Лабораторная работа № 0.1

Разработка простейшего web-сервера

Рассматривается задача разработки web-сервера на языке GO на основе пакета net/http.

Пример реализации простого web-сервера

```
package main

import (
    "fmt" // пакет для форматированного ввода вывода
    "net/http" // пакет для поддержки HTTP протокола
    "strings" // пакет для работы с  UTF-8 строками
    "log" // пакет для логирования
)

func HomeRouterHandler(w http.ResponseWriter, r *http.Request) {
    r.ParseForm() //анализ аргументов,
    fmt.Println(r.Form)  // ввод информации о форме на стороне сервера
    fmt.Println("path", r.URL.Path)
    fmt.Println("scheme", r.URL.Scheme)
    fmt.Println(r.Form["url_long"])
  for k, v := range r.Form {
     fmt.Println("key:", k)
    fmt.Println("val:", strings.Join(v, ""))
 }
 fmt.Fprintf(w, "Test!") // отправляем данные на клиентскую сторону
}

func main() {
 http.HandleFunc("/", HomeRouterHandler) // установим роутер
 err := http.ListenAndServe(":9000", nil) // задаем слушать порт
  if err != nil {
     log.Fatal("ListenAndServe: ", err)
 }
}
```

Задача 1: Реализовать web-сервер и запустить на заданном порте.

Задача 2: Изучить принимаемые web-сервером параметры, реализовать передачу данных методом GET.

Задача 3: Реализовать вывод форматированного гипертекста с контекстным меню в виде гиперссылок, при клике на гиперссылку должна выполняться подмена контента;

Лабораторная работа № 0.2

Разработка приложения обработки данных из RSS-канала

Рассматривается задача разработки приложения на языке GO реализующего синтаксический разбор XML файла формата RSS.

Для реализации данной задачи можно использовать библиотеку rss-parser-go, которая доступна по адресу https://github.com/masterjk/rss-parser-go.

Пример использования библиотеки

```
package main


import (
 "fmt"
 "github.com/RealJK/rss-parser-go"
)


func main() {


rssObject, err := rss.ParseRSS("http://somehost.ru/rss_vk.xml")
if err != nil {


                fmt.Printf("Title          : %s\n", rssObject.Channel.Title)
```

```
                fmt.Printf("Generator       : %s\n", rssObject.Channel.Generator)
                fmt.Printf("PubDate         : %s\n", rssObject.Channel.PubDate)
                fmt.Printf("LastBuildDate   : %s\n", rssObject.Channel.LastBuildDate)
                fmt.Printf("Description     : %s\n", rssObject.Channel.Description)
                fmt.Printf("Number of Items : %d\n", len(rssObject.Channel.Items))

                for v := range rssObject.Channel.Items {
                        item := rssObject.Channel.Items[v]
                        fmt.Println()
                        fmt.Printf("Item Number : %d\n", v)
                        fmt.Printf("Title       : %s\n", item.Title)
                        fmt.Printf("Link        : %s\n", item.Link)
                        fmt.Printf("Description : %s\n", item.Description)
                        fmt.Printf("Guid        : %s\n", item.Guid.Value)
                        }
        }
}
```

Замечание : Библиотеки для работы с RSS-форматом:

- https://github.com/mmcdole/gofeed
- https://github.com/SlyMarbo/rss
- https://github.com/IzeBerg/rss-parser-go

Задача: Реализовать получение данных из различных RSS- каналов по вариантам. Сравнить результаты разбора и сделать выводы.

Варианты

п/н	Вариант RSS-канала
1	http://www.rssboard.org/files/sample-rss-2.xml
2	https://lenta.ru/rss
4	https://news.mail.ru/rss/90/

5 https://vz.ru/rss.xml
6 http://www.kommersant.ru/RSS/main.xml
7 http://www.kommersant.ru/RSS/news.xml
8 https://news.google.com/atom?topic=t&hl=ru&gl=RU&ceid=RU:ru
9 https://www.aviaport.ru/digest/press-releases/rss/
10 https://www.bragazeta.ru/feed/
11 https://news.rambler.ru/rss/Ivanovo/
12 https://vesti-k.ru/rss/
13 https://tvsamara.ru/rss/
14 https://vmo24.ru/rss
15 https://mosreg.ru/sobytiya/novosti?format=rss
16 https://www.mos.ru/rss
17 https://www.interfax.ru/rss.asp
18 https://neftegaz.ru/export/yandex.php
19 https://news.rambler.ru/rss/Namibia/
20 http://mir-la.com/rss.xml
21 https://naked-science.ru/article/category/sci/feed
22 https://briansk.ru/rss20_briansk.xml
23 https://news.rambler.ru/rss/Magadan/
24 https://ldpr.ru/rss
25 https://kinolexx.ru/rss
24 http://www.msk-times.ru/feed.php
26 https://www.press-line.ru/feed
28 http://static.feed.rbc.ru/rbc/logical/footer/news.rss
29 https://rospotrebnadzor.ru/region/rss/rss.php?rss=y
30 https://news.rambler.ru/rss/technology/
31 https://news.rambler.ru/rss/Guadeloupe/

Лабораторная работа № 0.3

Разработка web-ориентированного клиент-серверного приложения получения и представления данных из RSS-канала

Целью данной лабораторной работы является произвести интеграцию результатов работ проведенных в лабораторной работе № 0.1 и лабораторной работе № 0.2.

Задача: необходимо разработать web-сервер, который выполняет соединение с удаленным (удаленными) серверами RSS-новостей и возвращает результаты обработки данных в структурированном виде (страница гипертекста) web-клиенту, в нашем случае в браузер по вариантам.