

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «Белгородский государственный
национальный исследовательский университет»**

Лабораторная работа №4
дисциплина «Современные методы, среды и языки программирования»
по теме «Применение порождающих шаблонов при разработке программных систем в Java»

Выполнил: студент группы 12002135
Проверил:

Макаров Д.С.

Лабораторная работа №4

«Применение порождающих шаблонов при разработке программных систем в Java»

Цель работы: Научиться осуществлять выбор порождающих шаблонов проектирования при разработке программных систем, писать код реализации шаблона на языке Java в соответствии с основными принципами объектноориентированного программирования.

Задание D

Паттерн Builder. Имеется текст статьи в формате TXT. Статья состоит из заголовка, фамилий авторов, самого текста статьи и хеш-кода текста статьи. Написать приложение, конвертировать документ в формате TXT в документ формата XML, необходимо также проверять корректность хеш-кода статьи.

Ход работы

Пример работы программы:

Входной файл text.txt

Заголовок/Фамилия И.О.;Фамилия О.И./a95c530a7af5f492 What is Lorem Ipsum? Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Why do we use it? It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

Where does it come from? Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32.

The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections 1.10.32 and 1.10.33 from "de Finibus Bonorum et Malorum" by Cicero are also reproduced in their exact original form, accompanied by English versions from the 1914 translation by H. Rackham.

Where can I get some? There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure there isn't anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making this the first true generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic words et

Результат работы программы text.xml

```
<document>
  <title>Заголовок</title>
  <text>What is Lorem Ipsum?&#xA;Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  → ...</text>
  <author>Фамилия И.О.</author>
  <author>Фамилия О.И.</author>
  <hash>a95c530a7af5f492</hash>
</document>
```

Приложение

Содержимое файла document.go

```
package document

import (
    "crypto/md5"
    "fmt"
)

type Document struct {
    Title, Text string
    Authors    []string
    Hash       [16]byte
}

type builder interface {
    reset()
    SetBody(string)
    SetTitle(string)
    AddAuthor(string)
    SetHash([16]byte)
    CheckHash() bool
    GetDocument() Document
}

// 1 строка заголовок/авторы через запятую/хеш
// все остальное тело статьи
type Builder struct {
    result Document
}

func (b *Builder) SetText(text string) {
    b.result.Text = text
}

func (b *Builder) SetTitle(title string) {
    b.result.Title = title
}

func (b *Builder) AddAuthor(author string) {
    b.result.Authors = append(b.result.Authors, author)
}

func (b *Builder) SetHash(textHash string) {
    hash := []byte(textHash)
    copy(b.result.Hash[:], hash)
}

func (b *Builder) CheckHash() bool {
    hashSum := md5.Sum([]byte(b.result.Text))
    return hashSum == b.result.Hash
}

func (b *Builder) Reset() {
    b.result = Document{}
}

func (b Builder) GetDocument() Document {
    if b.CheckHash() {
        fmt.Println("Document text is correct")
    } else {
        fmt.Println("Document text is incorrect")
    }
    return b.result
}
```

Содержимое файла input.go

```
package input

type Input interface {
    GetText() string
    GetTitle() string
    GetAuthors() []string
    GetHash() [16]byte
}
```

```

        ReadFile(string)
    }

```

Содержимое файла main.go

```

package main

import (
    "lab4/pkg/document"
    "lab4/pkg/input"
    "lab4/pkg/output"
)

func main() {
    fileInput := "text.txt"
    fileOutput := "text.xml"

    textInput := input.TxtInput{}
    textOutput := output.XmlOutput{}

    docBuilder := document.Builder{}

    textInput.ReadFile(fileInput)
    authors := textInput.GetAuthors()
    for _, author := range authors {
        docBuilder.AddAuthor(author)
    }
    docBuilder.SetTitle(textInput.GetTitle())
    docBuilder.SetText(textInput.GetText())
    docBuilder.SetHash(textInput.GetHash())
    doc := docBuilder.GetDocument()

    textOutput.SaveFile(doc, fileOutput)
}

```

Содержимое файла output.go

```

package output

import "lab4/pkg/document"

type Output interface {
    SaveFile(document.Document, string)
}

```

Содержимое файла txtInput.go

```

package input

import (
    "bufio"
    "log"
    "os"
    "strings"
)

type TxtInput struct {
    metaStringList []string
    text           string
}

func (i TxtInput) GetText() string {
    return i.text
}

func (i TxtInput) GetTitle() string {
    return i.metaStringList[0]
}

func (i TxtInput) GetAuthors() []string {
    authorListString := i.metaStringList[1]
    return strings.Split(authorListString, ";")
}

func (i TxtInput) GetHash() string {
    return i.metaStringList[2]
}

```

```

func (i *TxtInput) ReadFile(filename string) {
    file, err := os.Open(filename)
    if err != nil {
        log.Fatal(err)
    }
    reader := bufio.NewReader(file)

    metaString, err := reader.ReadString(byte('\n'))
    if err != nil {
        log.Fatal(err)
    }

    i.metaStringList = strings.Split(metaString, "/")

    text, err := reader.ReadString(byte(0))
    i.text = text

    file.Close()
}

```

Содержимое файла xmlOutput.go

```

package output

import (
    "encoding/xml"
    "lab4/pkg/document"
    "os"
)

type XmlOutput struct{}

func (o XmlOutput) SaveFile(doc document.Document, filename string) {
    type XmlDocument struct {
        XMLName xml.Name `xml:"document"`
        Title   string  `xml:"title"`
        Text    string  `xml:"text"`
        Authors []string `xml:"author"`
        Hash    [16]byte `xml:"hash"`
    }

    xmlDoc := XmlDocument{Title: doc.Title, Text: doc.Text, Authors: doc.Authors, Hash: doc.Hash}
    xmlOut, _ := xml.MarshalIndent(xmlDoc, " ", " ")
    xmlString := string(xmlOut)

    file, err := os.OpenFile(filename, os.O_RDWR|os.O_CREATE, 0664)
    if err != nil {
        panic(err)
    }
    file.WriteString(xmlString)
    file.Close()
}

```