ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| acc. |  |  |  | Д.А.Кочин |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2 |
| «Функциональное тестирование методом черного ящика» |
| по курсу: Управление качеством программного обеспечения |
|  |
|  |

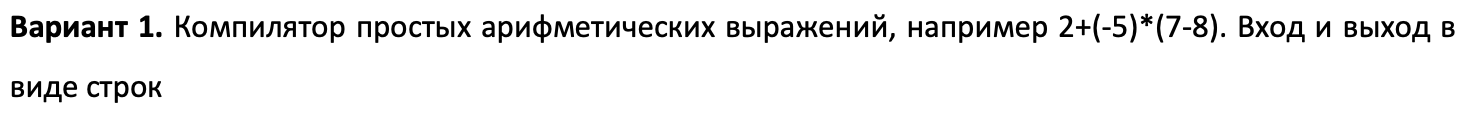
РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4831 |  | 08.10.2020 |  | К.А.Корнющенков |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2020

1. **Задание на лабораторную работу**

В рамках лабораторной работы необходимо произвести функциональное тестирование кода, оценить его покрытие и качество тестов



1. **Код программы**

func checkData(data:String) -> Bool {

for (index, value) in data.enumerated(){

if (index == 0 || index == data.count),(!value.isNumber){

return false

}

if value.isNumber || value == "+" || value == "\*" || value == "/" || value == "-"{

continue

} else{

return false

}

}

return true

}

func parseData(data: String, complition: @escaping([Int],[String]) -> Void) {

var number:[Int] = []

var action:[String] = []

var timeNumber = ""

for i in data{

if !i.isNumber {

action.append(String(i))

if timeNumber != ""{

number.append(Int(timeNumber)!)

timeNumber = ""

}

} else {

timeNumber += String(i)

}

}

if timeNumber != ""{

number.append(Int(timeNumber)!)

timeNumber = ""

}

complition(number,action)

}

func hardAction(number: [Int],action: [String],complition: @escaping([Int],[String]) -> Void){

var number = number

var action = action

var i = 0

let check = true

while check {

if action[i] == "\*"{

number[i] = number[i] \* number[i+1]

number.remove(at: i+1)

action.remove(at: i)

i = 0

}else if action[i] == "/"{

number[i] = number[i] / number[i+1]

number.remove(at: i+1)

action.remove(at: i)

i = 0

}

i += 1

if action.count == i{ break }

}

complition(number,action)

}

func simpleAction(number: [Int],action: [String],complition: @escaping([Int],[String]) -> Void) {

var number = number

var action = action

var i = 0

let check = true

while check {

if action[i] == "+"{

number[i] = number[i] + number[i+1]

number.remove(at: i+1)

action.remove(at: i)

i = 0

}else if action[i] == "-"{

number[i] = number[i] - number[i+1]

number.remove(at: i+1)

action.remove(at: i)

i = 0

}

i += 1

if action.isEmpty{ break }

}

complition(number,action)

}

let data = readLine()!

if checkData(data: data){

parseData(data: data) { number, action in

hardAction(number: number, action: action) { number, action in

simpleAction(number: number, action: action) { number, action in

print(number[0])

}

}

}

}else{

print("Ошибка при вводе данных")

}

1. **Ошибки в требованиях к спецификации**
2. В требовании номер один необходимо уточнить тип кофеварки и на чем основан принцип работы устройства (неполнота)
3. В данном требовании необходимо уточнить весь функционал электрокофеварка, чтобы полноценно описать и спроектировать устройство прибора (неполнота)
4. В данном требовании всё правильно
5. В данном требование необходимо какая кнопка должна включаться и какая крышка должна быть закрыть (неполнота)
6. Данное требование противоречит само себе. Непонятно куда необходимо наливать воду, если крышка закрыта (некорректность)
7. Необходимо уточнить с помощью чего надо делать управление электрокофеварки(экран на самом устройстве или какие-то стороннее приложение для отслеживания данных) (HTCPCP - протокол для управления, слежения и диагностики приборов для приготовления кофе.) (неполнота)
8. Требование номер семь невозможно проверить, необходимо согласовать дизайн с заказчиком в том смысле, что сперва будет утвержден дизайн и только после этого будет продолжена реализация. (непроверяемость)
9. **Спецификация на тесты**

Функция checkData(string)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя теста | Описание сценария | Входные данные | Выходные данные |
| Test\_checkData\_1 | Проверка входной строки на корректность нашего алфавита | 3+4\*5\*10 | true |
| Test\_checkData\_2 | Проверка входной строки на корректность нашего алфавита | +5+10 | false |
| Test\_checkData\_3 | Проверка входной строки на корректность нашего алфавита | 5+x+10 | False |

Функция parseData(string)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя теста | Описание сценария | Входные данные | Выходные данные |
| Test\_parseData\_1 | Преобразование входных данных в 2 динамических массива. 1 массив содержит в себе цифры, 2 простое арифметические операции | “4+5” | 1 параметр: [4,5]  2 параметр: [“+”] |
| Test\_parseData\_2 | Преобразование входных данных в 2 динамических массива. 1 массив содержит в себе цифры, 2 простое арифметические операции |  |  |

Функция hardAction([Int],[String])

|  |  |  |  |
| --- | --- | --- | --- |
| Имя теста | Описание сценария | Входные данные | Выходные данные |
| Test\_hardAction\_1 | Преобразование входные данных. Совершаем операцию умножение и деление | 1 параметр: [“+”,\*”]  2 параметр: [5,2,1] | 1 параметр: [“+”]  2 параметр: [5,2] |

Функция simpleAction([String],[Int])

|  |  |  |  |
| --- | --- | --- | --- |
| Имя теста | Описание сценария | Входные данные | Выходные данные |
| Test\_simpleAction\_1 | Преобразование входных данных. Совершаем операцию сложения и вычитания | 1 параметр: [“+”] 2 параметр: [5,2] | “7” |

1. **Вывод**

В ходе выполнения лабораторной работы были получены навыки написания тестов методом черного ящика.