

## Лабораторная работа № 4-5

### «Оценка сложности и оформление алгоритмов разветвляющейся структуры (ветвление)»

**Цель работы:** ознакомиться с оператором ветвления и научиться составлять программы с его использованием.

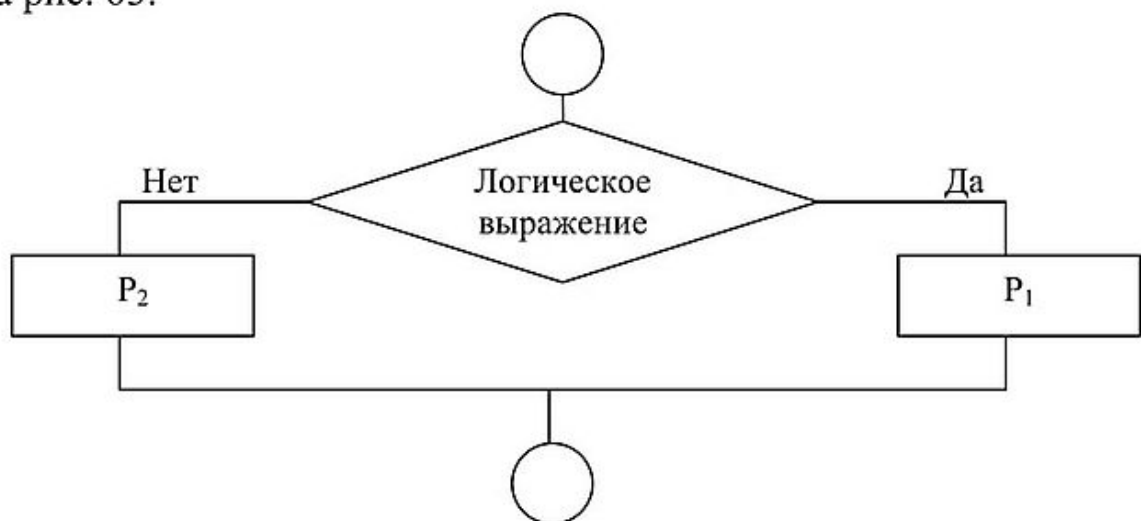
#### Теория

Алгоритм называется разветвляющимся, если последовательность выполнения шагов алгоритма изменяется в зависимости от выполнения некоторых условий. Условие — это логическое выражение, которое может принимать одно из двух значений: «ДА» — если условие верно (истинно), «НЕТ» — если условие неверно (ложно).

Простой условный оператор

Общий вид в алгоритме конструкции простого условного оператора представлен на рисунке 2.

ставлен на рис. 63.



Синтаксис простого условного оператора следующий:

```
if (Логическое выражение)
    p, ;
else
    p2;
```

где if (если), else (иначе) — зарезервированные слова, а P1 P2 — операторы. Простой условный оператор работает по следующему алгоритму.

Сначала вычисляется логическое выражение. Если результат есть true (истина), то выполняется оператор P1 а оператор P2 пропускается. Если результат есть false (ложь), то выполняется оператор P2, а оператор P1 пропускается.

Например, в листинге в зависимости от результата сравнения двух переменных

ВЫВОДИТСЯ ТОТ ИЛИ ИНОЙ ОТВЕТ.

Листинг

```
{
    private void Form1_Load(object sender, EventArgs e)
    {
        int a=1; int b=5; if (a<b) p=""></b><>
        MessageBox.Show("Тест 1 ветви ", "Заголовок окна", MessageBoxButtons.OK,
        else
        MessageBox.Show("Тест 2 ветви ", "Заголовок окна", MessageBoxButtons.OK,
        Application.Exit();
    }
}
```

### Ход работы:

4. Даны действительные числа  $x$  и  $y$ , не равные друг другу. Меньшее из этих двух чисел заменить половиной их суммы, а большее — их удвоенным произведением.

```
const inquirer = require('inquirer');

class ValidationError extends Error {
    constructor(message) {
        super(message); // (1)
        this.name = "ValidationError"; // (2)
    }
}

inquirer
    .prompt([
        { type: 'number', message: 'Введите первое число:', name: 'x' },
        { type: 'number', message: 'Введите второе число:', name: 'y' },
    ])
    .then((answers) => {
        if (answers.x == answers.y) throw new ValidationError("X не может быть равен Y");
        let res = clac(answers.x, answers.y)
        console.log(answers.x = res.x)
        console.log(answers.y = res.y)
    })
    .catch((error) => {
        console.log(error);
    });

function clac(x, y) {
    if (x > y) {(y + x)/2
        return {x: (y * x)*2, y: (y + x)/2}
    } else {
        return {x: (y + x)/2, y: (y * x)*2}
    }
}
```

```
}
```

```
}
```

выполнение:

```
? Введите первое число: 3
? Введите второе число: 5
4
30
```

### Контрольные вопросы:

1. Назначение, формы записи и порядок выполнения оператора ветвления?

- для ветвления алгоритма

- ```
if (*условие*) {*то*}
else {*то*}
```

- Сначала вычисляется логическое выражение. Если результат есть true (истина), то выполняется оператор P<sub>1</sub> а оператор P<sub>2</sub> пропускается. Если результат есть false (ложь), то выполняется оператор P<sub>2</sub>, а оператор P<sub>1</sub> пропускается.

2. Как выглядит блок-схема программы с ветвлением?

ставлен на рис. 63.

