Writing testable code in PHP*

by Andrei Kojusko

* Can be used with any language

About me

Andrei Kojusko, Moldova, Chisinau

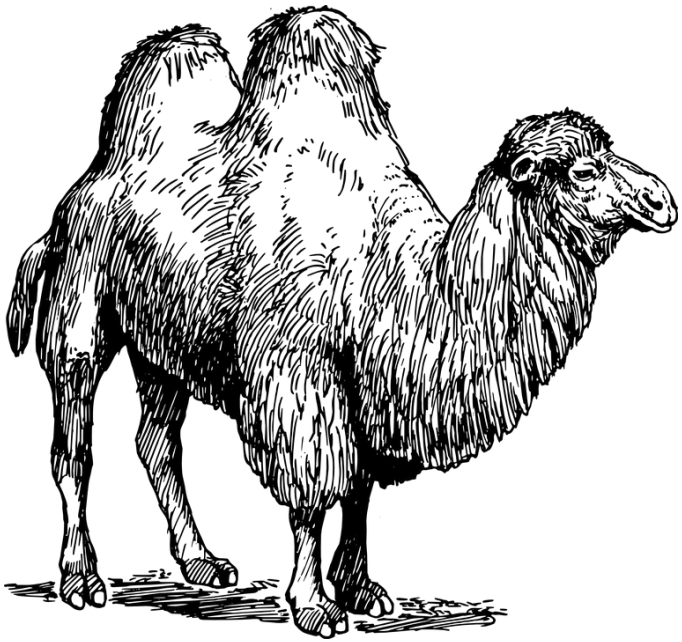
Coding from ‘93
Coding professionally from ‘02
In Pentalog from ‘08

Prerequisites

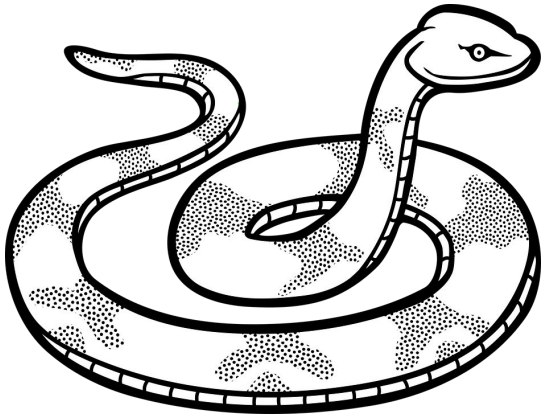
- OOP
- Some PHP
- Some architecture
- Some unit testing

Real project example

iLikeCamel



i_like_snake



Why bother?

- Bugs
- Real world vs. Ideal code
- Detect breaking changes
- Automation
- Timesaving
- Show seniority

What could go wrong?

```php
public function createAdmin(string $login, string $password) : int
{
  $user = new User();
  $user->setLogin($login);
  $user->setPassword($password);
  $user->addRole(Role::ADMIN);

  $this->repository->persist($user);

  return $user->getId();
}
```

The unit test

```php
public function testCreateAdminValid()
{
  $manager = $this->getManager();
  $return = $manager->createAdmin('admin', 'admin');
  $this->assertEquals(1, $return);
}

public function testCreateAdminInvalid()
{
  $this->expectException(InvalidArgumentException::class);
  $this->getManager()->createAdmin('', '');
}
```

How does it look?

```
> phpunit --verbose ManagerTest
PHPUnit 6.0.0 by Sebastian Bergmann and contributors.

..

Time: 0 seconds, Memory: 3.25Mb

OK (2 tests, 2 assertions)
```

Continuous integration

- **Local execution**
- **Build server execution**
- **GitHub, etc.**

Mock objects

```
class Manager {
  public function update() {
    return $this->db->update();
  }
}


class ManagerMock extends Manager {
  public function update() {
    return true;
  }
}
```

Problematic code

- Too many dependencies
- Hidden dependencies
- Heavy initializers
- Complicated logic
- Long distance calls
- Hidden inputs
- Side effects

Too many dependencies

```
class UserActivityManager {
  public function addFriend() {
    User::update();
    Database::save();
    FriendRequest::send();
    Mailer::send();
  }
  public function addImage() {
    Filesystem::read();
    Image::crop();
    Gallery::update();
    Database::save();
    WallPost::add();
  }
}
```

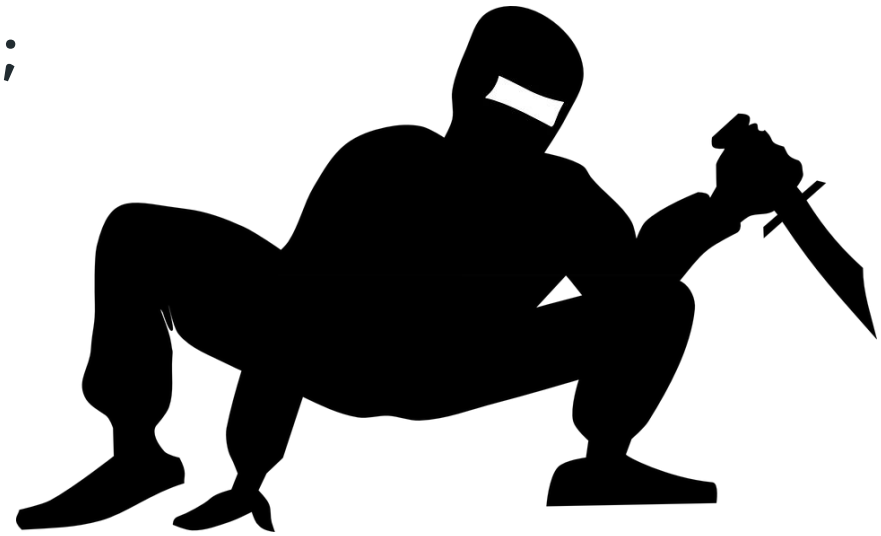
Too many dependencies - solution

```
class FriendManager {
  public function addFriend()
  {
    User::update();
    Database::save();
    FriendRequest::send();
    Mailer::send();
  }
}

class GalleryManager {
  public function addImage() {
    Filesystem::read();
    Image::crop();
    Gallery::update();
    Database::save();
  }
}
```

Hidden dependencies

```
class UserManager {
  public function updateUser(int $userId, array $data) {
    $db = new Connection();
    $repo = new Repository($db, 'user');
    /* ... */
  }
}
```



Hidden dependencies - solution

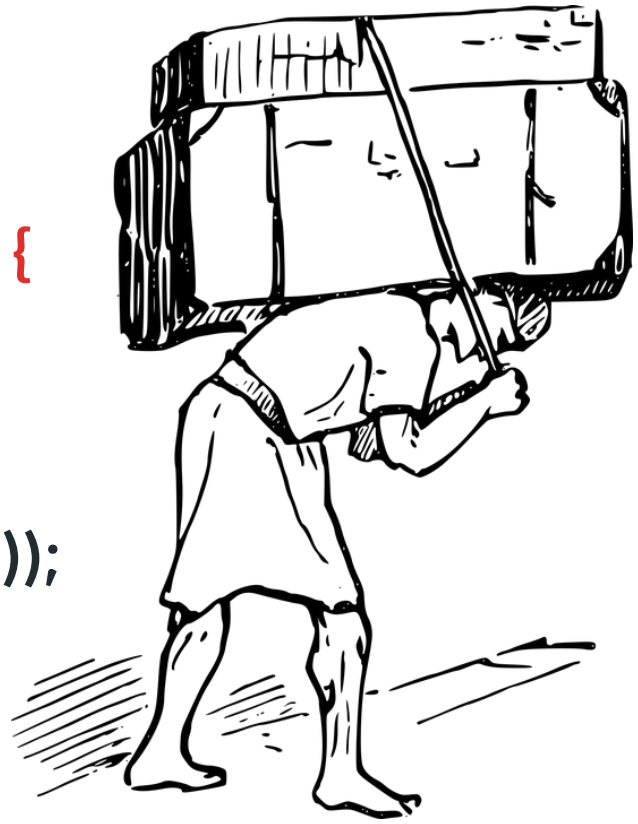
```
class UserManager {
  /** @var Repository */
  public $repository;

  public function __construct(Connection $connection) {
    $this->repository = new Repository($connection, 'user');
  }

  public function updateUser(int $userId, array $data) {
    /* ... */
    $this->repository->update();
  }
}
```

Heavy initializers

```
class StationManager {
  public function __construct(Station $station) {
     $this->db = Connection::getInstance();
     $this->mailer = new Mailer();
     $this->userManager =
          new UserManager($station->getOwner());
  }
}
```

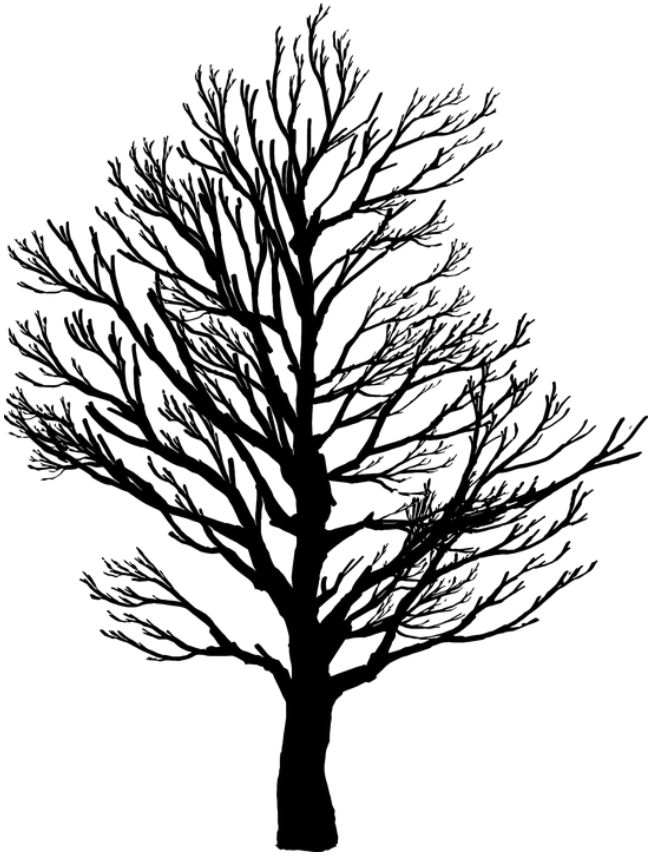


Heavy initializers - solution

```
class StationManager {
  public function __construct(
      Connection $connection,
      Mailer $mailer,
      UserManager $userManager
  ) {
      $this->db = $connection;
      $this->mailer = $mailer;
      $this->userManager = $userManager;
  }
}
```

Unclear execution paths

```
if (
  $station->getStatus() === Status::ACTIVE
  || (
    $station->getStatus() === Status::PENDING
    && $station->createdAt() < $yesterday
  )
  || $user->isOwner($station)
  || $user->isSuperAdmin()
) { /* .. */ }
```



Unclear execution paths - solution

```
if (
  $station->isEditable()
  ||
  $user->canEditStation($station)
) { /* .. */ }
```

Long distance calls

```
public function notifyOwner(int $stationId) {
  $email = $this
     ->stationManager
     ->getStation($stationId)
     ->getOwner()
     ->getEmail();
  /* .. */
}
```

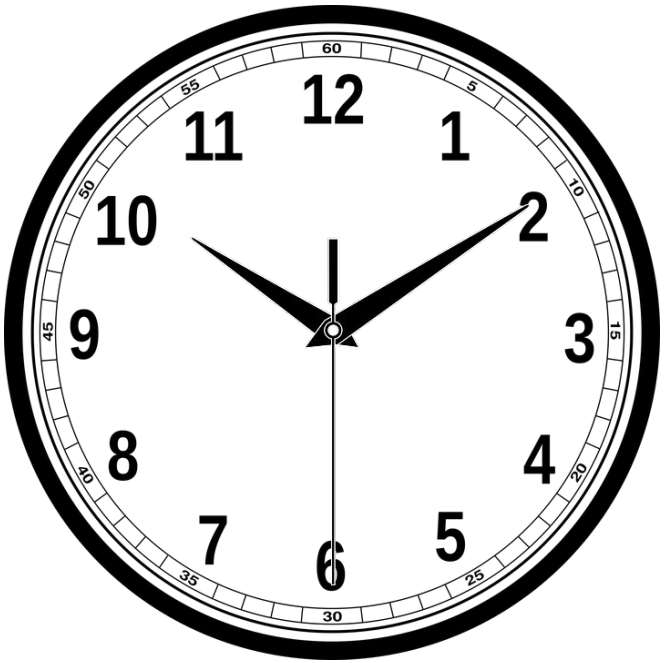
Long distance calls - solution

```
public function notifyOwner(Station $station) {
  $email = $station->getOwner()->getEmail();
  /* .. */
}
```

Hidden inputs

```php
public function todayIsWeekend() {
  $date = date();
  /* .. */
}

public function getUserDaytime() {
  $time = time();
  $userIp = $_SERVER['REMOTE_ADDR'];
  $location =
    unserialize(file_get_contents(
      'http://www.geolocate.org/?ip=' . $userIp
    ));
  /* .. */
}
```



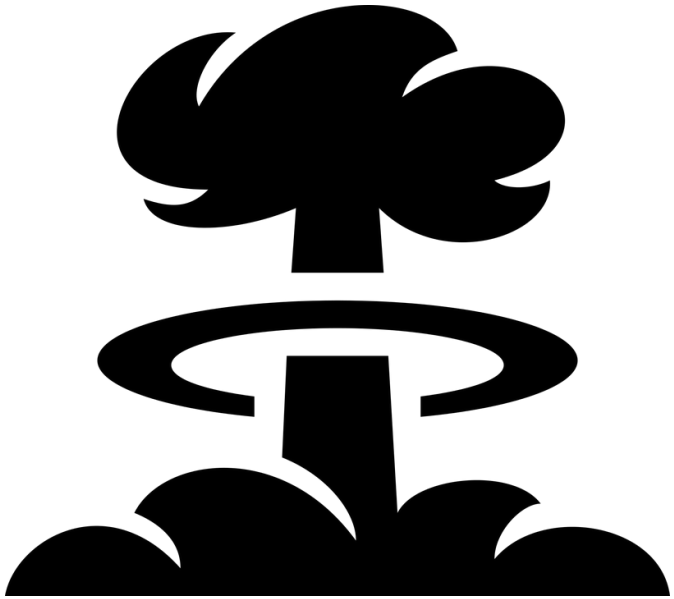
Hidden inputs - solution

```
public function dateIsWeekend(DateTime $date) {
  /* .. */
}

public function getLocationDaytime(
    Location $location,
    DateTime $time
) {
  /* .. */
}
```

Side effects

```
public function addImage(Image $image) {
  /* .. */
  Gallery::getInstance()->images++;
}
```



Side effects - solution

```
public function addImage(Image $image) {
  /* .. */
  $this->gallery->images++;
}
```

Perfect test

```
public function getItem(int $id) : Item
{
  return $this->repository->get($id);
}
```

Expectations

- Are not unit tests
- Do not replace unit tests

```
public function getItem(int $id) : Item {
  assert($this->repository->isOpen === true);

  return $this->repository->get($id);
}
```

Conclusions

- **SOLID, DRY, KISS**
- **Antipatterns**
- **Architecture**

Questions?

Thank you!