

EXERCISE 0

Checkout and do composer install

<https://github.com/lmuzinic/ipc-pragmatic-tdd>

```
git clone git@github.com:lmuzinic/ipc-pragmatic-tdd.git
cd ipc-pragmatic-tdd
composer install
vendor/bin/phpunit
```

OK (1 test, 1 assertion)

There is <10 MB of packages to download, so you can use your mobile data if WIFI does not work.



PRAGMATIC TDD

HELLO



Luka Muzinic
@lmuzinic

WORKSHOP RULES

ASK QUESTIONS

IF YOU STILL DO NOT UNDERSTAND, ASK QUESTIONS AGAIN

DISCUSS RIGHT NOW, DO NOT WAIT FOR THE “RIGHT MOMENT”

WHY WE NEED TESTING?

EXERCISE 0

Checkout and do composer install

<https://github.com/lmurzin/ipc-pragmatic-tdd>

```
git clone git@github.com:lmurzin/ipc-pragmatic-tdd.git
cd ipc-pragmatic-tdd
composer install
vendor/bin/phpunit
```

OK (1 test, 1 assertion)

There is <10 MB of packages to download, so you can use your mobile data if WIFI does not work.

**WHY AM I
HERE?**

**ARE WE
SOFTWARE TESTERS?**

**AND YET
WE KEEP ON SAYING
WE WRITE TESTS...**

**WHERE CAN I GET
MORE OF THOSE
TESTS?**



EXCUSES, EXCUSES

TESTS SLOW US DOWN

WE WILL NEVER GET TO 100% CODE COVERAGE

WE DO NOT HAVE TIME TO LEARN TESTING, WE'RE TOO BUSY SHIPPING CODE

EXCUSES, EXCUSES

TESTS SLOW US DOWN?

STOP TESTFILEING

DO YOU OFTEN DO THIS?

~ `php test.php`

OR THIS

`http://localhost/test_problem.php`

EXCUSES, EXCUSES

**WE WILL NEVER GET TO 100%
CODE COVERAGE**

CODE COVERAGE

210470	2406 / 4014	<div><div></div></div>	67.28%
0.00%	0 / 473	<div><div></div></div>	0.00%
64.78%	241 / 372	<div><div></div></div>	74.42%
100.00%	142 / 142	<div><div></div></div>	100.00%
100.00%	72 / 72	<div><div></div></div>	100.00%
0.00%	0 / 11	<div><div></div></div>	0.00%
77.36%	82 / 106	<div><div></div></div>	77.36%
0.00%	0 / 8	<div><div></div></div>	0.00%
62.59%	39 / 62	<div><div></div></div>	62.59%




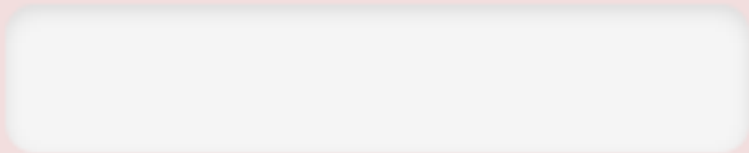
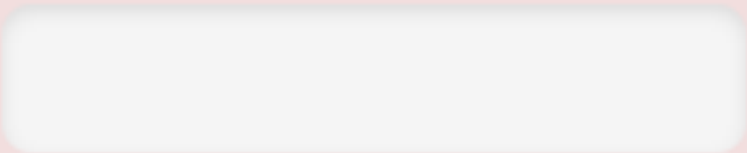
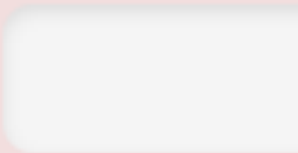















CODE COVERAGE

Classes and Traits		
<div><div></div></div>	56.88%	62 / 109
<div><div></div></div>	0.00%	0 / 8
<div><div></div></div>	64.29%	9 / 14
<div><div></div></div>	100.00%	9 / 9
<div><div></div></div>	100.00%	5 / 5
<div><div></div></div>	76.47%	13 / 17
<div><div></div></div>	48.15%	26 / 54
	n/a	0 / 0








CODE COVERAGE

Code Coverage					
Functions and Methods			Classes and Traits		
<div><div></div></div>	67.28%	368 / 547	<div><div></div></div>	56.88%	62 / 109
<div><div></div></div>	0.00%	0 / 28	<div><div></div></div>	0.00%	0 / 8
<div><div></div></div>	74.42%	32 / 43	<div><div></div></div>	64.29%	9 / 14
<div><div></div></div>	100.00%	52 / 52	<div><div></div></div>	100.00%	9 / 9
<div><div></div></div>	100.00%	23 / 23	<div><div></div></div>	100.00%	5 / 5
<div><div></div></div>	77.36%	82 / 106	<div><div></div></div>	76.47%	13 / 17
<div><div></div></div>	62.59%	179 / 286	<div><div></div></div>	48.15%	26 / 54
	n/a	0 / 0		n/a	0 / 0

CODE COVERAGE

Code Coverage						
Lines			Functions and Methods			
	59.94%	2406 / 4014		67.28%	368 / 547	
	0.00%	0 / 473		0.00%	0 / 28	
	64.78%	241 / 372		74.42%	32 / 43	
	100.00%	142 / 142		100.00%	52 / 52	
	100.00%	72 / 72		100.00%	23 / 23	
	67.04%	299 / 446		77.36%	82 / 106	
	67.54%	1652 / 2446		62.59%	179 / 286	
	n/a	0 / 0		n/a	0 / 0	

20 > 80

	Code Coverage				Cod
		Lines		Function	
Total	<div><div></div></div>	59.94%	2406 / 4014	<div><div></div></div>	
 Command	<div><div></div></div>	0.00%	0 / 473	<div><div></div></div>	
 Controller	<div><div></div></div>	64.78%	241 / 372	<div><div></div></div>	
 Entity	<div><div></div></div>	100.00%	142 / 142	<div><div></div></div>	
 Model	<div><div></div></div>	100.00%	72 / 72	<div><div></div></div>	
 Repository	<div><div></div></div>	67.04%	299 / 446	<div><div></div></div>	
 Service	<div><div></div></div>	67.54%	1652 / 2446	<div><div></div></div>	
 AppBundle.php		n/a	0 / 0		

EXCUSES, EXCUSES

**WE DO NOT HAVE TIME TO LEARN TESTING,
WE'RE TOO BUSY SHIPPING CODE**

//@TODO: STANDSTILL

DON'T BE SCARED OF PHPUNIT*

IT IS JUST A CODE RUNNER

UNIT, INTEGRATION OR ACCEPTANCE TESTS

SMOKE TESTS

WEBSITE SCRAPER

...

TESTS ANSWER QUESTIONS

IS MY CODE WORKING CORRECTLY?

WILL MY CODE WORK CORRECTLY?

HOW IS MY CODE SUPPOSE TO WORK?

IS MY CODE DESIGNED WELL?

CAN I REFACTOR MY CODE?

TESTS ANSWER QUESTIONS

IS MY CODE WORKING CORRECTLY?

WILL MY CODE WORK CORRECTLY?

HOW IS MY CODE SUPPOSE TO WORK?

IS MY CODE DESIGNED WELL?

CAN I REFACTOR MY CODE?

TESTS ANSWER QUESTIONS

IS MY CODE WORKING CORRECTLY?

WILL MY CODE WORK CORRECTLY?

HOW IS MY CODE SUPPOSE TO WORK?

IS MY CODE DESIGNED WELL?

CAN I REFACTOR MY CODE?

TESTS ANSWER QUESTIONS

IS MY CODE WORKING CORRECTLY?

WILL MY CODE WORK CORRECTLY?

HOW IS MY CODE SUPPOSE TO WORK?

IS MY CODE DESIGNED WELL?

CAN I REFACTOR MY CODE?

TESTS ANSWER QUESTIONS

IS MY CODE WORKING CORRECTLY?

WILL MY CODE WORK CORRECTLY?

HOW IS MY CODE SUPPOSE TO WORK?

IS MY CODE DESIGNED WELL?

CAN I REFACTOR MY CODE?

TESTS ANSWER QUESTIONS

IS MY CODE WORKING CORRECTLY?

WILL MY CODE WORK CORRECTLY?

HOW IS MY CODE SUPPOSE TO WORK?

IS MY CODE DESIGNED WELL?

CAN I REFACTOR MY CODE?

HOW TO WRITE TESTS?

GIVEN WHEN THEN

SETUP EXERCISE VERIFY TEARDOWN

ANATOMY OF PHPUNIT TEST CASE

```
class TeamTest extends PHPUnit\Framework\TestCase;
{
    private $team;

    public function setUp()
    {
        $this->team = Team::create('Hellas Verona');
    }

    public function testGetName()
    {
        $name = $this->team->getName();

        $this->assertEquals('Hellas Verona', $name);
    }
}
```

WORKSHOP RULES

ASK QUESTIONS

IF YOU STILL DO NOT UNDERSTAND, ASK QUESTIONS AGAIN

DISCUSS RIGHT NOW, DO NOT WAIT FOR THE “RIGHT MOMENT”



EXERCISE 0

Checkout and do composer install

<https://github.com/lmuzinic/ipc-pragmatic-tdd>

```
git clone git@github.com:lmuzinic/ipc-pragmatic-tdd.git
cd ipc-pragmatic-tdd
composer install
vendor/bin/phpunit
```

OK (1 test, 1 assertion)

There is <10 MB of packages to download, so you can use your mobile data if WIFI does not work.

EXERCISE 1

BINARY GAP

https://app.codility.com/programmers/lessons/1-iterations/binary_gap/

Find longest sequence of zeros in binary representation of an integer.

A binary gap within a positive integer N is any maximal sequence of consecutive zeros that is surrounded by ones at both ends in the binary representation of N .

How many tests we need to write?

EXERCISE 1 cont.

BINARY GAP

For example, number 9 has binary representation 1001 and contains a binary gap of length 2. The number 529 has binary representation 1000010001 and contains two binary gaps: one of length 4 and one of length 3. The number 20 has binary representation 10100 and contains one binary gap of length 1. The number 15 has binary representation 1111 and has no binary gaps. The number 32 has binary representation 100000 and has no binary gaps.

Reuse existing `GetTheBallRollingTest.php` to add more test cases.

EXERCISE 2

ODD OCCURRENCES IN ARRAY

https://app.codility.com/programmers/lessons/2-arrays/odd_occurrences_in_array/

A non-empty array A consisting of N integers is given. The array contains an odd number of elements, and each element of the array can be paired with another element that has the same value, except for one element that is left unpaired.

How many tests we need to write?

EXERCISE 2 cont.

ODD OCCURRENCES IN ARRAY

https://app.codility.com/programmers/lessons/2-arrays/odd_occurrences_in_array/

A[0] = 9 A[1] = 3 A[2] = 9 A[3] = 3 A[4] = 9 A[5] = 7 A[6] = 9
A[0] = 7 A[1] = 3 A[2] = 3
A[0] = 3 A[1] = 7 A[2] = 3
A[0] = 3 A[1] = 3 A[2] = 7

Again, reuse existing `GetTheBallRollingTest.php` to add more test cases.

SO FAR SO GOOD?

THIS IS EXTREMELY SIMPLIFIED EXAMPLE

- our applications are a bit bigger and a bit more complicated

FOOTBALL

SOMEWHAT SIMPLIFIED

- two teams, Home team and Away team play a Match
- winner of the match is a team that scored more goals *
 - winner gets three (3) points for winning
 - loser gets nothing
- goals that Home team scored are also goals that Away team received and vice versa
- repeat for desired number of times
- overall winner is a team with most points

- in case of both teams with equal goals
 - there is no winner/loser, draw is recorded
 - both get one (1) point

Matchday 7 of 34



Werder

2



FT

Fri, 05/10



Wolfsburg

0



Hannover 96

3



FT

Sat, 06/10



VfB Stuttgart

1



Fortuna

0

FT

Sat, 06/10



Schalke 04

2



Mainz

0



Hertha

0



Dortmund

4



FT

Sat, 06/10



FC Augsburg

3



Bayern

0



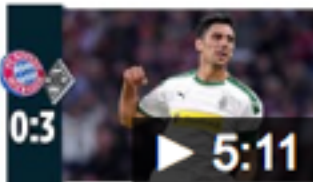
Mönchengladbach

3



FT

Sat, 06/10



SC Freiburg

0

FT

Sun, 07/10



Bayer

0



Hoffenheim

1



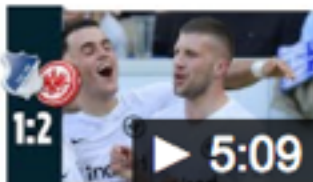
Eintracht














2



FT

Sun, 07/10



Club	MP	W	D	L	GF	GA	GD	Pts
1  Dortmund	7	5	2	0	23	8	15	17
2  RB Leipzig	7	4	2	1	16	9	7	14
3  Mönchengladbach	7	4	2	1	15	9	6	14
4  Werder	7	4	2	1	13	8	5	14
5  Hertha	7	4	2	1	12	7	5	14
6  Bayern	7	4	1	2	12	8	4	13
7  Eintracht	7	3	1	3	12	11	1	10
8  Mainz	7	2	3	2	4	4	0	9
9  Wolfsburg	7	2	3	2	10	11	-1	9
10  FC Augsburg	7	2	2	3	14	13	1	8
11  SC Freiburg	7	2	2	3	9	13	-4	8
12  Nürnberg	7	2	2	3	7	16	-9	8
13  Hoffenheim	7	2	1	4	11	12	-1	7

DISCUSSION

IMPLEMENT A FOOTBALL LEAGUE MANAGEMENT APP

– We want to display standings table on our website

WHAT IS YOUR BIGGEST CONCERN?

EXERCISE 3

START IMPLEMENTING STANDINGS

Write a test for getting sorted standings

Talk about domain

Implementation

EXERCISE 3 cont.

START IMPLEMENTING STANDINGS

Create `tests/Domain/Standings/StandingsTest.php`

EXERCISE 4

IMPLEMENT TEAM POSITION

An object that will hold position inside the league table

Focus just on this class, use `--filter`

EXERCISE 4 cont.

IMPLEMENT TEAM POSITION

Create

`tests/Domain/Standings/TeamPositionTest.php`

Possible test cases

`testGetPointsWithNoGames`

`testGetPointsWithThreeGamesPlayed`

`testGetGoalsScoredWithNoGames`

`testGetGoalsScoredWithThreeGamesPlayed`

`testGetGoalsAgainstWithNoGames`

`testGetGoalsAgainstWithThreeGamesPlayed`

GO TO VACATION

WHO WILL IMPLEMENT STANDINGS?

Your colleagues that stayed in the office.

Checkout branch [step-3](#)

EXERCISE 5

REFACTOR

Replace `spl_object_hash` with `sha1` from team name

Move checking, creating and returning a `TeamPosition` into separate method

SO FAR SO GOOD?

THERE WAS THIS ONE EDGE CASE

- two teams were tied at the end of the season, they had an equal amount of points and our system could not decide who should be first
- we should do something about it for next season

EXERCISE 6

NEXT YEAR FEATURES

League manager wants to change the rules for scoring – teams that have equal number of points, sort by goals scored (higher goals scored moves up). Keep the possibility to show scoring for last year as well.

EXERCISE 6 cont.

NEXT YEAR FEATURES

Introduce

`src/Domain/RuleBook/RuleBookInterface.php`

Introduce method

`decide(TeamPosition $teamA, TeamPosition $teamB)`

Extract sorting logic into separate class, cover all cases with tests.

`tests/Domain/RuleBook/SimpleRuleBookTest.php`

Test cases

`testDecideReturnsLessThanZeroWhenFirstTeamHasMorePointsThanSecond`

`testDecideReturnsGreaterThanZeroWhenSecondTeamHasMorePointsThanFirst`

`testDecideReturnsZeroWhenTeamsHaveEqualPoints`

EXERCISE 7

CAN TESTING OF RULEBOOKS BE EASIER?

Can we do something with less repeating?

EXERCISE 7 cont.

TEST DUMMIES/MOCK OBJECTS

Refactor the test using stubs

```
/** @var \PHPUnit_Framework_MockObject_MockObject $teamA */  
$teamA = $this->getMockBuilder(TeamPosition::class)->disableOriginalConstructor()->getMock();  
$teamA->method('getPoints')->willReturn(42);
```

EXERCISE 8

ADVANCED RULEBOOK

Write the test.

Write the implementation.

Push the both rulebooks into standings tests.

EXERCISE 8 cont.

ADVANCED RULEBOOK

Write the test.

Write the implementation.

Push the both rulebooks into standings tests.

Tests

tests/Domain/Standings/StandingsWithSimpleRuleBook.php

tests/Domain/Standings/StandingsWithAdvancedRuleBook.php

Implement scenarios

EXERCISE 9

CODECOVERAGE & CRAP

Run codecoverage

```
~ vendor/bin/phpunit --coverage-html var/coverage
```

Explain CRAP

$CRAP = CC^2 \times U^3 + CC$

EXERCISE 10

EXCEPTIONS

Last year I have entered a match with same teams

Last year I have created a team with no name

EXERCISE 10 cont.

EXCEPTIONS

Last year I have entered a match with same teams

Last year I have created a team with no name

Introduce

src/Domain/Exception/BallGameException.php

src/Domain/Exception/MatchBetweenSameTeamException.php

src/Domain/Exception/TeamWithEmptyNameException.php

Test with

@expectedException \BadMatchException

or

`$this->expectException(BadMatchException::class);`

EXERCISE 11

REPOSITORIES

It makes sense that Standings should use a Repository

Implement such repository, add `sleep(1)` to each method..

EXERCISE 11 cont.

REPOSITORIES

Refactor `Standings::record` **and** `Standings::getSortedStandings` **to use Repository.**

Introduce

`tests/Domain/Standings/StandingsWithRepositoryTest.php`

Use stubs for findAll method

Use spies for save method

```
->expects($this->once())->method('save');
```


WORKSHOP RECAP



QUESTIONS? /r/AMA?



Luka Muzinic

@lmuzinic

luka.muzinic.net

luka.muzinic.net/pdf/ipc-pragmatic-tdd.pdf

READING & LIBRARIES

Reading list

<https://www.devmynd.com/blog/five-factor-testing/>

<https://martinfowler.com/articles/practical-test-pyramid.html>

<https://dev.to/theobendixson/the-problem-that-unit-tests-solve-b2l>

<https://blog.liplex.de/testing-private-and-protected-methods-with-phpunit/>

Libraries

<https://github.com/sebastianbergmann/phpunit>

<https://github.com/phpspec/phpspec>

<https://github.com/Codeception/Codeception>

<https://github.com/phpstan/phpstan>

<https://github.com/infection/infection>

HOMEWORK

COMPLETE EXERCISES

- continue with BallGame example**
- follow along/[google/ask](#)**

HOMEWORK

GAMES WON

- implement the feature, given a scenario, where standings table displays game won

TIES

- implement the feature, given a scenario, where two teams have played a tie

TWO POINTS

- given a scenario where RuleBook also defines how much points should each team get after match win, write additional test case where each win gets you 2 points

ANYTHING GOES

- have an idea how to make this whole thing better?
- write tests and demonstrate :)

THANK YOU



Luka Muzinic

@lmuzinic

luka.muzinic.net

luka.muzinic.net/pdf/ipc-pragmatic-tdd.pdf

Photos by Les Anderson, Joshua Earle, Ian Espinosa and Tom Roberts on Unsplash