

Sonova Warsaw Service Center, 2018.12.15

Test-driven development dla początkujących

dr inż. Bartosz Papis

```
import unittest
from flask import Flask
from flask import render_template, request, redirect
from flask import url_for
from flask_testing import TestCase
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField

print(„All OK!")
```

https://github.com/bartoszkp/TDDWorkshop/blob/master/check_environment.py

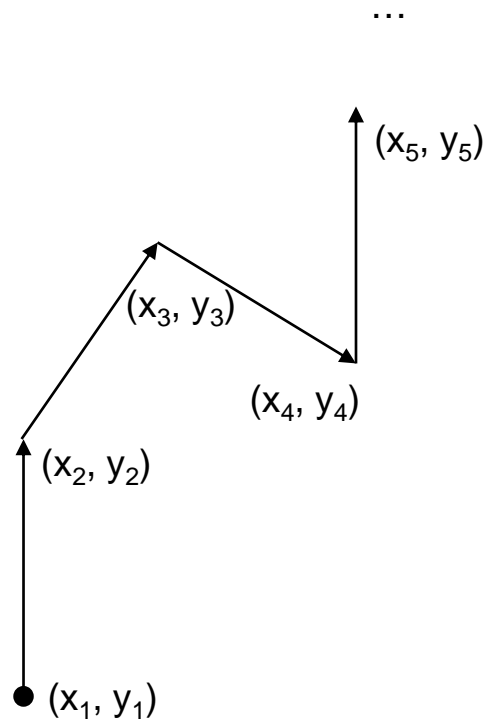
1. Przykład – Boat Kata
2. Po co pisze się testy?
3. Czemu nie pisze się testów?
4. Czym jest Test-Driven Development?
5. Przykład – Konwerter temperatur
6. Podsumowanie
7. Ćwiczenie

Kata created by Jarosław Karczmarczyk: <https://github.com/karczmarczyk/Boat-Kata>

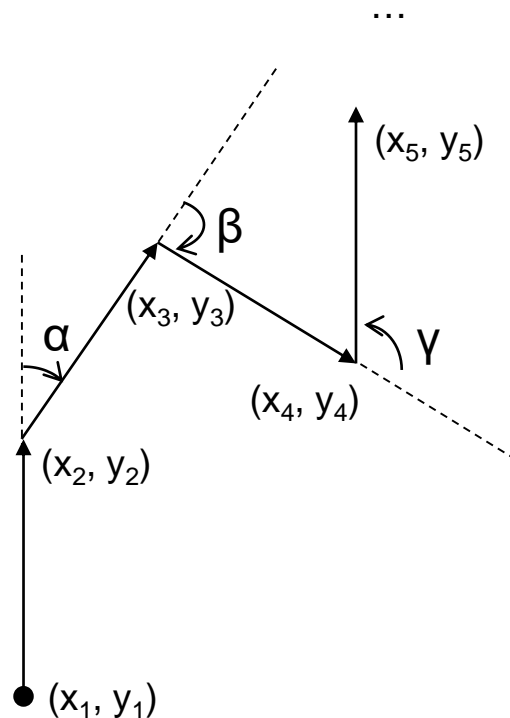


https://en.wikipedia.org/wiki/File:Boat_-_Little_Belt_-_02.jpeg (CC 3.0)

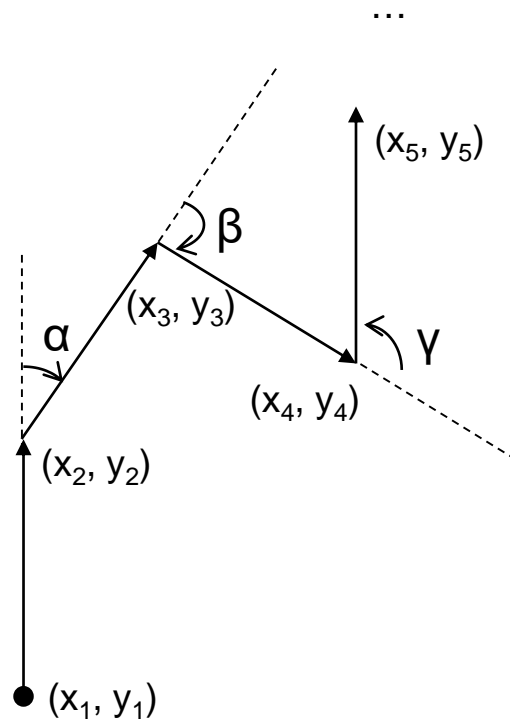
Mając daną listę punktów, obliczyć kierunek, w którym płynie łódka względem pozycji początkowej



Kierunek to Prawo wtedy i tylko wtedy gdy $\alpha + \beta - \gamma + \dots > 0$.
W przeciwnym przypadku Lewo.



Uproszczenie: założmy, że mamy już daną listę $[\alpha, \beta, \gamma, \dots]$



Prezentacja przykładowego rozwiązania

By się upewnić, że kod robi to, co się wydaje, że robi.

By się upewnić, że kod robi to, co się wydaje, że robi.

- Najłatwiejszy sposób na uruchomienie kodu

By się upewnić, że kod robi to, co się wydaje, że robi.

- Najłatwiejszy sposób na uruchomienie kodu
- Najłatwiejszy sposób na weryfikację przypadków brzegowych

By się upewnić, że kod robi to, co się wydaje, że robi.

- Najłatwiejszy sposób na uruchomienie kodu
- Najłatwiejszy sposób na weryfikację przypadków brzegowych
- Najłatwiejszy sposób na odtworzenie błędu

By się upewnić, że kod robi to, co się wydaje, że robi.

- Najłatwiejszy* sposób na uruchomienie kodu
- Najłatwiejszy* sposób na weryfikację przypadków brzegowych
- Najłatwiejszy* sposób na odtworzenie błędu

(*) zakładając, że kod jest testowalny

Po co pisze się testy?

By minimalizować liczbę regresji.

By minimalizować regresje.

- Nie da się wprowadzić drugi raz tego samego błędu

By minimalizować regresje.

- Nie da się* wprowadzić drugi raz tego samego błędu

(*) zakładając, że testy są uruchamiane i traktowane poważnie

Po co pisze się testy?

By zagwarantować dobrą strukturę kodu.

By zagwarantować dobrą strukturę kodu.

- Brak ukrytych zależności

By zagwarantować dobrą strukturę kodu.

- Brak ukrytych zależności
- Spełnienie Single Responsibility Principle

By zagwarantować dobrą strukturę kodu.

- Brak ukrytych zależności
- Spełnienie Single Responsibility Principle
- (...)

By zagwarantować dobrą strukturę kodu.

- Brak* ukrytych zależności
- Spełnienie* Single Responsibility Principle
- (...)*

(*) chyba, że ktoś się bardzo napracuje by stworzyć testy do kodu bez tych cech

Po co pisze się testy?

By stworzyć zawsze aktualną dokumentację.

By stworzyć zawsze aktualną dokumentację.

- Testy się nie przedawniają

By stworzyć zawsze aktualną dokumentację.

- Testy się nie* przedawniają

(*) chyba że nie są uruchamiane albo są ignorowane

- By się upewnić, że kod robi to, co się wydaje, że robi.
- By minimalizować regresje.
- By zagwarantować dobrą strukturę kodu.
- By stworzyć zawsze aktualną dokumentację.

By się upewnić, że kod robi to, co się wydaje, że robi.*

By minimalizować regresje.

By zagwarantować dobrą strukturę kodu.**

By stworzyć zawsze aktualną dokumentację.

(*) Bissi et al., The effects of test driven development on internal quality, external quality and productivity: A systematic review, 2016 Elsevier

(**) Madeyski, Test-Driven Development: An Empirical Evaluation of Agile Practice, 2010 Springer-Verlag

By sprzyjać odtwarzalności wyników.
Z poczucia odpowiedzialności.*

(*) "Uncle" Bob Martin - "The Future of Programming"

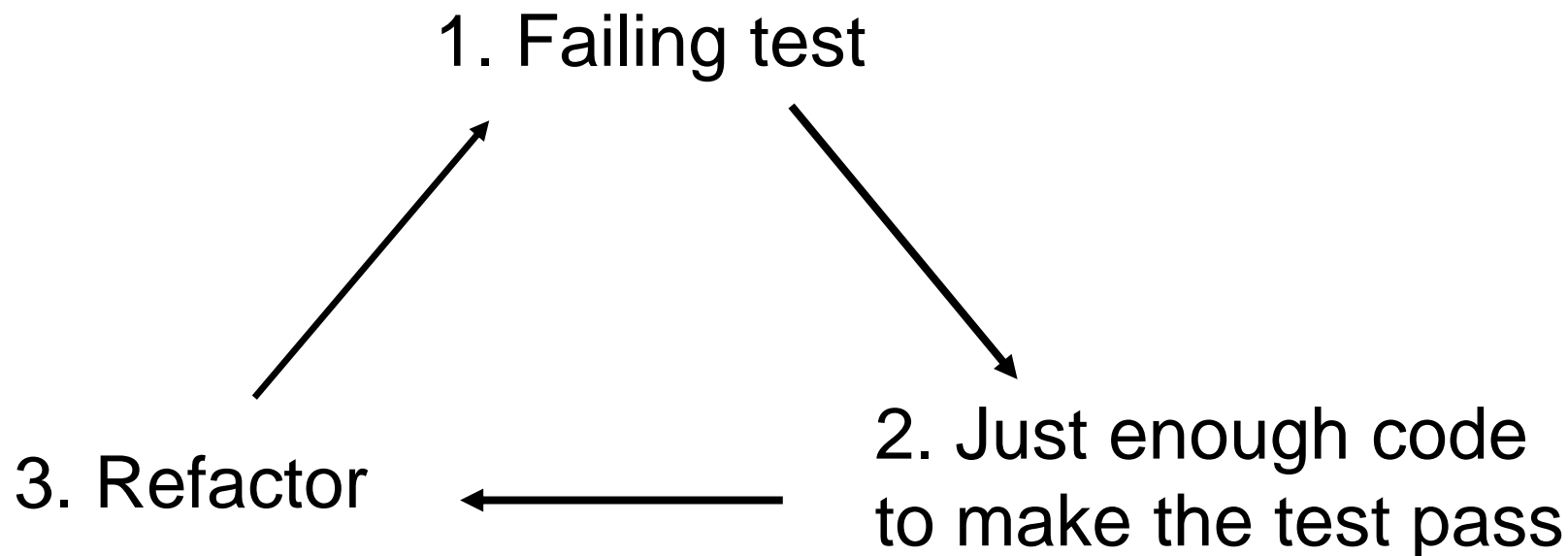
- Postrzegane jako strata czasu.

Czemu nie pisze się testów?

- Postrzegane jako strata czasu.
- Trudne.

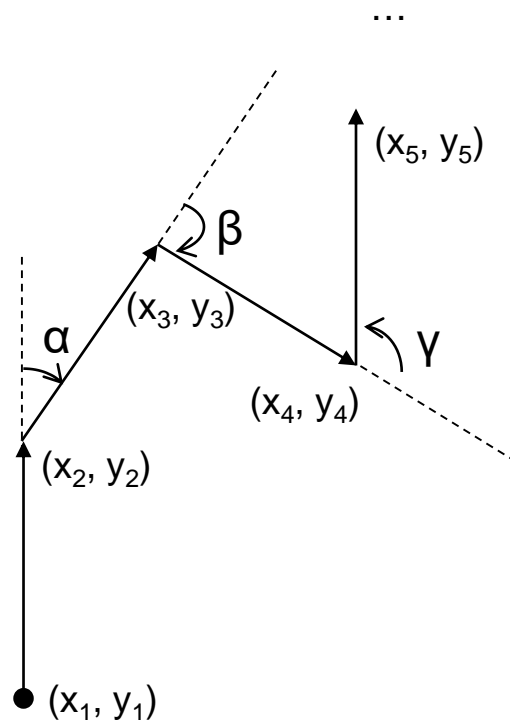
Czym jest Test-driven development?

Kent Beck, 2003 r.



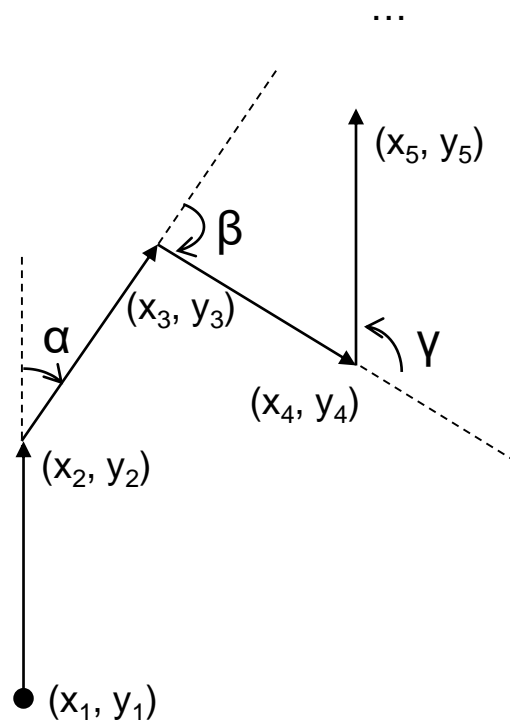
Zły sposób TM

- Kilka godzin na odświeżenie trygonometrii
- Kilka godzin obliczeń na kartce
- Kilka godzin implementacji



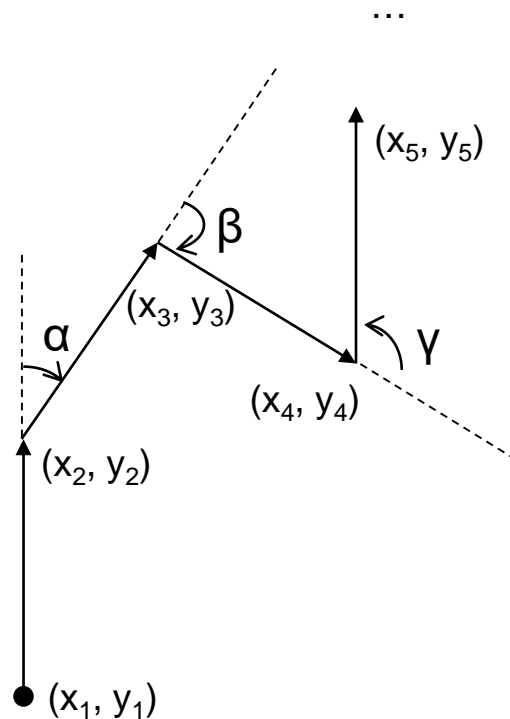
Niezły sposób TM

- Kilka godzin na odświeżenie trygonometrii
- Kilka godzin obliczeń na kartce
- Kilka godzin implementacji
- Kilka godzin pisania testów



Dobry sposób™

- Rozwiązywanie problemu małymi, weryfikowalnymi krokami



- Postrzegane jako strata czasu.
- Jeszcze trudniejsze niż pisanie testów.
- Źle rozumiane.
- Prezentowane tylko na „zabawkowych przykładach”.

Stworzyć aplikację webową umożliwiającą użytkownikowi konwersję podanej temperatury ze °C na °F i ze °F na °C.

Prezentacja przykładowego rozwiązania

- Metoda naukowa w programowaniu*

(*) Mugridge, Test Driven Development and the Scientific Method, 2003 IEEE (Agile Development Conference)

- Metoda naukowa w programowaniu*
- Narzędzie pomocne przy trudnych problemach

(*) Mugridge, Test Driven Development and the Scientific Method, 2003 IEEE (Agile Development Conference)

- Metoda naukowa w programowaniu*
- Narzędzie pomocne przy trudnych problemach
- ... którego można używać efektywnie**

(*) Mugridge, Test Driven Development and the Scientific Method, 2003 IEEE (Agile Development Conference)

(**) Beck, Test Driven Development: By Example, 2000 Addison-Wesley Professional

- Metoda naukowa w programowaniu*
- Narzędzie pomocne przy trudnych problemach
- ... którego można używać efektywnie**
- ... gwarantujące wysokie pokrycie kodu testami

(*) Mugridge, Test Driven Development and the Scientific Method, 2003 IEEE (Agile Development Conference)

(**) Beck, Test Driven Development: By Example, 2000 Addison-Wesley Professional

Dziękuję za uwagę 😊

Stworzyć funkcję, która przyjmuje dodatnią liczbę całkowitą, a zwraca jej zapis w notacji rzymskiej, tj.:

- 1 → "I"
- 2 → "II"
- ...
- 4 → "IV"
- 5 → "V"
- 6 → "VI"
- ...
- 9 → "IX"
- 10 → "X"
- ...
- 14 → "XIV"
- ...
- 40 → "XL"
- ...