Porovnávanie efektivity rôznych jazykov vo výuke programovania - Plán a postup páce

Ján Kelemen

Fizz buzz

Difficulty: easy

Description: Fizz buzz is a group word game for children to teach them about division. Players take turns to count incrementally, replacing any number divisible by... more

Added on: 31.03.22, 09:44:31

Get first 100 primes

Difficulty: easy

Description: Rewrite already an existing solution in Go into Python

Added on: 31.03.22, 09:44:31

Choose filters

☐ show only solutions that didn't fail
☐ show only final tests

First language: go Deselect language
Select from solutions: 07.04.22, 13:55:31 Change name Minimize/maximize
Select from test: 31.03.22, 09:44:31 (final) Change name Minimize/maximize

Second language: Deselect language

Description

Fizz buzz is a group word game for children to teach them about division. Players take turns to count incrementally, replacing any number divisible by three with the word "fizz", and any number divisible by five with the word "buzz".

Solution 1

```go
package main

import (
    "strconv"
)

func FizzBuzz1_000_000() []string {
    res := make([]string, 0, 1_000_000)
    for i := 1; i <= 1_000_000; i++ {
        if i%3 == 0 && i%5 == 0 {
            res = append(res, "fizzbuzz")
        } else if i%3 == 0 {
            res = append(res, "fizz")
        } else if i%5 == 0 {
            res = append(res, "buzz")
        } else {
            res = append(res, strconv.Itoa(i))
        }
    }
    return res
}

```

Test 1

```go
package main

import "testing"

func TestFizzBuzz1000(t *testing.T) {
    res := FizzBuzz1_000_000()

    tests := []struct {
        name      string
        expecting string
        got       string
    }{
        {name: "on index 14", expecting: "fizzbuzz", got: res[14]},
        {name: "on index 100_000", expecting: "100001", got: res[100_000]},
        {name: "on index 999_999", expecting: "buzz", got: res[999_999]},
    }
    for _, test := range tests {
        t.Run(test.name, func(t *testing.T) {
            if test.expecting != test.got {
                t.Errorf("%s got: %q, expecting: %q", test.name, test.got, test.ex
            }
        })
    }
}
```

Choose filters

☐ show only solutions that didn't fail
☐ show only final tests

First language: go | Deselect language
Select from solutions: 31.03.22, 09:44:31 | Change name | Minimize/maximize
Select from test: 31.03.22, 09:44:31 (final) | Change name | Minimize/maximize

Second language: python | Deselect language
Select from solutions: 31.03.22, 09:44:31 | Change name | Minimize/maximize
Select from test: 31.03.22, 09:44:31 (final) | Change name | Minimize/maximize

Description

Rewrite already an existing solution in Go into Python

Solution 1

```go
package main

import "math"

func primes() (res []int) {
    isPrime := func(n int) bool {
        limit := int(math.Pow(float64(n),
        for i := 2; i < limit; i++ {
            if n%i == 0 {
                return false
            }
        }
        return true
    }

    for i := 1; len(res) <= 100; i++ {
        if isPrime(i) {
            res = append(res, i)
        }
    }
    return
}
```

Test 1

```go
package main

import "testing"

func TestPrimes(t *testing.T) {
    got := primes()

    tests := []struct {
        name      string
        expecting int
        got       int
    }{
        {name: "on index 1", expecting: 2,
        {name: "on index 9", expecting: 23,
        {name: "on index 89", expecting: 46
    }
    for _, test := range tests {
        t.Run(test.name, func(t *testing.T)
            if test.expecting != test.got {
                t.Errorf("%s got: %d, expec
            }
        })
    }
```

Solution 2

```python
def is_prime(n):
    for i in range(2, int(n**1 /
        if n % i == 0:
            return False
    return True

def primes():
    res = []
    i = 1
    while len(res) < 100:
        if is_prime(i):
            res.append(i)
        i += 1
    return res
```

Test 2

```python
def test_primes():
    got = primes()
    assert got[1] == 2
    assert got[9] == 23
    assert got[89] == 461
```

Run 1. language

running

test OK

compilation time: 0.23 s

real time: 0 s

kernel time: 0 s

user time: 0 s

max ram usage: 2.56 mb

binary size: 2.93 mb

test output:

```
=== RUN   TestPrimes
=== RUN   TestPrimes/on_index_1
=== RUN   TestPrimes/on_index_9
=== RUN   TestPrimes/on_index_89
--- PASS: TestPrimes (0.00s)
    --- PASS: TestPrimes/on_index_1 (0.00s)
    --- PASS: TestPrimes/on_index_9 (0.00s)
    --- PASS: TestPrimes/on_index_89 (0.00s)
PASS
```

Run 2. language

running

test OK

compilation time: 0 s

real time: 0.23 s

kernel time: 0.02 s

user time: 0.14 s

max ram usage: 23.15 mb

binary size: 0 mb

test output:

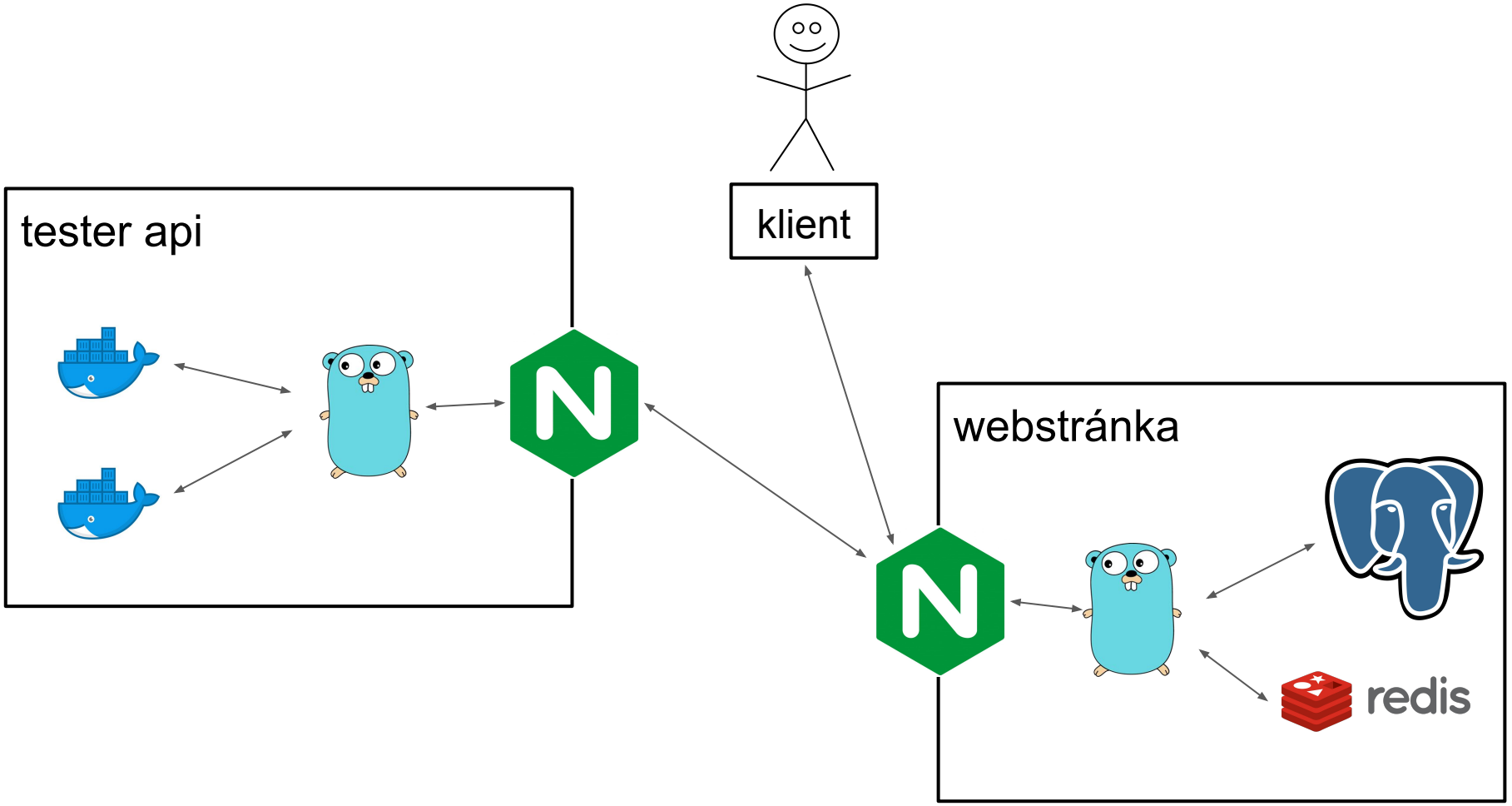
```
============================= test session starts ==============================
platform linux -- Python 3.10.2, pytest-6.2.5, py-1.11.0, pluggy-0.13.1 -- /usr/bin/python
cachedir: .pytest_cache
rootdir: /home/user_solutions/python/3013640126
collecting ... collected 1 item

home/user_solutions/python/3013640126/main.py::test_primes PASSED       [100%]

============================== slowest durations ===============================
0.00s call     main.py::test_primes
0.00s setup    main.py::test_primes
0.00s teardown main.py::test_primes
============================== 1 passed in 0.01s ===============================
```

Plán práce

- do 14.4.
  - mať na stránke 5-10 úloh, ktoré sa budú dať riešiť
- do 17.4.
  - prenajať Ubuntu inštancie na nejakého VPS providera
  - pokúsiť sa urobiť deployment
  - usability testing so spolužiakom
  - pridať stránku so štatistikami, kde si používatelia budú vedieť pozrieť, ako efektívne úlohu vyriešili iní
  - pri neúspešných riešeniach vo výpise vyparsovať len tie testy, ktoré neprešli
- apríl
  - urobiť load balancer na tester-api, aby vedel obsluhovať viac ako jeden request naraz
  - pridať medzi jazyky C++ a Javascript (momentálne tam je len Go a Python)
  - ukladať zbehnuté úlohy v Redise, aby sa dva krát nemuselo to isté riešenie testovať
- veci, ktoré by bolo dobré urobiť ak zvýši čas
  - komunikáciu medzi webstránkou a tester-api riešiť cez grpc, nie rest api


klient
tester api
webstránka
redis

Ďakujem za pozornosť