

PW, WEiTI, ZBiUM, 2018.10.29

Test-driven development na przykładzie funkcji matematycznej

dr inż. Bartosz Papis

1. Po pisze się testy?
 2. Czemu nie pisze się testów?
 3. Czym jest Test-Driven Development?
 4. Boat kata
 5. Podsumowanie
-

Po co pisze się testy?

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 regresje.

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"

Czemu nie pisze się testów?

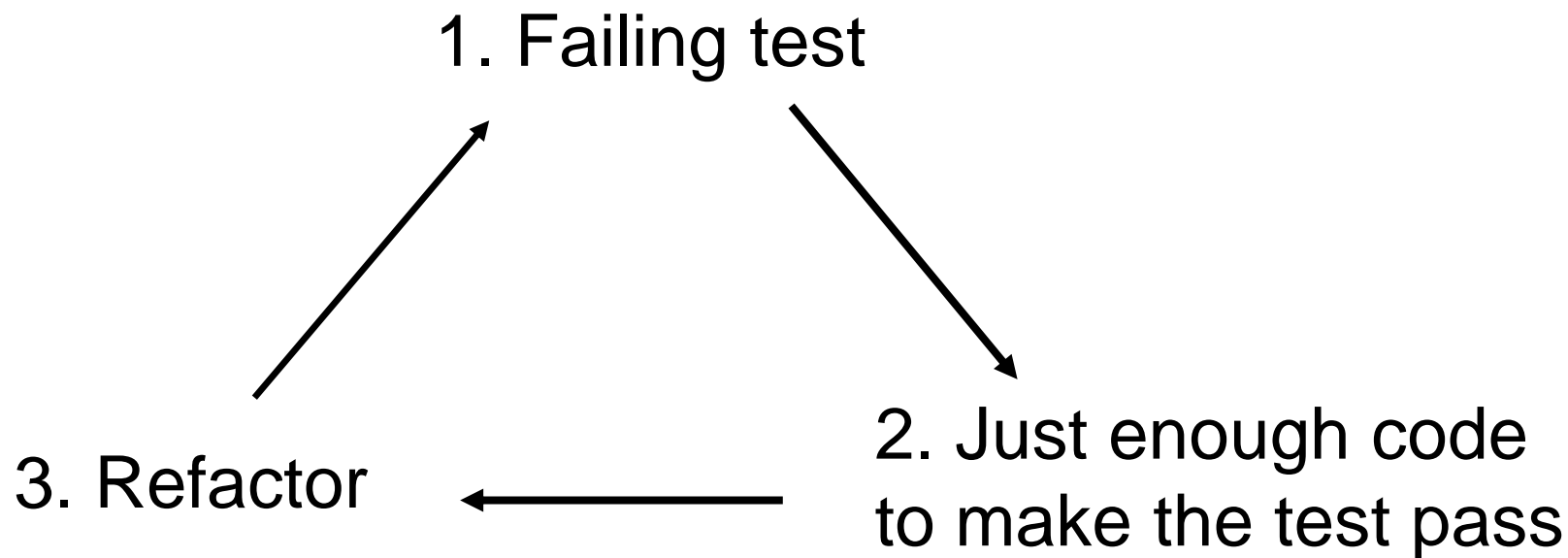
- Postrzegane jako strata czasu.

Czemu nie pisze się testów?

- Postrzegane jako strata czasu.
- Trudne.

Czym jest Test-driven development?

Kent Beck, 2003 r.



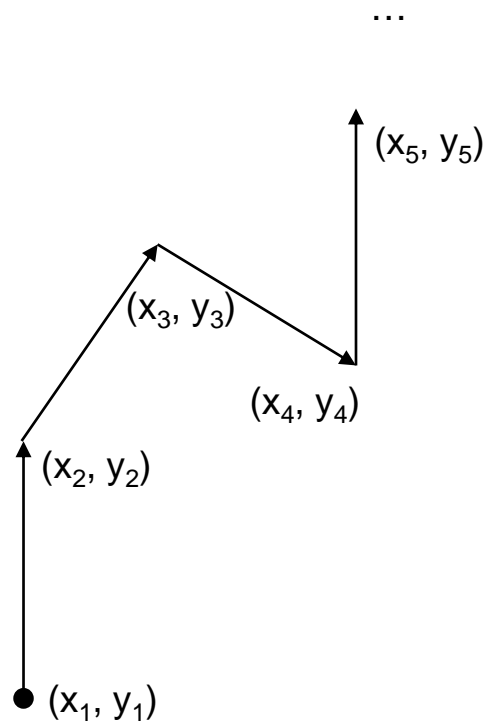
- Postrzegane jako strata czasu.
- Jeszcze trudniejsze niż pisanie testów.
- Źle rozumiane.

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

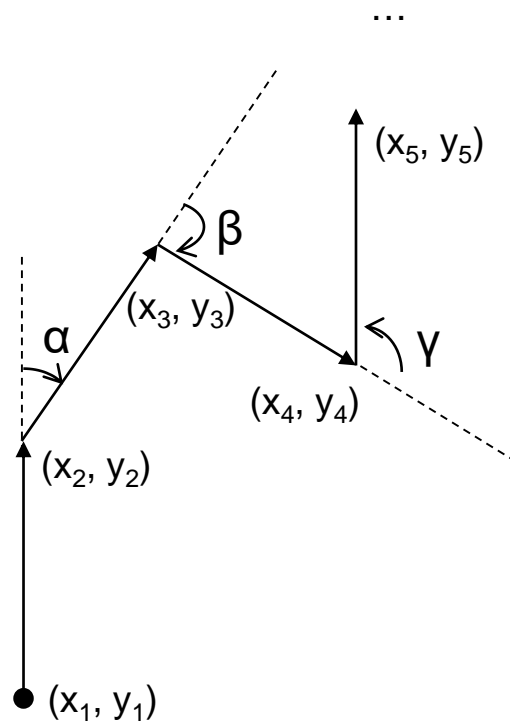


https://en.wikipedia.org/wiki/File:Boat_-_Little_Belt_-_02.jpeg (CC 3.0)

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

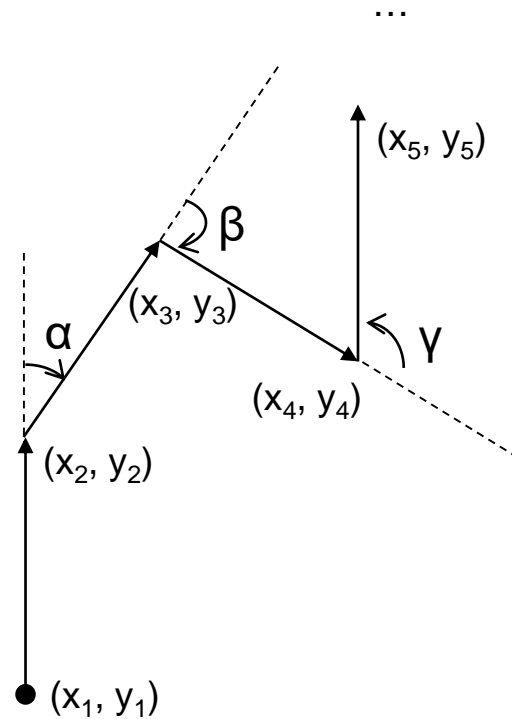


Kierunek to *Prawo* wtedy i tylko wtedy gdy $\alpha + \beta - \gamma + \dots > 0$ – wpp. *Lewo*



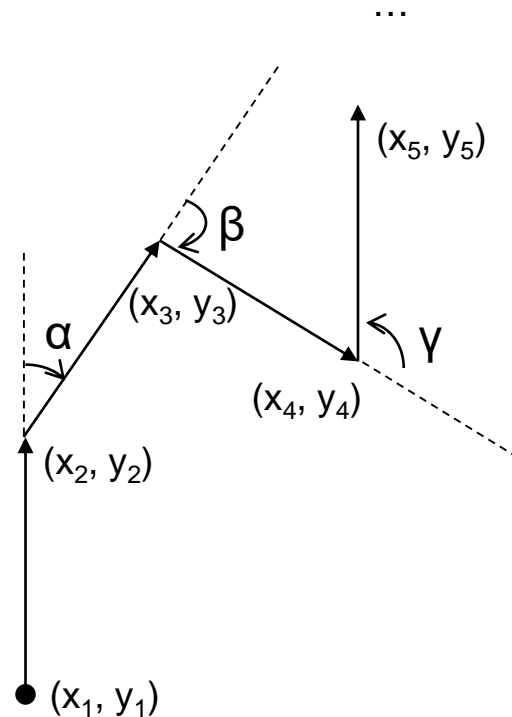
Zły sposób™

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



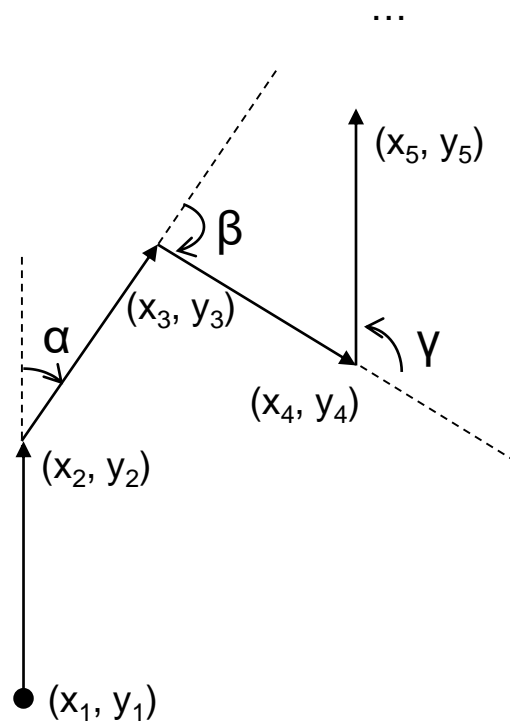
Niezły sposób™

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



Dobry sposób™

- Rozwiązywanie problemu małymi, weryfikowalnymi krokami



Dobry sposób™

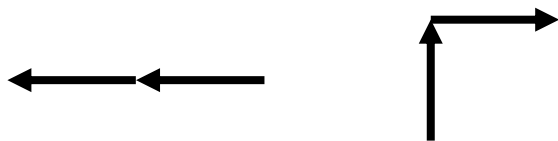
- Rozwiązywanie problemu małymi, weryfikowalnymi krokami

Prezentacja w Visual Studio

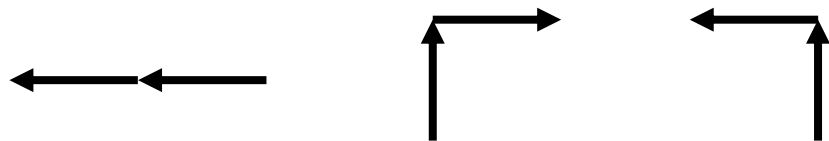
Dobry sposób™



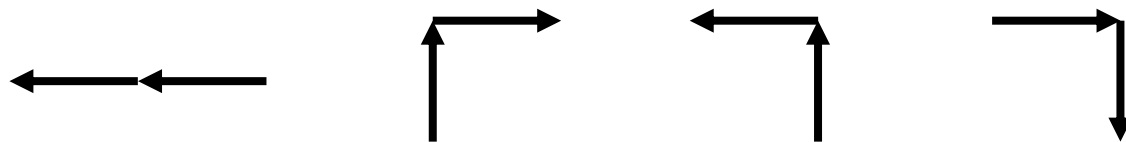
Dobry sposób™



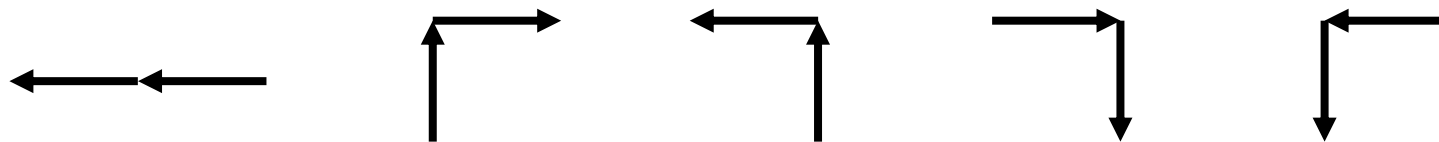
Dobry sposób™



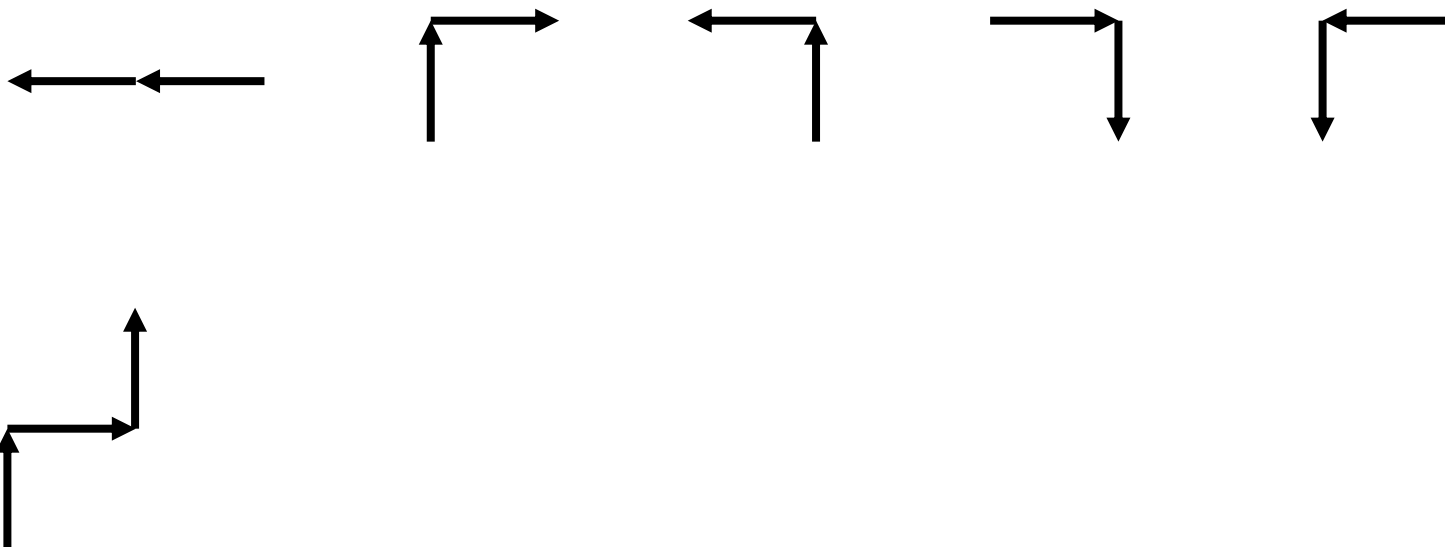
Dobry sposób™



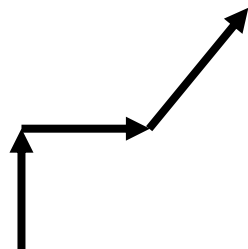
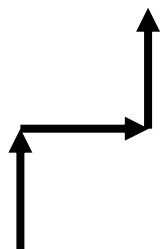
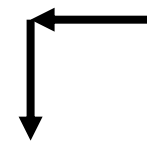
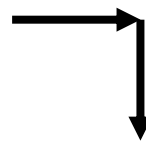
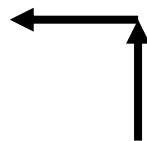
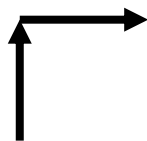
Dobry sposób™



Dobry sposób™



Dobry sposób™



- 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 😊