

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

Лабораторная работа №6
дисциплина «Современные методы, среды и языки программирования»
по теме «Применение структурных шаблонов при разработке программных систем в Java»

Выполнил: студент группы 12002135
Проверил:

Макаров Д.С.

Лабораторная работа №6

«Применение структурных шаблонов при разработке программных систем в Java»

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

Задание I

Паттерн Flyweight. Разработать систему учета процессов размножения колонии бактерий.

Ход работы

Пример вывода программы

```
[mda@mda-notebook lab6]$ go run main.go
Count of bacterias 1
Count of bacterias 1
Count of bacterias 1
Count of bacterias 1
Count of bacterias 1
Count of bacterias 1
Count of bacterias 1
Count of bacterias 1
Count of bacterias 1
Count of bacterias 1
Count of bacterias 2
Count of bacterias 2
Count of bacterias 2
Count of bacterias 2
Count of bacterias 2
Count of bacterias 2
Count of bacterias 2
Count of bacterias 2
Count of bacterias 2
Count of bacterias 2
Count of bacterias 3
Count of bacterias 4
Count of bacterias 4
Count of bacterias 4
Count of bacterias 4
Count of bacterias 4
Count of bacterias 4
Count of bacterias 4
Count of bacterias 4
Count of bacterias 4
Count of bacterias 5
Count of bacterias 7
Count of bacterias 8
Count of bacterias 8
Count of bacterias 8
Count of bacterias 8
Count of bacterias 8
Count of bacterias 8
Count of bacterias 8
Count of bacterias 8
Count of bacterias 9
Count of bacterias 12
Count of bacterias 15
Count of bacterias 16
Count of bacterias 16
Count of bacterias 16
Count of bacterias 16
Count of bacterias 16
Count of bacterias 16
```

Count of bacterias 17
Count of bacterias 21
Count of bacterias 27
Count of bacterias 31
Count of bacterias 32
Count of bacterias 32
Count of bacterias 32
Count of bacterias 32
Count of bacterias 32
Count of bacterias 33
Count of bacterias 38
Count of bacterias 48
Count of bacterias 58
Count of bacterias 63
Count of bacterias 64
Count of bacterias 64
Count of bacterias 64
Count of bacterias 64
Count of bacterias 65
Count of bacterias 71
Count of bacterias 86
Count of bacterias 106
Count of bacterias 121
Count of bacterias 127
Count of bacterias 128
Count of bacterias 128
Count of bacterias 128
Count of bacterias 129
Count of bacterias 136
Count of bacterias 157
Count of bacterias 192
Count of bacterias 227
Count of bacterias 248
Count of bacterias 255
Count of bacterias 256
Count of bacterias 256
Count of bacterias 257
Count of bacterias 265
Count of bacterias 293
Count of bacterias 349
Count of bacterias 419
Count of bacterias 475
Count of bacterias 503
Count of bacterias 511
Count of bacterias 512
Count of bacterias 513
Count of bacterias 522
Count of bacterias 558
Count of bacterias 642
Count of bacterias 768
Count of bacterias 894
Count of bacterias 978
Count of bacterias 1014
Stop program

Приложение

Содержимое файла main.go

```
package main

import (
    "fmt"
    "lab6/pkg/bacteria"
)

// Паттерн Flyweight. Разработать систему учета процессов размножения колонии бактерий.

func main() {
    bt := bacteria.NewBacteriaType(10)
    for {
        if bt.CountBacterias() < 1000 {
            bt.Tick()
        } else {
            fmt.Println("Stop program")
            break
        }
    }
}
```

Содержимое файла bacteria.go

```
package bacteria

import "fmt"

type IBacteria interface {
    tick()
    getCurrentLivetime() int
}

type BacteriaType struct {
    SplitPeriod int
    bacterias   []*Bacteria
}

func (bt BacteriaType) CountBacterias() int {
    fmt.Println("Count of bacterias", len(bt.bacterias))
    return len(bt.bacterias)
}

func NewBacteriaType(splitPeriod int) BacteriaType {
    bacterias := []*Bacteria{{currentLivetime: 1}}
    return BacteriaType{SplitPeriod: splitPeriod, bacterias: bacterias}
}

func (bt *BacteriaType) Tick() {
    newBacterias := 0
    for _, item := range bt.bacterias {
        item.tick()
        if item.getCurrentLivetime()%bt.SplitPeriod == 0 {
            newBacterias += 1
        }
    }
    for i := 0; i < newBacterias; i++ {
        bt.bacterias = append(bt.bacterias, &Bacteria{currentLivetime: 1})
    }
}

type Bacteria struct {
    currentLivetime int
}

func (b Bacteria) getCurrentLivetime() int {
    return b.currentLivetime
}

func (b *Bacteria) tick() {
    b.currentLivetime += 1
}
```