

```
breaklines=true, xleftmargin=25pt, xrightmargin=25pt, aboveskip=10pt,  
belowskip=10pt, basicstyle=, backgroundcolor=lightlightgray, showstringspaces=false,  
frame=ltrb, language=Matlab, tabsize=2, numbers=left, numberstyle=,  
numbersep=8pt, morekeywords=*, factorial, sum, erlang, keywordstyle=shpurple,  
commentstyle=shgreen, stringstyle=shred
```

Санкт-Петербургский Национальный Исследовательский Университет  
Информационных Технологий, Механики и Оптики

Кафедра Систем Управления и Информатики

**Лабораторная работа №4**  
*Вариант №1*

Выполнили: Антипов В.А.  
Труфанова А.А.  
Черняев М.К.

Проверили: Глебова Е.С.  
Мусаев А.А.

Санкт-Петербург  
2018

# Создание модуля для работы с матричными операциями и использование его в вычислении параметров линейной системы автоматического управления

## Создание модуля для работы с матрицами

В основе модуля лежит класс *Matrix*, который базируется на таком простом типе данных, как *list*. Основная часть арифметических операций для данного класса была перегружена, для удобства представления операций в последующем. Было перегружено матричное умножение:

### Транспонирование матрицы

За транспонирование матрицы отвечает метод (`transpose()`), который является внутренним методом класса (`Matrix`):

```
def transpose(self): return Matrix(list(map(list,zip(*self._matrix))))
```

## 2)Задание «Написание скриптов. Циклы»

1. Изучить и предоставить краткий теоретический материал по использованию аргументов и циклов.

Оболочка `bash` поддерживает циклы `for`, `while`, а также условия `if` и `case`.

Аргумент цикла `for` может принимать последовательно заданные значения, список файлов, директории, значения из файла (с использованием команд прочтения), также это могут быть не только простые(однословные) значения, но также и целые фразы. Если на вход подаётся файл, то разделением выступают знаки табуляции, пробелы и переносы строк, чтобы выбрать например по знаку `:`, следует записать следующую команду `IFS=:`.

Цикл `while` принимает на сравнение значение переменной или выполнение команды. И в `for` и в `while` можно использовать команды `break` и `continue`. Есть возможность использовать вложенные циклы. Также после выполнения цикла переменная хранит последнее значение, и с ней можно работать дальше.

В условии `if` подается команда (выполнение команды `= 0` = переход по `then`; невыполнение `= 1` = переход по `else`) или сравнение. Сравнение осуществляется либо численными значениями, либо строками (по кодировке ASCII) условия сравнения строк заключаются в квадратные скобки, чисел в круглые.

2. Напишите скрипт на `bash`, который будет определять, в какую возрастную группу попадают пользователи.

При запуске скрипт должен вывести сообщение "Enter your name:" и ждать от пользователя ввода имени (команда `read`). Затем должно быть выведено имя (Например: «The name you entered is <Имя>. Hi <Имя>!»). После чего у пользователя должен быть выбор, продолжать работу или нет. Когда имя введено и сделан выбор, то скрипт должен написать "Enter your age:" и ждать ввода возраста. Когда возраст введен, скрипт пишет на экран «Имя>, your group is <группа> где <группа> определяется на основе возраста по следующим правилам:

младше либо равно 16: "child

от 17 до 25 (включительно): "youth

старше 25: "adult".

После этого скрипт опять выводит сообщение "Enter your name:" и всё начинается по новой. Если в какой-то момент работы скрипта будет введено пустое имя или возраст 0 (недопустимые символы, буквы итп.), то скрипт должен написать на экран "bye" и закончить свою работу.

Содержимое скрипта можно увидеть на рисунке 2.1

Был выбран следующий алгоритм работы:

- При запуске скрипт выводит строку с просьбой ввести своё имя (`echo "Enter your name:"`), после чего считывает введенное значение в переменную (`read name`), проверив строку на пустое значение. (`if [ "$name" = "" ]`, в данной конструкции сравнивается значение переменной `'name'` с пустым значением, после чего осуществляется разветвление в `if`. Если пользователь ввел пустое значение (условие выполняется)(`then`), то в файл (Рисунок 2.3)

записывается сообщение 'echo "Bye :c»>"age.txt" и просходит выход из скрипта(exit 0), иначе(else) (пользователь ввел верное значение) скрипт продолжает работу) и в файл (P) записывается сообщение с указанием имени и приветствием(echo "The name you entered is \$name. Hi \$name!»>"age.txt").

- Следующей строкой ползвателю задаётся вопрос о желание продолжить выполнение команды (echo "Do you want continue this program? (Yes/No)"), введенное значение (read answer) также проверяется на пустую строку (if [ -z \$answer ]), после чего обрабатывается в стуктре case: если ползователь согласен (введенное значение определяется, как "да")(Y|y|yes|Yes)), то в файл (Рисунок 2.3) записывается сообщение и скрипт продолжает выполнение (echo "You're amazing c:»>"age.txt"), если ответ ползователя оценивается как отрицательный (No; no; N; n), записывается прощание и производится выход из скрипта ( echo "Bye :c»>"age.txt"; exit 0) .
- Далее отображается просьба ввести возраст (echo "Enter your age:"), после чего считывается введенное значение (read age), в файл (Рисунок 2.3) записывается введенный возраст (echo "\$age»>"age.txt") и выполнется проверка на недопустимые символы, посредством сравнение переменной с 0 (if ((0<"\$age))), из-за чего скрипт воспринимает только целочисленную переменную, как выполнение условия, в противном случае (не целое число), скрипт заносив в файл (Рисунок 2.3) 'Bye :(' и завершает выполнение. Если условие выполнилось, и скрипт продолжает работу, выполняется выбор группы посредством сравнений введенного значения с разграничивающими значениями групп с помощью вложенный условий (if ((" \$age«=16)); then; group=child; elif ((17<=" \$age"&& " \$age«=25)); then; group=youth; else; group=adult; fi).
- Поскольку весь скрипт заключет в бесконечный цикл (while (true); do ... done), он не прекращает свою работу, пока ползоватень не откажется от продолжения работы скрипта, либо не введет недопустимое значение в качестве ответа.
- Запуск, выполнение и выход из скрипта рисунок 2.2

```
#!/bin/bash

while (true)
do

echo "Enter your name:"
read name
if [ "$name" == "" ]
then
    echo "Bye :c">>"age.txt"
    exit 0
else
echo "The name you entered is $name. Hi $name!" >> "age.txt"
fi

echo "Do you want continue this program? (Yes/No)"
read answer
if [ -z $answer ]
then
    echo "Bye :c">>"age.txt"
    exit 0
else
case "$answer" in
Y|y|yes|Yes) echo "You're amazing c:" >>"age.txt"
;;
N|n|no|No) echo "Bye :c">>"age.txt"
exit 0
;;
esac
fi

echo "Enter your age:"
read age
if ((0<"$age"))
then
    if ((" $age" <= 16))
    then
        group=child

    elif ((17<= "$age" && "$age" <= 25))
    then
        group=youth

    else
        group=adult
    fi
    echo "$name, your group is $group.">>"age.txt"
else
    echo "Bye :c">>"age.txt"
    exit 0
fi

done
```

Рисунок 2.1 скрипт для определения возрастной группы

```

keks@keks101:~/Documents/IT/Lab3$ touch age.txt
keks@keks101:~/Documents/IT/Lab3$ bash 2.sh
Enter your name:
Keks
Do you want continue this program? (Yes/No)
Yes
Enter your age:
3
Enter your name:
Shkiper
Do you want continue this program? (Yes/No)
yes
Enter your age:
21
Enter your name:
Kuki
Do you want continue this program? (Yes/No)
y
Enter your age:
83
Enter your name:
Jek
Do you want continue this program? (Yes/No)
Y
Enter your age:
*(
2.sh: line 35: ((: 0<*( : syntax error: operand expected (error token is "*(")
keks@keks101:~/Documents/IT/Lab3$ █

```

Рисунок 2.2 Запуск, ход выполнения и завершение выполнения скрипта на терминале

```

The name you entered is Keks. Hi Keks!
You're amazing c:
3
Keks, your group is child.
The name you entered is Shkiper. Hi Shkiper!
You're amazing c:
21
Shkiper, your group is youth.
The name you entered is Kuki. Hi Kuki!
You're amazing c:
83
Kuki, your group is adult.
The name you entered is Jek. Hi Jek!
You're amazing c:
*(
Bye :c
The name you entered is John. Hi John!
Bye :c

```

Рисунок 2.3 текстовый файл age.txt с результатом работы скрипта

## Вывод

В качестве вывода можно выделить некоторые особенности написания скриптов на `bush`: так например, вокруг знаков присвоения недопустимо ставить пробелы, а при написании условия в `if` напротив перед скобками следует ставить пробелы, также были изучены способности заносить в циклы и условия не только математические операции, но и выполнение/невыполнение различных команд. Конечно были изучены структура и синтаксис циклов, условий, функций и некоторых операций для написания скриптов, о чем изложено в работе выше.

Вкупе с приведенными ранее моментами можно сказать о том, что нами были изучены и освоены основы написания скриптов в оболочке `bush`. А также были решены небольшие практические задачи, подробнее о которых можно узнать из пунктов 1-3.



Спасибо за внимание