Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 2

по курсу «Компьютерные сети»

«Клиент и сервер HTTP»

Студент группы ИУ9-31Б Горбунов А. Д.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

Формирование списка фильмов с https://www.afisha.ru/msk/cinema/

2 Результаты

Исходный код программы представлен в листинге 1, 2, 3, 4

Листинг 1 — download.go

```
package main
import (
  "github.com/mgutz/logxi/v1"
  "golang.org/x/net/html"
  "net/http"
  "fmt"
)
func getAttr(node *html.Node, key string) string {
  for _, attr := range node.Attr { if attr.Key == key { return attr.Val } }
  return ""
}
func getChildren(node *html.Node) []*html.Node {
  var children []*html.Node
  for c := node.FirstChild; c != nil; c = c.NextSibling {
    children = append(children, c)
  }
  return children
}
func isElem(node *html.Node, tag string) bool {
  return node != nil && node.Type == html.ElementNode && node.Data == tag
}
func isText(node *html.Node) bool {
  return node != nil && node.Type == html.TextNode
}
func isDiv(node *html.Node, class string) bool {
  return isElem(node, "div") && getAttr(node, "class") == class
}
type Item struct {  Ref, Time, Title string }
func readItem(item *html.Node) *Item {
  if a := item.FirstChild; isElem(a, "a") {
    if cs := getChildren(a); len(cs) == 2 && isElem(cs[0], "time") && isText(cs[1]) {
      return &Item{
        Ref:   getAttr(a, "href"),
        Time:  getAttr(cs[0], "title"),
        Title: cs[1].Data,
      }
    }
  }
  return nil
}
```

Листинг 2 — download(продолжение 1).go

```
func search(node *html.Node) []*Item {
  if isDiv(node, "oby5F") {
    var items []*Item
    for a := getChildren(node)[0]; a != nil; a = a.NextSibling {
      if isDiv(a, "wsXlA pZeTF SUiYd") {
        for b := getChildren(a)[0]; b != nil; b = b.NextSibling{
          if isDiv(b, "wsXlA pZeTF IS7Va") {
            for c := getChildren(b)[0]; c != nil; c = c.NextSibling{
              if isDiv(c, "wsXlA pZeTF CxVPH") {
                for d := getChildren(c)[0]; d != nil; d = d.NextSibling{
                  if isDiv(d, "naAYr") {
          for e := getChildren(d)[0]; e != nil; e = e.NextSibling{
          if getAttr(e, "style") == "height:100%;max-height:48px" {
          for f := getChildren(e)[0]; f != nil; f = f.NextSibling{
          if getAttr(f, "class")=="vcSoT b6DKO jkBWH f5gWK" {
          for g := getChildren(f)[0]; g != nil; g = g.NextSibling{
                              if isDiv(g, "mQ7Bh"){
          for h := getChildren(g)[0]; h != nil; h = h.NextSibling{
fmt.Print(h.Data, " ", "https://www.afisha.ru" + getAttr(f,"href"), "\n\
    n")
                                    if isText(h){
                                      items = append(items, &Item{
                                        Ref:  getAttr(f,"href"),
                                        Title: h.Data,
                                      })
                                    }
                                  }
                                }
                              }
                            }
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
    return items
  }
  for c := node.FirstChild; c != nil; c = c.NextSibling {
    if items := search(c); items != nil {
      return items
    }
  }
  return nil
}
```

Листинг 3 — download(продолжение 2).go

```
1  func downloadNews() []*Item {
2    log.Info("sending request to www.afisha.ru/msk/cinema")
3    if response, err := http.Get("https://www.afisha.ru/msk/cinema/"); err
       != nil {
4      log.Error("request to www.afisha.ru/msk/cinema failed", "error", err
       )
5    } else {
6      defer response.Body.Close()
7      status := response.StatusCode
8      log.Info("got response from www.afisha.ru/msk/cinema", "status",
       status)
9      if status == http.StatusOK {
10       if doc, err := html.Parse(response.Body); err != nil {
11         log.Error("invalid HTML from www.afisha.ru/msk/cinema", "error",
       err)
12       } else {
13         log.Info("HTML from www.afisha.ru/msk/cinema parsed successfully
       ")
14         return search(doc)
15       }
16     }
17   }
18   return nil
19 }
```

Листинг 4 — server.go

```
package main
import (
  "github.com/mgutz/logxi/v1"
  "html/template"
  "net/http"
)
const INDEX_HTML = `
    <!doctype html>
    <html lang="ru">
        <head>
            <meta charset="utf-8">
            <title>List of films from www.afisha.ru/msk/cinema</title>
            <style>
            a {
                color: black;
                text-decoration: none;
                background-color: yellow;
                border: solid;
                border-radius: 5px;
                text-alight: center;
                hight: 50px;
                weight: 100px;
                font-family: 'Bradley Hand', cursive;
                padding: 5px 10px 5px 10px;
            }
            ul{ list-style-type: none; }
            body{ background-color: grey;}
            </style>
        </head>
        <body>
            <ul>
            {{range .}}
                <li><a href="https://www.afisha.ru{{.Ref}}">{{.Title}}</
    a></li>
                <br>
            {{end}}
            </ul>
        </body>
    </html>
    `
var indexHtml = template.Must(template.New("index").Parse(INDEX_HTML))
func serveClient(response http.ResponseWriter, request *http.Request) {
  path := request.URL.Path
  log.Info("got request", "Method", request.Method, "Path", path)
  if path != "/" && path != "/index.html" {
    log.Error("invalid path", "Path", path)
    response.WriteHeader(http.StatusNotFound)
  } else if err := indexHtml.Execute(response, downloadNews()); err !=
    nil {
    log.Error("HTML creation failed", "error", err)
  } else {
    log.Info("response sent to client successfully")
  }
}
func main() {
  http.HandleFunc("/", serveClient)
  log.Info("starting listener")
  log.Error("listener failed", "error", http.ListenAndServe("
    127.0.0.1:6060", nil))
}
```

Результат запуска представлен на рисунке 1, 2

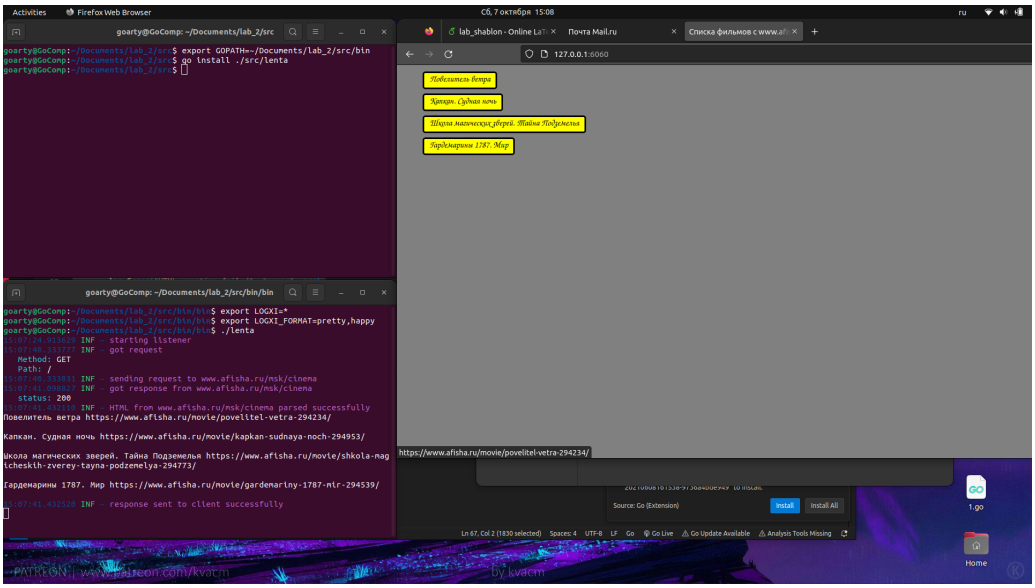

Рис. 1 — Вывод списка фильмов(вид страницы)

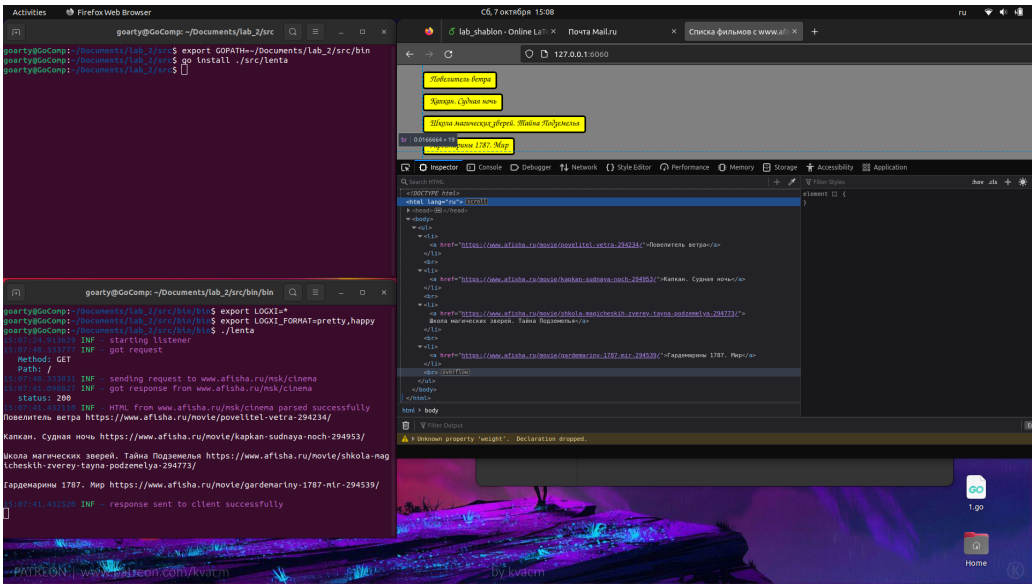

Рис. 2 — Вывод списка фильмов(html-код)