



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



**Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego**

Testowanie aplikacji w Ruby

7 wykładów po 2 godz. + 1 wykład 1godz.



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



**Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego**

Wykład 1

Wprowadzenie do języka Ruby

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Skrypt *hello_world.rb*

```
#!/bin/env ruby  
puts "hello world"
```

Bash:

```
export RUBYOPT=-w
```

Gemy – co to jest?

Więcej informacji:

M. Marohnić. Ruby verbose mode and how it's broken.

<http://mislav.uniqpath.com/2011/06/ruby-verbose-mode/>

Bozhidar Batsov. Ruby Style Guide.

<https://github.com/bbatsov/ruby-style-guide>

Integracja statycznego analizatora kodu rubocop.

<https://github.com/bbatsov/rubocop> z edytorami (Atom, Emacs).



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europejskich 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Podstawowe elementy języka: **instrukcje warunkowe**

- `if / elsif / else / end`, modyfikator `if`
- `unless / else / end`, modyfikator `unless`
- `case / when / else / end`



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Podstawowe elementy języka: **pętle**

- `while / end`, modyfikator `while`
- `begin / end / while`
- `until / end`, modyfikator `until`
- `for / in / end`
- wcześniejsze kończenie pętli: `break`, `next`, `redo`



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Podstawowe elementy języka: **metody (funkcje)**

- `def / end`, domyślne wartości parametrów
- wartość (wartości) zwracane
- zmienna liczba parametrów
- `alias` i `undef`



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Podstawowe elementy języka: **klasy**

- obiektość: *data encapsulation, data abstraction, polymorphism, inheritance*
- `class Identifier / end`
- `obiekt = ClassName.new`
- zmienne lokalne, zmienne instancji, zmienne klasowe, zmienne globalne
- metody obiektu, klasy



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Korzystamy z *Gemów*

- Bundler – co to jest? <http://bundler.io/>
- pliki `Gemfile` i `Gemfile.lock`
- Bundler i RVM (Ruby Version Manager)

Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

hello_bundler.rb

```
#!/bin/env ruby
```

```
require 'bundler/setup'
```

```
require 'nokogiri' # http://nokogiri.org/  
require 'open-uri'
```

```
doc =  
Nokogiri::HTML(open('http://www.google.com/search?q=rspec'))  
  
doc.css('h3.r a').each do |link|  
  puts link.content  
end
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



**Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego**

Wykład 2

Podstawowe klasy i moduły



Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Klasa *Array*

Dokumentacja (Ruby Core):

<http://www.ruby-doc.org/core-2.1.2/Array.html>

Składnia:

```
a = [4, 8, 2]; b = -10...0, 0..10]; c = [[1,2], [3,4]]  
words = %w{ała ma kota}
```

Indeksowanie:

```
d = ('a'..'e').to_a  
d[-2,2]; d[0..2]; d[0...-1]
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Klasa *Hash*

Dokumentacja (Ruby Core):

<http://www.ruby-doc.org/core-2.1.2/Hash.html>

Składnia (Ruby 2.0.0+):

```
liczby = {one: 1, two: 2, three: 3}
liczby[:one]
liczby2 = {"one" => 1, "two" => 2, "three" => 3}
liczby2["one"]
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Moduł *Enumerable*

Dokumentacja (Ruby Core):

<http://www.ruby-doc.org/core-2.1.2/Enumerable.html>

Pętle:

```
(2..4).each { |x| print x }  
(2..4).each_with_index { |x,i| print x,i }  
(1..10).each_slice(4) { |x| priny x }
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Sortowanie:

```
w = %w{wiśnia gruszka czereśnia}
```

```
w.sort
```

```
w.sort { |a,b| b<=>a }
```

```
w.sort { |a,b| a.size<=>b.size }
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Klasy *Date*, *Time* i *DateTime*

Tworzenie obiektu *Time*. *wyklady/2/10-hello_time.rb*

Co się składa na czas? *wyklady/2/20-hello_time.rb*

Strefy czasowe. *wyklady/2/30-hello_time.rb*

Formatowanie *Time* i *Date*. *wyklady/2/40-hello_time.rb*

Parsowanie *Time* i *Date*. *wyklady/2/50-hello_time.rb*

Arytmetyka i porównywanie *Time*. *wyklady/2/60-hello_time.rb*



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Dokumentacja (*Ruby StdLib*):

<http://www.ruby-doc.org/stdlib-2.1.2/libdoc/date/rdoc/Date.html>

<http://www.ruby-doc.org/stdlib-2.1.2/libdoc/time/rdoc/Time.html>

<http://www.ruby-doc.org/stdlib-2.1.2/libdoc/date/rdoc/DateTime.html>



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



**Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego**

Wykład 3

Ogólnie o testowaniu.

Dostępne narzędzia i technologie

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Testowanie to proces który ma na celu weryfikację oraz walidację oprogramowania. Weryfikacja oprogramowania ma na celu sprawdzenie, czy wytwarzane oprogramowanie jest zgodne ze specyfikacją. Walidacja sprawdza, czy oprogramowanie jest zgodne z oczekiwaniami użytkownika.

W środowisku Ruby najpopularniejsze framweworki do testowania to:

- RSpec – <https://github.com/rspec>
- Minitest – <https://github.com/seattlerb/minitest>

Minitest jest dostępny w Ruby od wersji 1.9. Dużo „prostszy” niż RSpec. Jednak zapewnia wszystkie elementy potrzebne do testowania: unit, spec, mock, benchmark. RSpec – najpopularniejszy, kompletny system do testowania kodu w Ruby.

Nie można wskazać lepszego czy gorszego, to bardziej kwestia składni i narzędzi. Minitest często wykorzystywany do testowania gemów by nie zwiększać ich zależności



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Plik *Gemfile* dla Rspec

```
group :development do  
  gem 'rspec'  
end
```

A tak uruchamiamy RSpec:

```
$ bundle exec rspec
```

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Testowanie to proces który ma na celu weryfikację oraz walidację oprogramowania. Weryfikacja oprogramowania ma na celu sprawdzenie, czy wytwarzane oprogramowanie jest zgodne ze specyfikacją. Walidacja sprawdza, czy oprogramowanie jest zgodne z oczekiwaniami użytkownika.

W środowisku Ruby najpopularniejsze framweworki do testowania to:

- RSpec – <https://github.com/rspec>
- Minitest – <https://github.com/seattlerb/minitest>

Minitest jest dostępny w Ruby od wersji 1.9. Dużo „prostszy” niż RSpec. Jednak zapewnia wszystkie elementy potrzebne do testowania: unit, spec, mock, benchmark. RSpec – najpopularniejszy, kompletny system do testowania kodu w Ruby.

Nie można wskazać lepszego czy gorszego, to bardziej kwestia składni i narzędzi. Minitest często wykorzystywany do testowania gemów by nie zwiększać ich zależności

Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Korzystamy z gemów Capybara i Factory Girl

Cytat: Factory Girl is a fixtures replacement with a straightforward definition syntax, support for multiple build strategies (saved instances, unsaved instances, attribute hashes, and stubbed objects), and support for multiple factories for the same class (user, admin_user, and so on), including factory inheritance.

Plik *Gemfile*:

```
group :development do
  gem 'rspec'
  gem 'factory_girl'
end
```

Plik *spec_helper.rb*:

```
RSpec.configure do |config|
  config.include FactoryGirl::Syntax::Methods
end
```



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Prosty przykład

```
# This will guess the User class
FactoryGirl.define do
  factory :user do
    first_name "John"
    last_name  "Doe"
    admin false
  end

  # This will use the User class
  # (Admin would have been guessed)
  factory :admin, class: User do
    first_name "Admin"
    last_name  "User"
    admin true
  end
end
```

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Capybara helps you test web applications by simulating how a real user would interact with your app

I dopisujemy `require 'capybara/rspec'`
w pliku z testami lub `spec_helper`.

```
describe "the signin process", :type => :feature do  
  before :each do  
    User.make(:email => 'user@example.com', :password => 'caplin')  
  end
```

Plik *Gemfile*:

```
group :development do  
  gem 'rspec'  
  gem 'capybara'  
end
```

```
  it "signs me in" do  
    visit '/sessions/new'  
    within("#session") do  
      fill_in 'Login', :with => 'user@example.com'  
      fill_in 'Password', :with => 'password'  
    end  
    click_link 'Sign in'  
    expect(page).to have_content 'Success'  
  end  
end
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Użyteczne narzędzia, gemy

- Ruby Version Manager (RVM) lub rbenv
- gem guard
- gem simplecov
- Platforma continuous integration – Travis
- Edytory Atom/Emacs/Vi o ile skonfigurujemy je do pracy z frameworkami



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



**Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego**

Wykład 4

Testy jednostkowe

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Co to jest test jednostkowy?

Zautomatyzowany test pisany przez programistę testujący **pojedynczy** element systemu **w izolacji**. Przykład:

```
describe Array do
  describe '#sum' do
    it 'returns 0 for empty array' do
      expect( [].sum ).to eq(0)
    end
    it 'returns proper sum of all elements' do
      expect( [1, 2, 3, 4, 5].sum ).to eq(15)
    end
    it 'raise error if array include non numeral values' do
      expect{
        [1, 2, 3, 4, 5].sum
      }.to raise_error
    end
  end
end
end
```

```
class Array
  def sum
    reduce(0, :+)
  end
end
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Po co piszemy testy jednostkowe?

- Natychmiastowy feedback.
- Wyraźna lokalizacja błędu.
- Lepszy kod – mniejsze metody, zgodne z filozofia Unix.
(robiące tylko jedną rzecz). Z wyraźnymi granicami –
unikamy stanów globalnych, przekazujemy stan do metody.



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Automatyzacja z pomocą RSpec i Guard

Najprostsze testowanie robi każdy, nawet o tym nie wiedząc.
Kiedy sprawdzamy ręcznie napisany kod, czy to w konsoli
czy przeglądarce.

Taka forma jest nieoptymalna, zabiera dużo czasu –
znudzeni omijamy sprawdzenie wszystkiego za każdym razem.
Automatyzacja pozwala napisać serie skryptów (testów),
które manualne sprawdzanie automatyzują.

Guard pozwala zrezygnować z ostatniej manualnej czynności,
czyli każdorazowego uruchamiania testów ręcznie.
Obserwuje on pliki nad którymi pracujemy i automatycznie
uruchamia powiązane z nimi testy.



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Przykład

Gemfile:

```
group :development do
  gem 'rspec'
  gem 'guard-rspec', require: false
end
```

Bash:

```
$ bundle install
$ bundle exec guard init rspec
$ bundle exec guard
```

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Testujemy: **test-first** czy **test-last**?

Nie ma jednej właściwej odpowiedzi na to pytanie.
Każde z podejść ma swoje plusy i minusy.

Test first – daje nam 100% pokrycie testami. (Każdy kod jaki dodajemy do aplikacji poprzedzony jest napisanymi do niego testami.)
Pojawia się dużo testów niskopoziomowych (unitów), stosunkowo mniej integracyjnych. Dodawane by sprawdzić połączenia między unitami.

Przeciwnicy najbardziej zwracają uwagę na to, że piszemy testy do kodu, który nie do końca rozumiemy. Pisanie zajmuje więcej czasu, ale kod otrzymany na końcu jest zazwyczaj lepszy.

Najlepszą odpowiedzią na pytanie czy test-first czy też test-last jest zrozumienie problemu. Jeśli doskonale wiemy co nasz kod ma robić. Rozumiemy go, to najlepiej zacząć od testów, dodając kod później. W przeciwnym wypadku, najlepiej sprawdza się drugie podejście.

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Piszemy kod: **top-down** czy **bottom-up**?

Zazwyczaj ta decyzja nie od nas zależy.

Top-down jest spotykany zazwyczaj w dużych projektach, zespołach (korporacjach), gdzie inni ludzie projektują system, a inni go piszą. Developer wtedy zazwyczaj dostaje do napisania konkretny kawałek systemu, który został zaprojektowany przez kogoś innego.

Bottom-up występuje w sytuacji gdy z małych części systemu budujemy złożony produkt. Często na początku nie znając dokładnego kształtu systemu. Z małych klocków budujemy złożone systemy

https://en.wikipedia.org/wiki/Top-down_and_bottom-up_design



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Mantra **Test Driven Development** (TDD)

Red ⇒ **Green** ⇒ **Refactor**

Red – napisz test, który nie przechodzi

Green – dopisz kod, tak aby test przeszedł

Refactor – wyeliminuj nadmiarowy/zbędny kod

W trakcie pisania kodu nasze zrozumienie problemu i co traktować jako „unit” ewoluują.

Kiedy zmieniamy kod, musimy zmienić też testy.



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Izolacja

Jak osiągamy izolację?

Wyznaczamy wyraźne granice między elementami systemu
Nie korzystamy z zewnętrznych części aplikacji czy systemu,
np. baza danych, system plików

Każda część (unit) w aplikacji powinien wykonywać jedną rzecz,
powinien być oddzielony od reszty systemu wyraźną granicą.

Powinien mieć ustawiony stan na wejściu operować na nim
i ustawiać stan wyjściowy. Ten stan wejściowy mockujemy
(sztucznie go ustawiamy), by móc go przetestować w izolacji
i we wszystkich możliwych przypadkach.

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Jak testowany jest kod Ruby Core i Std-lib?

Przykładami dobrego testowania i kodu są projekty zgromadzone w tej bazie:

<http://microrb.com>

Jest tam dużo micro projektów – małych bibliotek, przetestowanych i napisanych profesjonalnie.

Ruby Core: <https://github.com/ruby/ruby/tree/trunk/test/ruby>

Array: https://github.com/ruby/ruby/blob/trunk/test/ruby/test_array.rb

Hash: https://github.com/ruby/ruby/blob/trunk/test/ruby/test_hash.rb

Ruby Std-lib:

Date, DateTime, Time: <https://github.com/rubysl/rubysl-date/tree/2.0/spec>

StringScanner: <https://github.com/rubysl/rubysl-strscan/tree/2.0/spec>



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



**Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego**

Wykład 5

Refaktoryzacja kodu

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Na czym polega refaktoryzacja kodu?

Red -> Green -> Refactor

Program wyliczający dla każdego klienta opłaty za wypożyczenie filmu (przykład z książki „Refactoring. Ruby Edition”).

Plik *wyklad/5/00-wypożyczalnia.rb*: klasy Movie, Rental, Customer

Uwagi odnośnie kodu: „quick and dirty program”.
Słabo zaprojektowany. Mało obiektowy.

Dlaczego? Przekonamy się o tym jak tylko zaczniemy zmieniać kod.
Na przykład, dodamy nowe results w HTML, zmieniają się zasady naliczania opłat .
Wtedy do poprawki będą oba results), zmieniają się / dojdą nowe kategorie filmów.

Zanim zaczniemy refaktoryzację kodu powinniśmy dopisać testy!

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Refaktoryzacja #1 (pierwszy krok)

Dekompozycja i redystrybucja metody *statement*.
Dlaczego? za długi kod. Mniejszymi kawałkami łatwiej zarządzać.

Plik: *wyklad/5/01-wypożyczalnia.rb*

Bash:

```
diff -U 0 00-wypożyczalnia.rb 01-wypożyczalnia.rb
```

Zastosowane techniki:

Extract Method do instrukcji case .

Jak to zrobić w bezpieczny sposób bez wprowadzenia bugów do kodu?

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Instrukcję case zamieniamy na `this_amount = amount_for(element)`

```
this_amount = 0
case element.movie.price_code
when Movie::Regular
  this_amount += 2
  this_amount += (element.days_rented - 2) * 1.5 \
    if element.days_rented > 2
when Movie::NEW_RELEASE
  this_amount += element.days_rented * 3
when Movie::CHILDRENS
  this_amount += 1.5
  this_amount += (element.days_rented - 3) * 1.5 \
    if element.days_rented > 3
end
```

W oryginalnym kodzie zmieniamy nazwy zmiennych:
`element -> rental, this_amount -> result`



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Refaktoryzacja #2

Przenosimy obliczenia `amount_for` do klasy `Rental`.

Dlaczego?

Korzysta z informacji z `Rental` i nie korzysta – z `Customer`.

Plik: *wyklad/5/02-wypożyczalnia.rb*

Bash:

```
diff -U 0 01-wypożyczalnia.rb 02-wypożyczalnia.rb
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Klasy Rental i Customer (tylko zmiany w kodzie)

```
class Rental
  def charge
    ...
  end
end
class Customer
  def amount_for(rental)
    rental.charge
  end
end
```

Po napisaniu metody charge i sprawdzeniu,
że niczego nie popsuliśmy, starą metodę amount_for
zastępujemy nową charge i this_amount (jest zbędne).

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Refaktoryzacja #3

Ekstrakcja frequent_renter_points:
przenosimy z klasy Customer do klasy Rental

```
frequent_renter_points += 1  
if element.movie.price_code = \  
    Movie::NEW_RELEASE && element.days_rented > 1  
    frequent_renter_points += 1  
end
```

Plik: *wyklad/5/03-wypożyczalnia.rb*



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIWERSYTET GDAŃSKI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Klasy Rental i Customer

```
class Rental
  def frequent_renter_points
    movie.price_code = \
      (Movie::NEW_RELEASE && days_rented > 1) ? 2 : 1
  end
end
class Customer
  ...
  @rentals.each do |element|
    frequent_renter_points += element.frequent_renter_points
  end
  ...
end
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Refaktoryzacja #4

Niebezpieczne zmienne tymczasowe:

użyteczne w kontekście metody,
zachęcają do pisania długich skomplikowanych metod

Usuniemy dwie zmienne tymczasowe:

`total_amount` i `frequent_renter_points`

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Zamieniamy `total_amount` metodą prywatną `total_charge`

Plik: *wyklad/5/04-wypożyczalnia.rb*

To nie jest prosty przypadek eliminacji zmiennej tymczasowej –
musimy wykonać jeszcze raz pętlę.

```
def total_charge
  # result = 0
  # @rentals.each do |element|
  #   result += element.charge
  # end
  # result
  @rentals.inject(0) {|sum, rental| sum += rental.charge}
end
```

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Usuwamy zmienną tymczasową `frequent_renter_points`

Postępujemy tak samo jak w wypadku `total_amount` powyżej.

Plik: *wyklad/5/05-wypożyczalnia.rb*

Bonus: dodajemy metodę `Customer#html_statement`

Plik: *wyklad/5/06-wypożyczalnia.rb*



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Jaki kod wymaga refaktoryzacji?

ze zduplikowanym kodem
metody z dużą ilością kodu
duże klasy
długa lista argumentów
„shotgun surgery” – zmiana w jednym miejscu pociąga za sobą drobne zmiany w wielu innych miejscach
„divergent change” – przeciwieństwo „shotgun surgery” –
„feature envy” – metoda korzystająca z wielu metod innej klasy
przypadek instrukcji case – czasami można użyć polimorfizmu
do wyeliminowania tej instrukcji – „message chains” –
łańcuchy wywołań – szaleństwo metaprogramowania

Co to jest **bad smells** w kodzie?

W literaturze spotyka się też określenie **smoked code**.



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Wykład 6

Obiekty *doubles*,

Zwane też *mockami* lub *stubami*



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Doubles, zwane też **mockami** lub **stubami**

Kod który piszemy zazwyczaj nie ma wyraźnych granic i często wydzielić z niego pojedyncze jednostki nie zależne od innych części systemu.

By to osiągnąć posługujemy się doubles,
by podmienić części systemu na „sztuczny kod”.

Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Kiedy używamy **doubles**?

In a unit test, mock objects can simulate the behavior of complex, real objects and are therefore useful when a real object is impractical or impossible to incorporate into a unit test.

If an actual object has any of the following characteristics, it may be useful to use a mock object in its place.

Jeśli rzeczywisty obiekt:

- supplies non-deterministic results (e.g., the current time or the current temperature);
- has states that are difficult to create or reproduce (e.g., a network error);
- is slow (e.g., a complete database, which would have to be initialized before the test);
- does not yet exist or may change behavior; would have to include information and methods exclusively for testing purposes (and not for its actual task).

https://en.wikipedia.org/wiki/Mock_object



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Prawo Demeter

– zasada minimalnej wiedzy lub Reguła ograniczania interakcji
(„rozmawiaj tylko z (bliskimi) przyjaciółmi”)

W pełnej formie mówi ono, iż metoda danego obiektu może odwoływać się jedynie do metod należących do:

- metod tego samego obiektu,
- metod dowolnego parametru przekazanego do niej,
- dowolnego obiektu przez nią stworzonego,
- dowolnego składnika, klasy do której należy dana metoda.

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

stub, mock, fake?

- stub – obiekt zawierający informację o stanie końcowym, pożądanym lub faktycznym
- mock – obiekt zawierający informacje o zachowaniu, sposobie dotarcia do tego stanu końcowego
- fake – klasa zawierająca metody, które zwracają ściśle określone wartości, np. wpisane na sztywno w kod klasy

Innymi słowy:

stub – umożliwia testowanie stanu

mock – umożliwia testowanie zachowania

<http://martinfowler.com/articles/mocksArentStubs.html>



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Testy integracyjne

Testy jednostkowe, testują pojedyncze unity (jednostki),
testując je w izolacji nie wiemy czy między sobą działają dobrze.

Testy integracyjne testują właśnie te połączenia między unitami.
Zbierają kilka części systemu i sprawdzają interakcje między nimi.



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



**Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego**

Testy widoku, żądania

Testy symulujące prawdziwe zachowanie użytkownika.
Symulują jego zachowanie, nie testują kodu jako takiego
a zewnętrzny interfejs i jego zachowanie.

Frameworkiem do pisania takich testów w środowisku
Ruby jest Capybara.



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Wykład 7

Refaktoryzacja *Legacy Code*

(zaawansowane techniki refaktoryzacji kodu)



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Zaczynamy od kodu z pliku: *wyklady/7/06-wypożyczalnia.rb*

W klasie Rental w instrukcji case widzimy warunek korzystający z wartości atrybutu obiektu innej klasy.

Pachnie to problemami na odległość.



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Dlatego przenosimy Rental#charge do Movie#charge

```
class Rental
  def charge
    movie.charge(days_rented)
  end
end
class Movie
  def charge(days_rented)
    result = 0
    ... wklejamy case bez zmian
  end
end
```

Plik: *wyklady/7/07-wypożyczalnia.rb*

Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Przenosimy Rental#frequent_renter_points do Movie

```
class Rental
  def frequent_renter_points
    movie.frequent_renter_points(days_rented)
  end
end
class Movie
  def frequent_renter_points(days_rented)
    (price_code == Movie::NEW_RELEASE && days_rented > 1) \
      ? 2 : 1
  end
end
```

Plik: *wyklady/7/08-wypożyczalnia.rb*

Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Wymieniamy case korzystając z **polymorfizmu**

Każdy gatunek filmu jest nieco inaczej obsługiwany
(inaczej odpowiada na tę samą message).

Usuniemy instrukcję case korzystając ze **State Pattern**
(wzorzec projektowy z książki autorów określanych jako Gang of Four).



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Krok #1

```
class Movie
  attr_reader :price_code
  def price_code=(value)
    @price_code = value
  end
  def initialize(title, the_price_code)
    @title, self.price_code = title, the_price_code
  end
end
```

Plik: *wyklady/7/09-wypożyczalnia.rb*

Po tych zmianach wszystko powinno działać:

```
$ ruby 09-wypożyczalnia.rb
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Dodajemy trzy nowe klasy:

RegularPrice, NewReleasePrice, ChildrenPrice

I zmieniamy kod metody price_code=

```
class Movie
  def price_code=(value)
    @price_code = value
    @price = case price_code
              when REGULAR then RegularPrice.new
              when NEW_RELEASE then NewReleasePrice.new
              when CHILDRENS then ChildrenPrice.new
            end
  end
end
```

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Wybieramy metodę, która ma być polimorficzna:

Movie#charge

```
class RegularPrice
  def charge(days_rented)
    result += 2
    result += (days_rented - 2) * 1.5 if days_rented > 2
    result
  end
end
class Movie
  def charge(days_rented)
    result = 0
    case price_code
    when Movie::REGULAR
      @price.charge(days_rented)
```



Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

W podobny sposób upraszczamy instrukcję case dla
Movie::NEW_RELEASE oraz Movie::CHILDRENS

```
def charge(days_rented)
  result = 0
  case price_code
  when Movie::REGULAR
    return @price.charge(days_rented)
  when Movie::NEW_RELEASE
    return @price.charge(days_rented)
  when Movie::CHILDRENS
    return @price.charge(days_rented)
  end
end
```

Co można uprościć do:

Plik: *wyklady/7/10-wypożyczalnia.rb*

```
def charge(days_rented)
  @price.charge(days_rented)
end
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Polymorficzne Movie#frequent_renter_points

Metoda frequent_renter_points jest taka sama dla
ChildrenPrice i RegularPrice, a inna dla NewReleasePrice.

Poniżej użyjemy modułów i domieszkowania.

Plik: *wyklady/7/11-wypożyczalnia.rb*

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Moduły i domieszkowanie kodu modułów w klasach

```
module DefaultPrice
  def frequent_renter_points(days_rented)
    1
  end
end
class RegularPrice
  include DefaultPrice
  ...
class ChildrenPrice
  include DefaultPrice
  ...
class NewReleasePrice
  def frequent_renter_points(days_rented)
    days_rented > 1 ? 2 : 1
  end
  ...
class Movie
  def frequent_renter_points(days_rented)
    @price.frequent_renter_points(days_rented)
  end
  ...
```


Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Ostatni krok...

Usuujemy case z `price_code=.` Jak? `wyklady/7/12-wypożyczalnia.rb`

Podmieniamy przykład użycia:

```
movie1 = Movie.new("Milion sposobów, jak zginąć na Zachodzie", \
  NewReleasePrice.new)
movie2 = Movie.new("Uśpieni", ChildrenPrice.new)
```

Teraz można też zmienić kategorię filmu:

```
movie2.price = RegularPrice.new
```

Poprzednio nie było to możliwe!



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



**Projekt *Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)*
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego**

Wykład 8

Praca z Legacy Code

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Legacy Code definiowany jest różnie.
Zazwyczaj jako „zastany kod”. Kod który napisał ktoś inny
(którego nie jesteśmy autorami) a z którym musimy pracować.

Znaczy to, że nie mamy pełnej wiedzy jak dokładnie każdy element systemu
działa i nie czujemy się komfortowo by go rozszerzać i zmieniać.

Michael Feathers, autor „Working Effectively with Legacy Code”,
idzie krok dalej i definiuje *Legacy Code* jako kod nie pokryty testami.

Nie jest ważne kto jest autorem kodu, nawet jeśli sami go napisaliśmy,
to on się rozrasta i komplikuje, a po pewnym czasie tracimy pewność
siebie jeśli mamy coś w nim zmienić lub go rozszerzyć.

Testy, dają nam pewność, że jakkolwiek zmiana dokonana w systemie
nie spowoduje, że przestanie on działać.

Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Jak sobie radzić z legacy code?

Każdy w pewnym momencie swojej kariery programistycznej, spotyka się z sytuacją w której zostajemy zmuszeni do pracy z legacy code. Ktoś z zespołu odszedł i trzeba przejąć jego kod. Zostaliśmy wynajęci by dopisać do istniejącej aplikacji nową funkcjonalność...

Jak sobie z taką sytuacją poradzić?

Zazwyczaj będziemy bali się cokolwiek zmieniać, będziemy ograniczać naszą interakcje w kodzie do minimum, by „nic nie zepsuć”.

Nie jest to dobre podejście. Raczej powinniśmy taki kod pokryć testami, by zyskać pewność że działa tak jak powinien, przed i po zmianach

Zaczynamy od testów integracyjnych, wysokiego poziomu, które w sposób ogólny opiszą działanie systemu, i powoli „schodzić” niżej dodając testy jednostkowe, które pomogą nam refaktoryzować kod.



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



**Projekt Program rozwoju Uniwersytetu Gdańskiego w obszarach Europa 2020 (UG2020)
jest współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego**

Kiedy będziemy mieli już kod pokryty testami, możemy przejść do refaktoryzacji kodu (rozbić duże metody, wyeksportować wspólne kawałki kodu, dodać wyraźne granice, oraz przejść do dodawania nowych funkcjonalności.

Początek może być trudny, ale następnym razem jeśli my lub ktoś inny będzie musiał pracować przy tym kodzie, będzie miał już otestowany kod i będzie mógł od razu przejść do pracy nad nim.

W programowaniu jak i w życiu warto pamiętać o harcerskiej zasadzie:

Zawsze zostawiaj miejsce (biwaku) lepszym niż było.

Tak samo w programowaniu, zawsze pozostawiaj kod lepszym niż zastałeś.

