МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации

**

**ОТЧЁТ**

**по лабораторной работе № 2**

**«***Классификация языков программирования***»**

**по дисциплине: «***Программирование***»**

Выполнил:Проверил:

Студент гр. «АБ-121», «АВТФ» *доцент кафедры ЗИ*

*Втюрин Александр Романович Ершов И. А.*

«\_\_\_» \_\_\_\_\_\_ 2022г«\_\_\_» \_\_\_\_\_\_ 2022 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (подпись)

Новосибирск 2022

**Цели и задачи работы:** изучение функций ввода-вывода данных, программирования вычисления значения выражения.

**Задание к работе:**

Реализовать линейный вычислительный процесс. Самостоятельно решить задачу в соответствии с индивидуальным вариантом. Реализовать представленные задачи на языках программирования C++, Python, Haskell (только одно задание), C#, Go, Ruby (только одно задание), Java.

Реализовать линейный вычислительный процесс любого задания на языке программирования Assembler.

Представленные задачи можно реализовать на каждом языке в одной программе с последовательным выполнением.

***Задания на разных языках программирования***

Задание 1: Даны длины ребер a, b, c прямоугольного параллелепипеда. Найти его объем V = a·b·c и площадь поверхности S = 2·(a·b + b·c + a·c).

Задание 2: Учитывая массив nums целых чисел, вернуть, сколько из них содержат нечетное количество цифр.

Задание 3: Для каждого трёхзначного числа определить является ли оно палиндромом (Читается одинаково с любой стороны, например 101, 222 и т.д.).

С++

#include <iostream>

#include <vector>

using namespace std;

void Task1()

{

    int a, b, c;

    cout << "Enter the length:";

        cin >> a;

    cout << "\nEnter the width:";

        cin >> b;

    cout << "\nEnter the height:";

        cin >> c;

    int V, S;

    S = 2 \* (a \* b + b \* c + a \* c);

    cout << "S = " << S << endl;

    V = a \* b \* c;

    cout << "V = " << V << endl;

}

void Task2()

{

    int n, s;        //n - кол-во входящих чисел, s - кол-во подходящих чисел

    s = 0;

    cout << "Enter the number of items in the selection: "; cin >> n;

    vector <int> nums(n, 0);       //инициализация массива

    for (int i = 0; i < n; i++) {

        nums[i] = rand() % 9999;         //рандом значений

        cout << nums[i] << " ";             //вывод заполненного массива

    }

    cout << "\n";

    for (int i = 0; i < n; i++) {         //перебор значений

        int o = nums[i];      //значение из вектора

        int k = 0;                  //кол-во цифр

        if (o != 0) {                //проверка на 0

            while (o != 0) {                      //подсчёт цифр

                o = o / 10;

                k++;

            }

            if (k % 2 == 1)

                s++;

        }

    }

    cout << s;

}

void Task3()

{

    while (true)

    {

        int num;

        cout << "Enter The number:";

        cin >> num;

        if (num >= 100 && num < 1000) { // Проверка, является ли число трёхзначным или нет

            if (num / 100 == num % 10)// Проверка, является ли число палиндромом

                cout << "Yes" << endl;

            else cout << "No" << endl;

        }

        else

            cout << "No" << endl;

        num = 0;

        bool res;

        cout << "Continue or not(1/0):";

        cin >> res;

        if (res == 0)

            break;

    }

}

int main()

{

    int task;

    cout << "Select a task:";

    cin >> task;

    if (task == 1) {

        cout << "Task 1:" << endl;

        Task1();

    }

    if (task == 2) {

        cout << "\nTask 2:" << endl;

        Task2();

    }

    if (task == 3) {

        cout << "\nTask 3:" << endl;

        Task3();

    }

    else

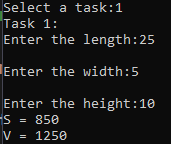
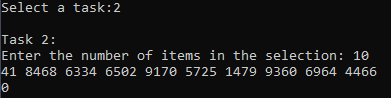
    {

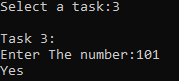
        cout << "Invalid value";

    }

    return 0;

}





С#

using System;

using System.Collections.Generic;

public static class Лаба

{

    /\*

    ЗАДАНИЕ 1.

    Даны длины ребер a, b, c прямоугольного параллелепипеда.

    Найти его объем V = a·b·c и площадь поверхности S = 2·(a·b + b·c + a·c).

    ЗАДАНИЕ 2

    Учитывая массив nums целых чисел,

    вернуть, сколько из них содержат нечетное количество цифр.

    ЗАДАНИЕ 3

    Для каждого трёхзначного числа определить является ли оно палиндромом

    (Читается одинаково с любой стороны, например 101, 222 и т.д.).

    \*/

    public static void Task1()

    {

        int a;

        int b;

        int c;

        Console.Write("Enter the length:");

            a = Console.ReadLine();

        Console.Write("\nEnter the width:");

            b = Console.ReadLine();

        Console.Write("\nEnter the height:");

            c = Console.ReadLine();

        int V;

        int S;

        S = 2 \* (a \* b + b \* c + a \* c);

        Console.Write("S = ");

        Console.Write(S);

        Console.Write("\n");

        V = a \* b \* c;

        Console.Write("V = ");

        Console.Write(V);

        Console.Write("\n");

    }

    public static void Task2()

    {

        int n; //n - кол-во входящих чисел, s - кол-во подходящих чисел

        int s;

        s = 0;

        Console.Write("Enter the number of items in the selection: ");

        n = Console.ReadLine();

        List<int> nums = new List<int>(n); //инициализация массива

        for (int i = 1; i <= n; i++)

        {

            nums[i] = RandomNumbers.NextNumber() % 9999; //рандом значений

            Console.Write(nums[i]);

            Console.Write(" ");

        }

        Console.Write("\n");

        for (int i = 1; i <= n; i++)

        { //перебор значений

            int o = nums[i]; //значение из вектора

            int k = 0; //кол-во цифр

            if (o != 0)

            { //проверка на 0

                while (o != 0)

                { //подсчёт цифр

                    o = o / 10;

                    k++;

                }

                if (k % 2 == 1)

                {

                    s++;

                }

            }

        }

        Console.Write(s);

    }

    public static void Task3()

    {

        while (true)

        {

            int num;

            Console.Write("Enter The number:");

            num = Console.ReadLine();

            if (num >= 100 && num < 1000)

            { // Провека, является ли число трёхзначным или нет

                if (num / 100 == num % 10) // Проверка, является ли число полиндромом

                {

                    Console.Write("Yes");

                    Console.Write("\n");

                }

                else

                {

                    Console.Write("No");

                    Console.Write("\n");

                }

            }

            else

            {

                Console.Write("No");

                Console.Write("\n");

            }

            num = 0;

            bool res;

            Console.Write("Continue or not(1/0):");

            res = Console.ReadLine();

            if (res == false)

            {

                break;

            }

        }

    }

    static int Main()

    {

        int task;

        Console.Write("Select a task:");

        task = Console.ReadLine();

        if (task == 1)

        {

            Console.Write("Task 1:");

            Console.Write("\n");

            Task1();

        }

        if (task == 2)

        {

            Console.Write("\nTask 2:");

            Console.Write("\n");

            Task2();

        }

        if (task == 3)

        {

            Console.Write("\nTask 3:");

            Console.Write("\n");

            Task3();

        }

        else

        {

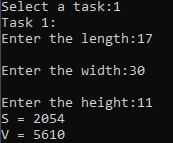
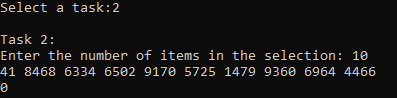
            Console.Write("Invalid value");

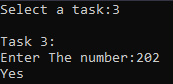
        }

        return 0;

    }

}





Haskell

s (a, b, c) = 2\*(a\*b+b\*c+a\*c)

v (a,b,c)=a\*c\*b

main = do

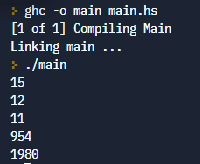
a <- readLn

b <- readLn

c <- readLn

print $ s (a,b,c)

print $ v (a,b,c)



Java

import java.util.Scanner;

public class Лаба

{

    public static void Task1()

    {

        int a;

        int b;

        int c;

        Scanner in = new Scanner(System.in);

        System.out.print("Enter the length:");

            a = in.nextInt();

        System.out.print("\nEnter the width:");

            b = in.nextInt();

        System.out.print("\nEnter the height:");

            c = in.nextInt();

        int V;

        int S;

        S = 2 \* (a \* b + b \* c + a \* c);

        System.out.print("S = ");

        System.out.print(S);

        System.out.print("\n");

        V = a \* b \* c;

        System.out.print("V = ");

        System.out.print(V);

        System.out.print("\n");

        in.close();

    }

    public static void Task2()

    {

        Scanner in = new Scanner(System.in);

        int n; //n - кол-во входящих чисел, s - кол-во подходящих чисел

        int s;

        s = 0;

        System.out.print("Enter the number of items in the selection: ");

        n = in.nextInt();;

        ArrayList<Integer> nums = new ArrayList<Integer>(n); //инициализация массива

        for (int i = 1; i <= n; i++)

        {

            nums.set(i, tangible.RandomNumbers.nextNumber() % 9999); //рандом значений

            System.out.print(nums.get(i));

            System.out.print(" ");

        }

        System.out.print("\n");

        for (int i = 1; i <= n; i++)

        { //перебор значений

            int o = nums.get(i); //значение из вектора

            int k = 0; //кол-во цифр

            if (o != 0)

            { //проверка на 0

                while (o != 0)

                { //подсчёт цифр

                    o = o / 10;

                    k++;

                }

                if (k % 2 == 1)

                {

                    s++;

                }

            }

        }

        System.out.print(s);

        in.close();

    }

    public static void Task3()

    {

        Scanner in = new Scanner(System.in);

        while (true)

        {

            int num;

            System.out.print("Enter The number:");

            num = in.nextInt();;

            if (num >= 100 && num < 1000)

            { // Провека, является ли число трёхзначным или нет

                if (num / 100 == num % 10) // Проверка, является ли число полиндромом

                {

                    System.out.print("Yes");

                    System.out.print("\n");

                }

                else

                {

                    System.out.print("No");

                    System.out.print("\n");

                }

            }

            else

            {

                System.out.print("No");

                System.out.print("\n");

            }

            num = 0;

            boolean res;

            System.out.print("Continue or not(1/0):");

            res = in.nextBoolean();

            if (res == false)

            {

                break;

            }

        }

        in.close();

    }

    public static int Main()

    {

        Scanner in = new Scanner(System.in);

        int task;

        System.out.print("Select a task:");

        task = in.nextInt();

        if (task == 1)

        {

            System.out.print("Task 1:");

            System.out.print("\n");

            Task1();

        }

        if (task == 2)

        {

            System.out.print("\nTask 2:");

            System.out.print("\n");

            Task2();

        }

        if (task == 3)

        {

            System.out.print("\nTask 3:");

            System.out.print("\n");

            Task3();

        }

        else

        {

            System.out.print("Invalid value");

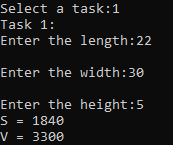
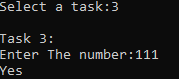
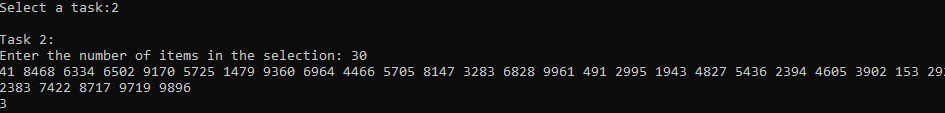
        }

        in.close();

        return 0;

    }

}

Python

import math

import random

class Globals:

@staticmethod

def Task1():

a = None

b = None

c = None

print("Enter the length:", end = '')

a = input

print("\nEnter the width:", end = '')

b = input

print("\nEnter the height:", end = '')

c = input

V = None

S = None

S = 2 \* (a \* b + b \* c + a \* c)

print("S = ", end = '')

print(S, end = '')

print("\n", end = '')

V = a \* b \* c

print("V = ", end = '')

print(V, end = '')

print("\n", end = '')

@staticmethod

def Task2():

n = None #n - кол-во входящих чисел, s - кол-во подходящих чисел

s = None

s = 0

print("Enter the number of items in the selection: ", end = '')

n=input

nums = [0 for \_ in range(n)] #инициализация массива

for i in range(1, n + 1):

nums[i] = random.random() #рандом значений

print(nums[i], end = '')

print(" ", end = '')

print("\n", end = '')

for i in range(1, n + 1):

o = nums[i] #значение из вектора

k = 0 #кол-во цифр

if o != 0:

while o != 0:

o = math.trunc(o / float(10))

k += 1

if math.fmod(k, 2) == 1:

s += 1

print(s, end = '')

@staticmethod

def Task3():

while True:

num = None

print("Enter The number:", end = '')

num = input

if num >= 100 and num < 1000:

if math.trunc(num / float(100)) == math.fmod(num, 10): # Проверка, является ли число полиндромом

print("Yes", end = '')

print("\n", end = '')

else:

print("No", end = '')

print("\n", end = '')

else:

print("No", end = '')

print("\n", end = '')

num = 0

res = None

print("Continue or not(1/0):", end = '')

res = input

if res == False:

break

def main():

task = None

print("Select a task:", end = '')

task = input

if task == 1:

print("Task 1:", end = '')

print("\n", end = '')

Globals.Task1()

if task == 2:

print("\nTask 2:", end = '')

print("\n", end = '')

Globals.Task2()

if task == 3:

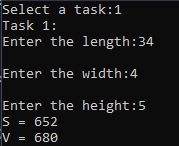
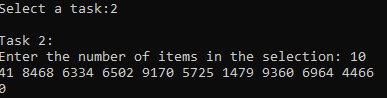
print("\nTask 3:", end = '')

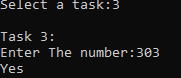
print("\n", end = '')

Globals.Task3()

else:

print("Invalid value", end = '')





Ruby

print 'a='

a = gets.chomp.to\_i

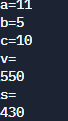
print 'b='

b = gets.chomp.to\_i

print 'c='

c = gets.chomp.to\_i

puts 'v=', a \* b \* c, 's=',  2 \* (a \* b + b \* c + a \* c)

****

Go

package main // пакет main

import "fmt" // библиотека для вывода

import "math" // библиотека мат.функций

func Num3(riad[100] int, dlin int) int{ // функция для 3 номера

var h int = 0 // объявление и инициализация переменной

var i int

for i = 1; i < dlin; i++{ // шагаем по массиву

if riad[i] == riad[i - 1]{ // проверка условия

h += 1

}

}

return h; // возврат значения функцией

}

func Flowers(potrfl[100] int, kolfl int, lei int) int{ // функция для второго задания

var b int = lei // кол-во воды в лейке

var steps int = 1 // шаги

var i int

for i = 0; i < (kolfl - 1); i++{ // шагаем по массиву

if potrfl[i + 1] > (b - potrfl[i]){ // если не хватает на след

steps += (i + 1) + i + 2 // треб шагов

b = lei

}else{ // хватает на след

b -= potrfl[i]

steps += 1

}

}

return steps;

}

func Area(x1 float64, y1 float64, x2 float64, y2 float64) float64{ // функция площади

var a float64 = math.Abs(x1 - x2) \* math.Abs(y1 - y2) // площадь

return a

}

func Perim(x1 float64, y1 float64, x2 float64, y2 float64) float64{ // функция периметра

var a float64 = (math.Abs(x1 - x2) + math.Abs(y1 - y2)) \* 2 // периметр

return a

}

func main() {

var x1, y1, x2, y2 float64

fmt.Println("Number 1") // вывод сообщения в консоль

fmt.Println("Enter the coordinates of the first point")

fmt.Scanf("%float64", &x1) // ввод значения переменной x1 типа float64

fmt.Scanf("%float64", &y1)

fmt.Println("Enter the coordinates of the second point")

fmt.Scanf("%float64", &x2)

fmt.Scanf("%float64", &y2)

fmt.Println("Area = ", Area(x1, y1, x2, y2)) // вывод площади

fmt.Println("Perimetr = ", Perim(x1, y1, x2, y2)) // вывод периметра

fmt.Println("Number 2")

var potrfl[100] int

var kolfl, lei int

fmt.Println("Enter the number of flowers")

fmt.Scanf("%d\n", &kolfl)

fmt.Println("Consistently enter the need of each flower")

var i, a int

for i = 0; i < kolfl; i++{ // цикл заполнения массива потребностей цветков

fmt.Scanf("%d\n", &a)

potrfl[i] = a

}

fmt.Println("Enter the volume of the watering can")

fmt.Scanf("%d\n", &lei)

fmt.Println("Needed", Flowers(potrfl, kolfl, lei), "steps") // вывод необходимого кол-ва шагов

fmt.Println("Number 3")

var riad[100] int

var dlin int

fmt.Println("Enter the number of numbers")

fmt.Scanf("%d\n", &dlin)

fmt.Println("Enter each number of row") // заполнение ряда числами

for i = 0; i < dlin; i++{

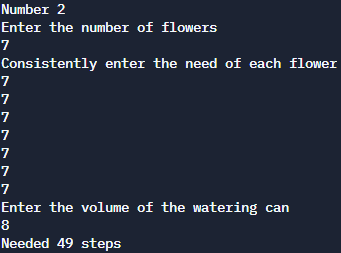
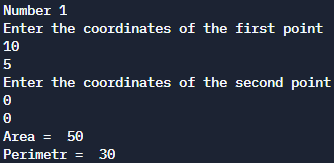
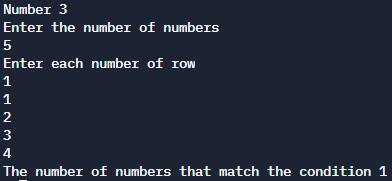
fmt.Scanf("%d\n", &a)

riad[i] = a

}

fmt.Println("The number of numbers that match the condition", Num3(riad, dlin)) // вывод ответа на 3 задание

}



Asembler

main:

        push    rbp

        mov     rbp, rsp

        sub     rsp, 32

        mov     DWORD PTR [rbp-4], 25

        mov     DWORD PTR [rbp-8], 15

        mov     DWORD PTR [rbp-12], 10

        mov     eax, DWORD PTR [rbp-4]

        imul    eax, DWORD PTR [rbp-8]

        mov     edx, eax

        mov     eax, DWORD PTR [rbp-4]

        imul    eax, DWORD PTR [rbp-12]

        add     edx, eax

        mov     eax, DWORD PTR [rbp-8]

        imul    eax, DWORD PTR [rbp-12]

        add     eax, edx

        add     eax, eax

        mov     DWORD PTR [rbp-16], eax

        mov     eax, DWORD PTR [rbp-4]

        imul    eax, DWORD PTR [rbp-8]

        mov     edx, DWORD PTR [rbp-12]

        imul    eax, edx

        mov     DWORD PTR [rbp-20], eax

        mov     eax, DWORD PTR [rbp-16]

        cdqe

        mov     rdi, rax

        mov     eax, 0

        call    printf

        mov     eax, DWORD PTR [rbp-20]

        cdqe

        mov     rdi, rax

        mov     eax, 0

        call    printf

        mov     eax, 0

        leave

        ret



**Дополнительные задания**

**Список странных ЯП**

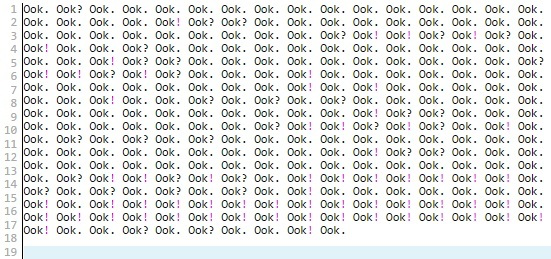
## Ook!



Целое семейство эзотерических языков выглядит так, будто основывается на принципе английского философа Уильяма Оккама: «Не следует множить сущее без необходимости». Они отрицают какой-либо дополнительный синтаксис, а их разработчики часто гонятся за уменьшением размера компилятора.

Несмотря на внешнюю примитивность, такие языки могут иметь бесконечный набор ячеек и тьюринговскую полноту, и, следовательно, по потенциальным возможностям не уступать «настоящим», подобным C, Pascal или Java. Например, для поклонников Терри Пратчетта существует «Ook!». Этот необычный язык программирования имитирует речь одного из героев «Плоского мира» — орангутанга-библиотекаря из Незримого Университета.

Пример кода для вывода сообщения «Hello, world!»:

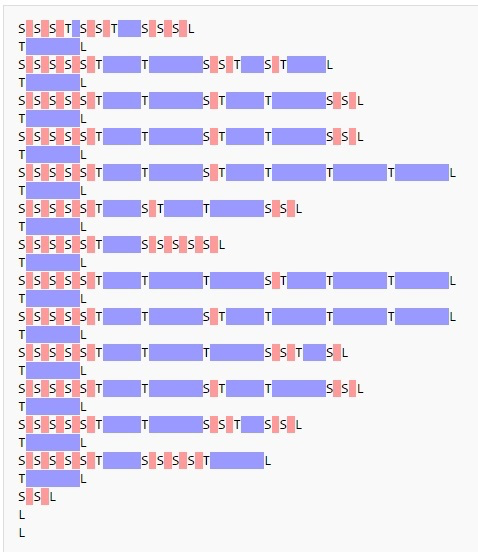


## Whitespase



Whitespace (англ. пробел) — эзотерический язык программирования, созданный Э. Брэди и К. Моррисом. Существенным его отличием является то, что для управляющих конструкций используются только непечатаемые символы, а именно: пробел, перевод строки и табуляция. Интересным следствием этого факта является то, что текст программы на языке Whitespace можно «скрыть» внутри исходных кодов другой программы. Кстати язык был выпущен 1 апреля 2003 года, и многие восприняли его как шутку на День смеха.

Пример кода для вывода сообщения «Hello, world!» (с подсветкой спецсимволов):



## Chef



Chef — эзотерический язык программирования, разработанный Дэвидом Морган-Маром, программы на котором сходны с кулинарными рецептами. Каждая программа в языке состоит из названия, списка переменных и их значений, списка инструкций. Переменные могут быть названы только названиями основных продуктов питания. Стек, в которые помещаются значения переменных, называется англ. mixing bowl («чаша для смешивания»), а операции для манипуляции с переменными — mix («смешать»), stir («взболтать») и так далее. Язык идеально подходит для домохозяек и для Джеймса Бонда.

Пример кода для вывода сообщения «Hello, world!»:

Hello World Souffle.

*Ingredients.*

*72 g haricot beans*

*101 eggs*

*108 g lard*

*111 cups oil*

*32 zucchinis*

*119 ml water*

*114 g red salmon*

*100 g dijon mustard*

*33 potatoes*

*Method.*

*Put potatoes into the mixing bowl.*

*Put dijon mustard into the mixing bowl.*

*Put lard into the mixing bowl.*

*Put red salmon into the mixing bowl.*

*Put oil into the mixing bowl.*

*Put water into the mixing bowl.*

*Put zucchinis into the mixing bowl.*

*Put oil into the mixing bowl.*

*Put lard into the mixing bowl.*

*Put lard into the mixing bowl.*

*Put eggs into the mixing bowl.*

*Put haricot beans into the mixing bowl.*

*Liquefy contents of the mixing bowl.*

*Pour contents of the mixing bowl into the baking dish.*

*Serves 1.*

## Shakespeare



Если же вы не готовите, ходите есть в рестораны и предпочитаете духовную пищу, найдется вариант и для вас. Shakespeare — эзотерический язык программирования, разработанный Джоном Аслаудом и Карлом Хассельстромом. Язык Shakespeare призван замаскировать исходный код программы под пьесы Уильяма Шекспира.

Список персонажей в начале программы служит для объявления количества стеков, носящих имена, например, «Ромео» и «Джульетта». Эти герои общаются друг с другом — тем самым совершая операции ввода-вывода, могут задавать вопросы (аналог условного оператора). Части исходного кода на Shakespeare называются Актами (Act), которые разбиваются на Сцены (Scene). Каждые Акт и Сцена пронумерованы римскими цифрами и служат метками для GOTO.

Прежде чем персонажи смогут участвовать, они должны сначала выйти на сцену. Чтобы поместить персонажа на сцену, его необходимо вызвать командой Enter. Одновременно на сцене могут находиться только два персонажа. Иначе не ясно, с кем ведется общение. Чтобы персонаж покинул сцену, необходимо обратиться к нему командой Exit. Если акт заканчивается, или необходимо просто прогнать нескольких персонажей, то необходимо использовать команду Exeunt. В результате получается и пьеса, и программа.

Пример кода для вывода сообщения «Hello, world!»:

*Romeo, a young man with a remarkable patience.*

*Juliet, a likewise young woman of remarkable grace.*

*Ophelia, a remarkable woman much in dispute with Hamlet.*

*Prince Hamlet, the flatterer of Andersen Insulting A/S.*

*Act I: Hamlet's insults and flattery.*

*Scene I: The insulting of Romeo.*

*[Enter Hamlet and Romeo]*

*Hamlet:*

*You lying stupid fatherless big smelly half-witted coward! You are as*

*stupid as the difference between a handsome rich brave hero and thyself!*

*Speak your mind!*

*You are as brave as the sum of your fat little stuffed misused dusty*

*old rotten codpiece and a beautiful fair warm peaceful sunny summer's*

*day. You are as healthy as the difference between the sum of the*

*sweetest reddest rose and my father and yourself! Speak your mind!*

*You are as cowardly as the sum of yourself and the difference*

*between a big mighty proud kingdom and a horse. Speak your mind.*

*Speak your mind!*

*[Exit Romeo]*

*Scene II: The praising of Juliet.*

*[Enter Juliet]*

*Hamlet:*

*Thou art as sweet as the sum of the sum of Romeo and his horse and his*

*black cat! Speak thy mind!*

*[Exit Juliet]*

*Scene III: The praising of Ophelia.*

*[Enter Ophelia]*

*Hamlet:*

*Thou art as lovely as the product of a large rural town and my amazing*

*bottomless embroidered purse. Speak thy mind!*

*Thou art as loving as the product of the bluest clearest sweetest sky*

*and the sum of a squirrel and a white horse. Thou art as beautiful as*

*the difference between Juliet and thyself. Speak thy mind!*

*[Exeunt Ophelia and Hamlet]*

*Act II: Behind Hamlet's back.*

*Scene I: Romeo and Juliet's conversation.*

*[Enter Romeo and Juliet]*

*Romeo:*

*Speak your mind. You are as worried as the sum of yourself and the*

*difference between my small smooth hamster and my nose. Speak your*

*mind!*

*Juliet:*

*Speak YOUR mind! You are as bad as Hamlet! You are as small as the*

*difference between the square of the difference between my little pony*

*and your big hairy hound and the cube of your sorry little*

*codpiece. Speak your mind!*

*[Exit Romeo]*

*Scene II: Juliet and Ophelia's conversation.*

*[Enter Ophelia]*

*Juliet:*

*Thou art as good as the quotient between Romeo and the sum of a small*

*furry animal and a leech. Speak your mind!*

*Ophelia:*

*Thou art as disgusting as the quotient between Romeo and twice the*

*difference between a mistletoe and an oozing infected blister! Speak*

*your mind!*

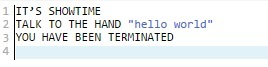
*[Exeunt]*

## ArnoldC



Любителей кино порадует язык программирования ArnoldC, составленный из цитат героев Арнольда Шварцнеггера. Не будем углубляться в подробности.

Пример кода для вывода сообщения «Hello, world!»:



MarioLANG

Пришло время вспомнить далёкое детство… Нет, речь не про **BASIC**, хотя сегодня он и считается практически реликвией. У всех или почти всех ведь была «Денди»? И принцессу искали, и монетки собирали, и по трубам лазили… Встречайте новое амплуа итальянского водопроводчика в мире программирования.

Структура

Исходный код на этом языке действительно напоминает мир игр серии Super Mario. В качестве условных операторов используются символы, выступающие строительным материалом для локаций, а также указания для усатого сантехника. Всё, что не распознаётся как команда, является комментарием.

Примеры кода:

++++: > > +:+:+:+:+:+:+:::::

====+ >^=== """==================

+:-):(:^= = !

========= = #

= ! .+.,:-<

=### ======"

В результате выполнения кода выводится последовательность чисел: 4 6 0 5 6 7 8 9 10 11 12 12 12 12 12 11. После этого пользователь должен ввести букву, и программа выведет её и следующую в алфавите.

++++:

====

Эта программа выводит цифру 4. Двоеточие в конце означает, что Марио останавливается и выполняет вывод.

Вудс и Лайон хотели создать язык, на котором было бы очень сложно или вовсе невозможно программировать, и добились своего. Даже самые простые операции в INTERCAL заставят кодера попотеть. Язык использует метод вывода строк «лента Тьюринга», в котором массив чисел обрабатывается слева направо по одному символу за раз. Некоторые команды блокируют выполнение последующих команд или изменения переменных в коде. Визитная карточка INTERCAL — это модификатор PLEASE, который не несет никакой смысловой нагрузки, но должен встречаться в коде четыре или пять раз. Если вы используете PLEASE меньше трех раз, программа выдаст ошибку «Программист недостаточно вежлив», если больше пяти — «Программист слишком вежлив».

Пример кода для Hello,World на языке INTERCAL:

**DO** ,1 <- #13

PLEASE **DO** ,1 SUB #1 <- #238

**DO** ,1 SUB #2 <- #108

**DO** ,1 SUB #3 <- #112

**DO** ,1 SUB #4 <- #0

**DO** ,1 SUB #5 <- #64

**DO** ,1 SUB #6 <- #194

PLEASE **DO** ,1 SUB #7 <- #48

**DO** ,1 SUB #8 <- #26

**DO** ,1 SUB #9 <- #244

PLEASE **DO** ,1 SUB #10 <- #168

**DO** ,1 SUB #11 <- #24

**DO** ,1 SUB #12 <- #16

**DO** ,1 SUB #13 <- #162

PLEASE READ OUT ,1

PLEASE GIVE UP

INTERCAL запустил традицию создания странных и экспериментальных языков, которую подхватили другие программисты. Например, в 1998 году превзойти INTERCAL по невыносимости решил Бен Олмстад. Он изобрел язык Malbolge, который назвал в честь восьмого круга ада из «Божественной комедии» Данте. Пожалуй, само название говорит о сложности языка лучше всего.

**Befunge**

Этому языку уже почти 30 лет, и он был придуман как самый сложный для компиляции язык программирования. Работает он так:

Есть таблица с командами для процессора.

Каждая команда обозначается каким-то символом из кодовой таблицы [ASCII](http://thecode.media/symbols/?utm_source=zen&utm_medium=social&utm_campaign=post).

Эти команды можно извлекать из таблицы с помощью модификаторов p и g.

Кроме таблицы в языке есть много односимвольных команд, каждая из которых делает какую-то одну простейшую операцию.

При компиляции программа собирается из содержимого таблицы и родных команд.

Даже самое простое описание этого языка уже выглядит очень сложно, не говоря уже о программах. Вот реализация алгоритма Фибоначчи, которая выдаёт первые 14 чисел последовательности:

|  |
| --- |
| 62\*1+v>01p001>+v>\:02p\:02gv |
|  | 0 ^ < |
|  | . :p |
|  | » .1 |
|  | v 0," "<0 |
|  | " >1g12-+:| |
|  | , @ |
|  | >^ |

## Malbolge

Когда Бен Олмстед в 1998 году узнал, что Befunge считается самым сложным языком, то попросил подержать его пиво, и пока кто-то держал его пиво, создал Malbolge. Название языка совпадает с названием восьмого круга ада у Данте, что должно намекать на его адовую сложность.

Особенность этого языка в том, что он работает в троичной системе счисления и имеет всего три регистра для указателей и управления данными. Ещё у этого языка нет стандарта синтаксиса, команды могут идти в произвольном порядке. А чтобы было ещё интереснее, после загрузки вся свободная память интерпретатора заполняется с помощью команды crazy: она сравнивает в троичной системе 2 числа по своим правилам и результат отправляется в память.

И напоследок: после того, как выполняется очередная команда, она шифруется с помощью таблицы перевода.

Исходные символы:

!»$%>& '()\*+, -./0123456789:;<=>? @ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^\_`abcdefghijklmnopqrstuvwxyz{|}~

Закодированные символы:

5z]& gqtyfr$(we4{WP)H-Zn, [%\3dL+Q;>U! pJS72FhOA1CB6v^=I\_0/8|jsb9m<.TVac`uY\*MK’X~xDl}REokN: #? G«i@

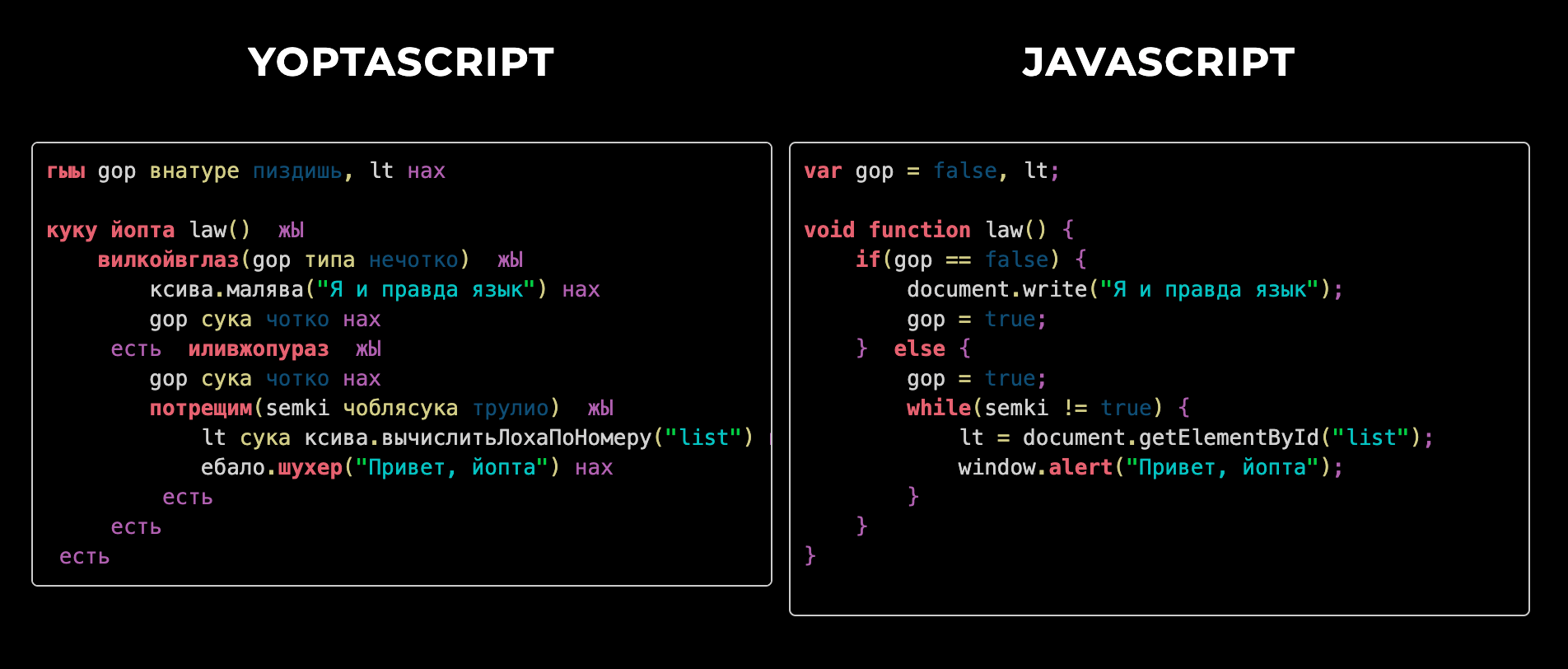
Язык получился настолько сложным, что даже его создатель не смог написать на нём программу. Первая программа на Malbolge появилась два года спустя, когда энтузиасты с помощью Lisp перебирали варианты кода, чтобы вывести классическое «Hello, world»:

(‘&%:9]!~}|z2Vxwv-,POqponl$Hjig%eB@@>}=<M:9wv6WsU2T|nm-,jcL(I&%$#"`CB]V?Tx<uVtT`Rpo3NlF.Jh++FdbCBA@?]!~|4XzyTT43Qsqq(Lnmkj«Fhg${z@>

На сегодняшний день в мире есть всего 3 программы, написанные на этом языке, и ни одна из них не была создана человеком.

В отличии от прошлых примеров YoptaScript написан нашими ребятами, студентами из петербургского университета. YoptaScript – это язык для настоящих гопников, которые хотят из дворов перебраться в элиту программистов.

Структуру языка можно быстро освоить и уже через пару часов писать JavaScript подобный код с использованием синтаксиса YoptaScript.



Кстати, назвать YoptaScript отдельным языком нельзя. Это скриптовый язык, что транслируется специальной библиотекой в JavaScript код. Общий принцип языка построен на изменённых частях языка Javascript. К примеру, вместо слова «void» используется слово «куку», а вместо «function» – «йопта».

У ребят получилось сделать забавный язык. Попробовать его в действии и написать свои вполне нормальные программы можно [прямиком на их сайте](https://yopta.space/).

## Piet

Напоследок — красивый язык программирования. Вот как выглядит программа «Hello, world»:



Язык работает так: компилятор оценивает разницу между соседними цветами, чтобы понять, что ему делать дальше. При этом он ещё перебирает оттенки цветов по кругу — так программа всегда знает, к какому цвету нужно перейти дальше.

Кроме оттенков программа сравнивает и яркость цветов: светло-синий и тёмно-синий — это два разных цвета, каждый из которых обрабатывается по-своему.

Ещё немного программ на Piet:



Выводит восклицательный знак



Тоже «Hello, world», который выглядит как грибочек из Марио

**Хронология создания ЯП:**

****

**Вывод**

В ходе проделанной работы был реализован линейный вычислительный процесс. Решены задачи в соответствии с индивидуальным вариантом на разных языках программирования, а именно: C++, Python, C#, Go, Haskell, Java, Ruby, Assembler. Также были выполнены дополнительные задания к лабораторной работе и изучена теоретическая часть.