

Lesson 3 Ruby task

Task: Module-parser

Description:

Реализовать **Ruby** модуль для парсинга определенных данных следующей архитектуры:

Условия:

- Все должно быть заскоуплено в модуле **Parser**
- Внутри модуля должен быть класс **Base** с общим функционалом для всех кастомных парсеров
- В **Base** должна быть константа **TIMEOUT_LIMIT** со значением по-умолчанию 1 секунда. Данная константа может быть переопределена в каждом кастомном классе
- Каждый кастомный парсер должен наследоваться от **Base**
- Во всех классах публичными должны быть только перечисленные методы, вся остальная реализация должна быть сокрыта.
- Список доступных публичных методов (за исключением стандартных):
 - **parse**
 - **report**
 - **search_key**
 - **search_key=(...)**
- Обзор парсеров:
 - **class Base**: базовый парсер, принимает на вход текст и подстроку, его **report** возвращает количество найденных совпадений, а **search_key** позволяет изменить подстроку для поиска.

Пример работы:

```
base = Parser::Base.new("hello world\n0ko", 'o')
puts base.parse.report # => 3
base.search_key = 'world'
base.parse
puts base.report # => 1
```

- **class Link**: парсер который достает все ссылки из переданного текста, принимает только текст. Пример работы:

```
link = Parser::Link.new("<div><a href='x.com'>x</a></div>\n<a href='y.com'>y</a>")
link.parse
p link.report # => ["x.com", "y.com"]
```

- **class Custom**: парсер который реализует построчный поиск переданного текста произвольным образом, принимает только текст. Пример работы:

```
custom = Parser::Custom.new("hello <a href='tut.by'>tost</a> world\nhello <a href='tut.by'>tost2</a> world")
print( custom.parse do |line|
  line.match(/(tost\w*)</)/[1]
end.report ) # => ["tost", "tost2"]
```

В случае возникновения ошибки при парсинге, метод **parse** не должен падать, а метод **report** в таком случае возвращает соответствующее классу нулевое значение: `0` – для **Base**, `[]` – для всех остальных.

```
custom = Parser::Custom.new("abs")
custom.parse { |line| line.match(/(\d+)/)[1] }
p custom.report # => []
```