejpeg( $image,'c:\mintakep_output.jpg' );
```

PÉLDA

```
class myImage {
  const fontfile = 'c:\arial.ttf';
  public function write( &$image, $size, $angle, $x, $y, $color, $text ) {
    imagettftext( $image , $size, $angle, $x, $y, $color, self::fontfile, $text
);
  }
}

...
$myImage -> write( $image, 12, 0, 4, 16, $colorWhite, $text );
...
```

```
class myImage {
  const fontfile = 'c:\arial.ttf';
  public $image = null;
  public $color = null;
  public $angle = 0;
  protected $imageFilePathSource = null;

  function __construct( $imageFilePathSource ) {
    $this -> imageFilePathSource = $imageFilePathSource;
    $this -> image = imagecreatefromjpeg( $this -> imageFilePathSource );
    $this -> color = imagecolorallocate( $this -> image, 255, 255, 255 );
  }

  public function write( $size, $x, $y, $text ) {
    imagettftext($this -> image, $size, $this -> angle, $x, $y, $this -> color, self::fontfile, $text);
  }
}

$myImage = new myImage( 'c:\mintakep.jpg' );

$text = 'árvíztűrő tükörfúrógép ÁRVÍZTŰRŐ TÜKÖRFÚRÓGÉP‘;
$myImage -> write( 12, 4, 16, $text );

imagejpeg( $myImage -> image,'c:\mintakep_output.jpg' );
```

```
class myImage {
  public function write( &$image, $size, $x, $y, $text ) {
    imagettftext($image , $size, $this -> angle, $x, $y, $this -> color, self::fontfile, $text);
  }
}

...
$text = 'árvíztűrő tükörfúrógép ÁRVÍZTŰRŐ TÜKÖRFÚRÓGÉP‘;
$myImage -> write( 12, 4, 16, $text );
...
```

```
class myImage {
  ...
  const sizeDefault = 12;
  public $size = self::sizeDefault;

  function __construct( $imageFilePathSource, $size = null ) {
    ...

    if( !is_null( $size ) ) {
     $this -> size = $size;
    }
  }

  function write($x, $y, $text ) {
    imagettftext($this->image, $this->size, $this->angle, $x, $y, $this->color, self::fontfile, $text);
  }
}

$myImage = new myImage( 'c:\mintakep.jpg', 12 );

$myImage -> write(4, 16, 'árvíztűrő tükörfúrógép ÁRVÍZTŰRŐ TÜKÖRFÚRÓGÉP' );

imagejpeg( $myImage -> image,'c:\mintakep_output.jpg' );
```

```
class myImage {
...
}

$myImage = new myImage( 'c:\mintakep.jpg', 12 );

$myImage -> write(4, 16, 'árvíztűrő tükörfúrógép ÁRVÍZTŰRŐ TÜKÖRFÚRÓGÉP' );

imagejpeg( $myImage -> image,'c:\mintakep_output.jpg' );
```

```
class myImage {
  ...
  public function save( $imageFilePathDestination ) {
    imagejpeg( $this -> image, $imageFilePathDestination );
  }
}

$myImage = new myImage( 'c:\mintakep.jpg', 12 );

$myImage -> write( 4, 16, 'árvíztűrő tükörfúrógép ÁRVÍZTŰRŐ TÜKÖRFÚRÓGÉP' );

$myImage -> save( 'c:\mintakep_output.jpg' );
```

```
class myImage {
...
}

$myImage = new myImage( 'c:\mintakep.jpg', 12 );

$myImage -> write(4, 16, 'árvíztűrő tükörfúrógép ÁRVÍZTŰRŐ TÜKÖRFÚRÓGÉP' );

imagejpeg( $myImage -> image,'c:\mintakep_output.jpg' );
```

```
class myImage {
  ...
  protected function write($x, $y, $text ) {
    imagettftext($this->image, $this->size, $this->angle, $x, $y, $this->color, self::fontfile, $text);
  }
  ...
  public function writeLeftTop( $text, $delta = 0 ) {
    $this -> write( $delta, $this -> size + $delta, $text );
  }
}

$myImage = new myImage( 'c:\mintakep.jpg', 12 );

$myImage -> writeLeftTop( 'árvíztűrő tükörfúrógép ÁRVÍZTŰRŐ TÜKÖRFÚRÓGÉP', 4 );

$myImage -> save( 'c:\mintakep_output.jpg' );
```

EREDMÉNY

Miért is van szükségünk a refactoringra?

REFACTORING (ISMÉTLÉS)

- *áttekinthető*, *könnyen* érthető kód,
- *gyorsan* elvégezhető hibajavítási folyamatok,
- *rugalmasan* módosítható, *többször felhasználható* kód.

EGYSZERŰBB FORRÁSKÓD

Egy átlagos projekt

Az ügyfél elképzelései

Üzleti igény

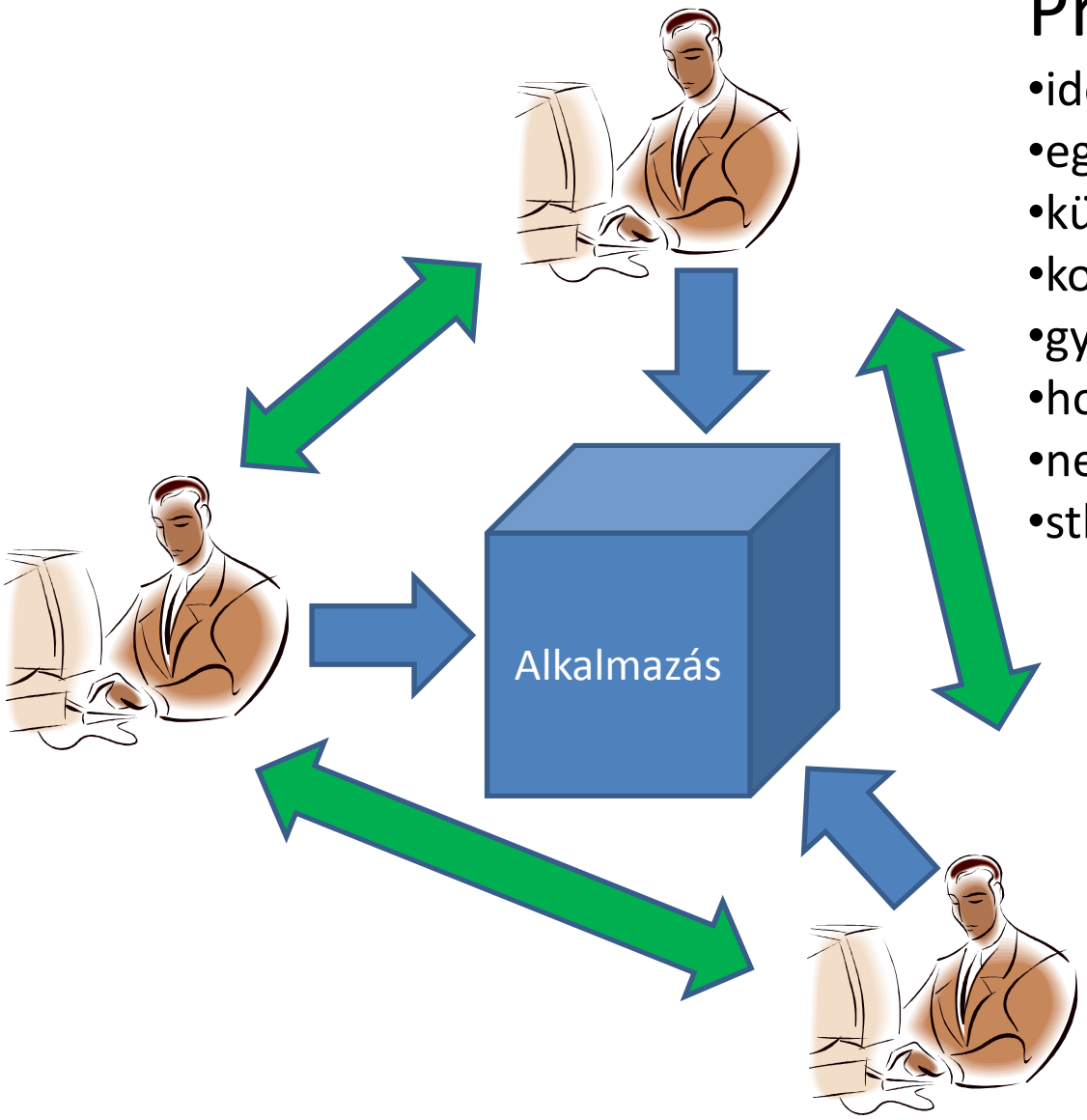
Határidő

Költség

Az ügyfél

AZ EREDMÉNY

Igények kiszolgálása (céges fejlesztés; csapat)



Problémák lehetnek

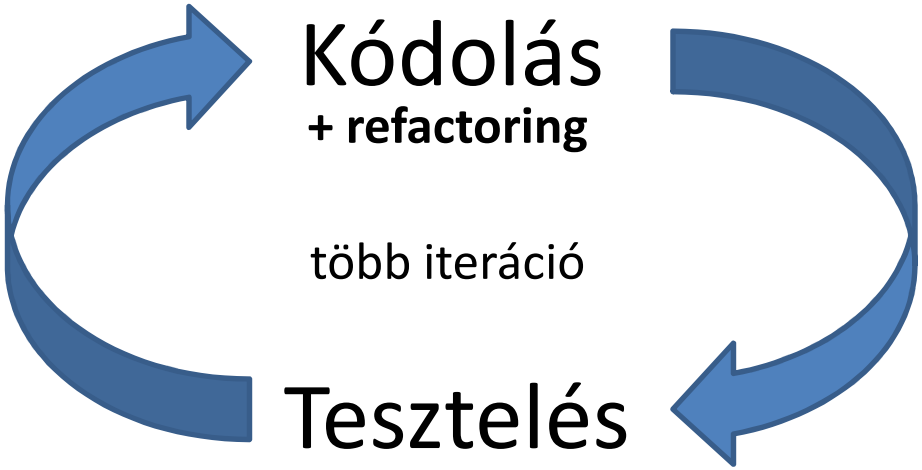
- időben eltérő fejlesztési fázisok,
- egymással összefüggő szolgáltatások,
- különféle gondolkodásmódok,
- kommunikációs nehézségek,
- gyakorlat/rutin hiánya,
- hosszú válaszidők,
- nehezebb erőforrás-elosztás,
- stb.

CÉL

Fejlesztési költségek csökkentése!

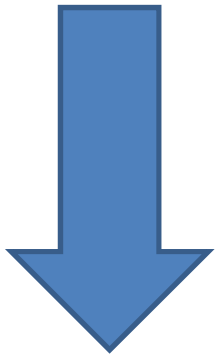
AZ EREDMÉNY

A programozás javasolt folyamata

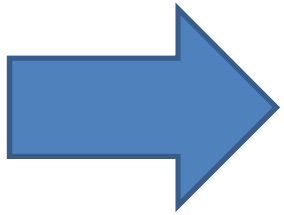


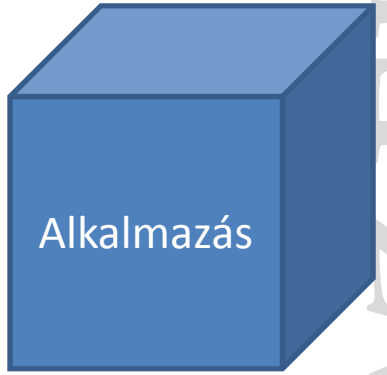
- a kód módosítása *apró lépésekkel*,
- az átalakítások *következetesek*,
- *könnyen* tesztelhető kód.

Ha hibára futunk az utolsó átalakítások valamelyike hibás, ami *könnyen megtalálható, javítható*.



Elkészült algoritmusok





A refactoring „hozadéka”

- *egyszerűbb*, érthetőbb kód,
- *gyorsabb* fejlesztés (szolgáltatások),
- *kisebb* kódméret (szolgáltatások felhasználása),
- *könnyebb* tesztelhetőség (kis feladatot ellátó metódusok),
- *önmagát dokumentáló* kód (beszédes hívások).

Kisebb erőforrásigény.

rövidebb kivitelezési idő,
csökkenő fejlesztési költségek,
piacképes óradíjak

Osztálykönyvtárak alkalmazása.

Újrahasznosítható kód

http://develop.blog.hu/2011/01/25/ceges_tudasbazis_osztalykonyvtarak_epitese

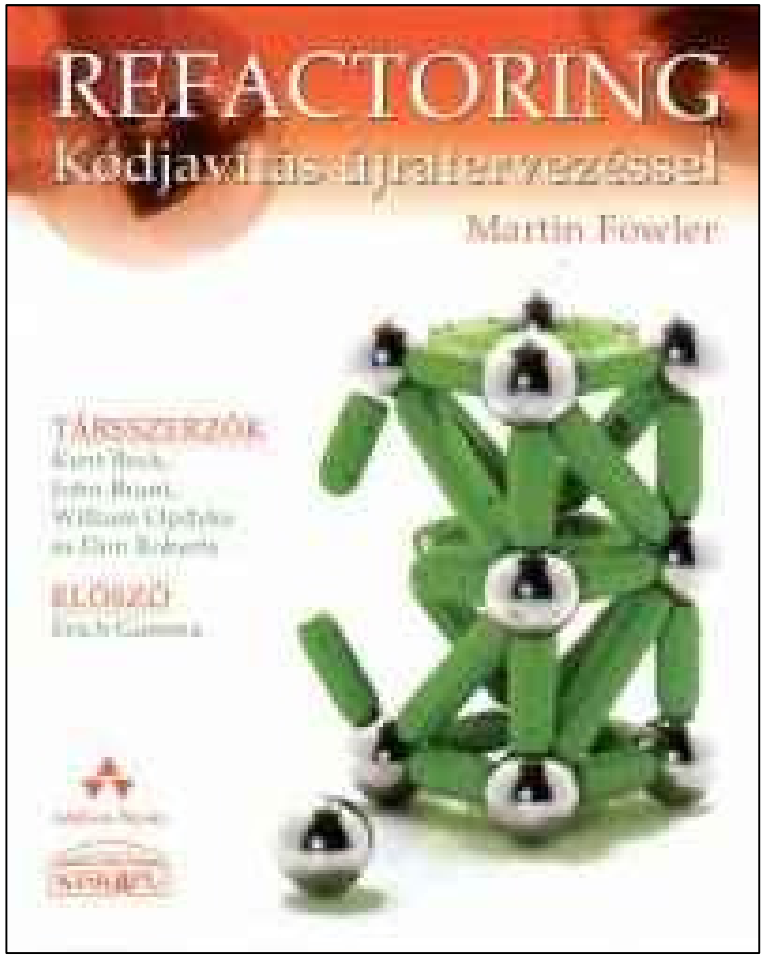
A refactoring „hozadéka”

- *gyorsabb* fejlesztési ciklusok
- *rövidebb* válaszidők,
- *kisebb* költség,
- *hosszú távú* kapcsolat.

ELÉGEDETT ÜGYFÉL

Szakirodalom

Martin Fowler: Refactoring



Robert C. Martin: Tiszta kód

... amiről talán legközelebb ☺

- elnevezési ajánlások,
- hierarchikus nevezéktan,
- objektumparaméter használata,
- stb.

http://develop.blog.hu

Köszönöm a figyelmet

szabo@nexum.hu

http://develop.blog.hu

Szabó Gábor
nexum Magyarország Kft.