Основы программирования

ФИСТ 1 курс
Власенко
Олег
Федосович

Лекция 7.1
Стандартные типы.
Работа с консолью.
Развилки.
(Основные) логические операции

Основные типы данных

Тип	Длина байт	Диапазон значений	Минимальное значение	Максимальное значение
char				
double				
short				
long				

Составить таблицу символов

```
#include <stdio.h>

void main() {
	char ch = ' ';

	int i = 0;
	do {
		printf("%4d--> '%c'\t", ch, ch);
		ch = ch + 1;
		i = i + 1;
	} while (i <= 256);
}
```

Основные типы данных (ASCII)

Тип	Длина байт	Диапазон значений	Минимальное значение	Максимальное значение
char	**1**	**256**	**-128**	**+127**
double				
short				
long				

Подсчитать MAX short

```
void main() {
    short i = 1;
    long  n = 0;
    do {
        i = i + 1;
        n = n + 1;
    } while (i > 0);
    printf("%li\n", n);
}
```

Основные типы данных

Тип	Длина байт	Диапазон значений	Минимальное значение	Максимальное значение
char	1	256	-128	+127
double				
short				**32767**
long				

Сколько байт в short и long?

```
void main() {

    short i;
    long l;

    printf("sizeof short = %d\n", sizeof(i));
    printf("sizeof long = %d\n", sizeof(l));
}
```

Основные типы данных

Тип	Длина байт	Диапазон значений	Минимальное значение	Максимальное значение
char	1	256	-128	+127
double				
short	**2**			32767
long	**4**			

Основные типы данных

Тип	Длина байт	Диапазон значений	Минимальное значение	Максимальное значение
char	1	2^8 = 256	-128	+127
double	**8**	**IEEE 754 standard**	**2.22507e-308**	**1.79769e+308**
short	2	**2^16= 65 536**	**-32 768**	32767
long	4	**2^32 = +4,294,967,296**	**−2,147,483,648**	**+2,147,483,647**

Строка форматирования

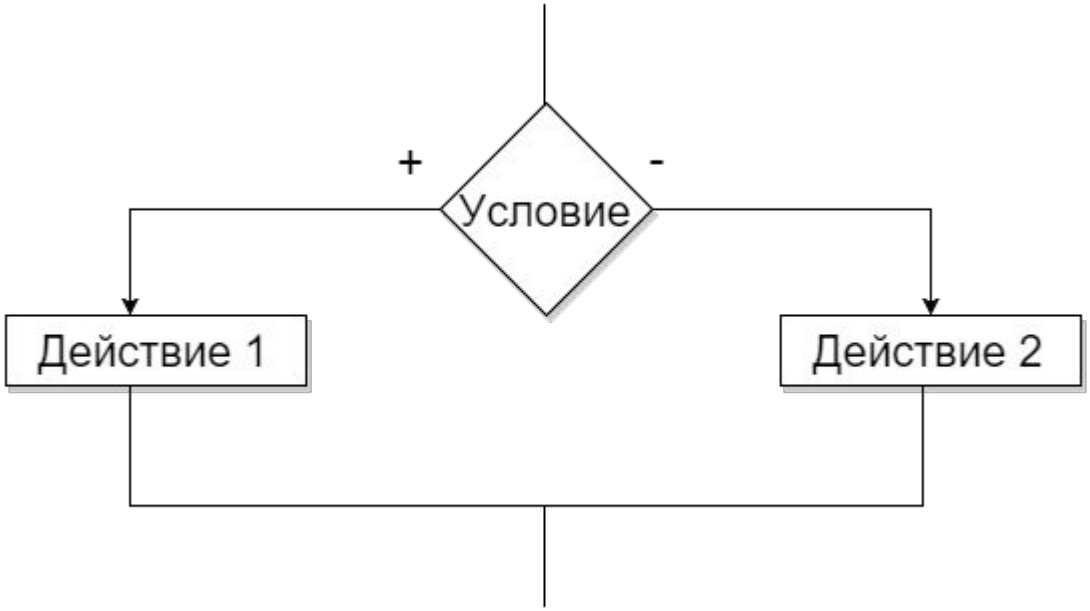
Тип	scanf/printf
char	%c
short	%hi
int	%d или %i
long	%li
float	%f
double	%lf
long double	%Lf

Консоль – что из себя представляет.
Знакоместо – что это такое.

Поиск корней квадратного уравнения

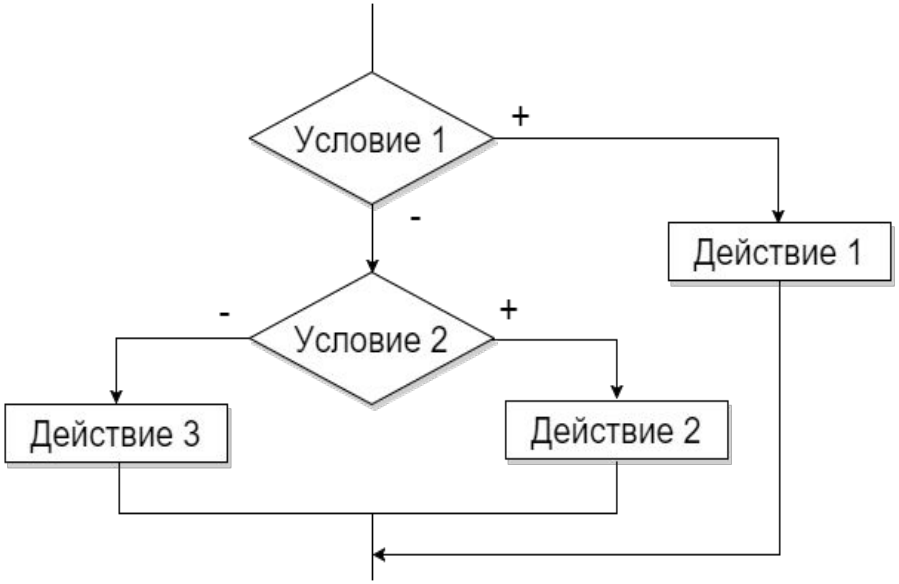
```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <math.h>

void main() {
	double a, b, c;
	double D;
	double x1, x2;

	scanf("%lf", &a);
	scanf("%lf", &b);
	scanf("%lf", &c);

	D = b * b - 4 * a * c;

	x1 = (-b + sqrt(D)) / (2 * a);
	x2 = (-b - sqrt(D)) / (2 * a);

	printf("x1 = %lf", x1);
	printf("x2 = %lf", x2);
}
```

Развилка




```
if (Условие)
    Действие1;
else
    Действие2;
```

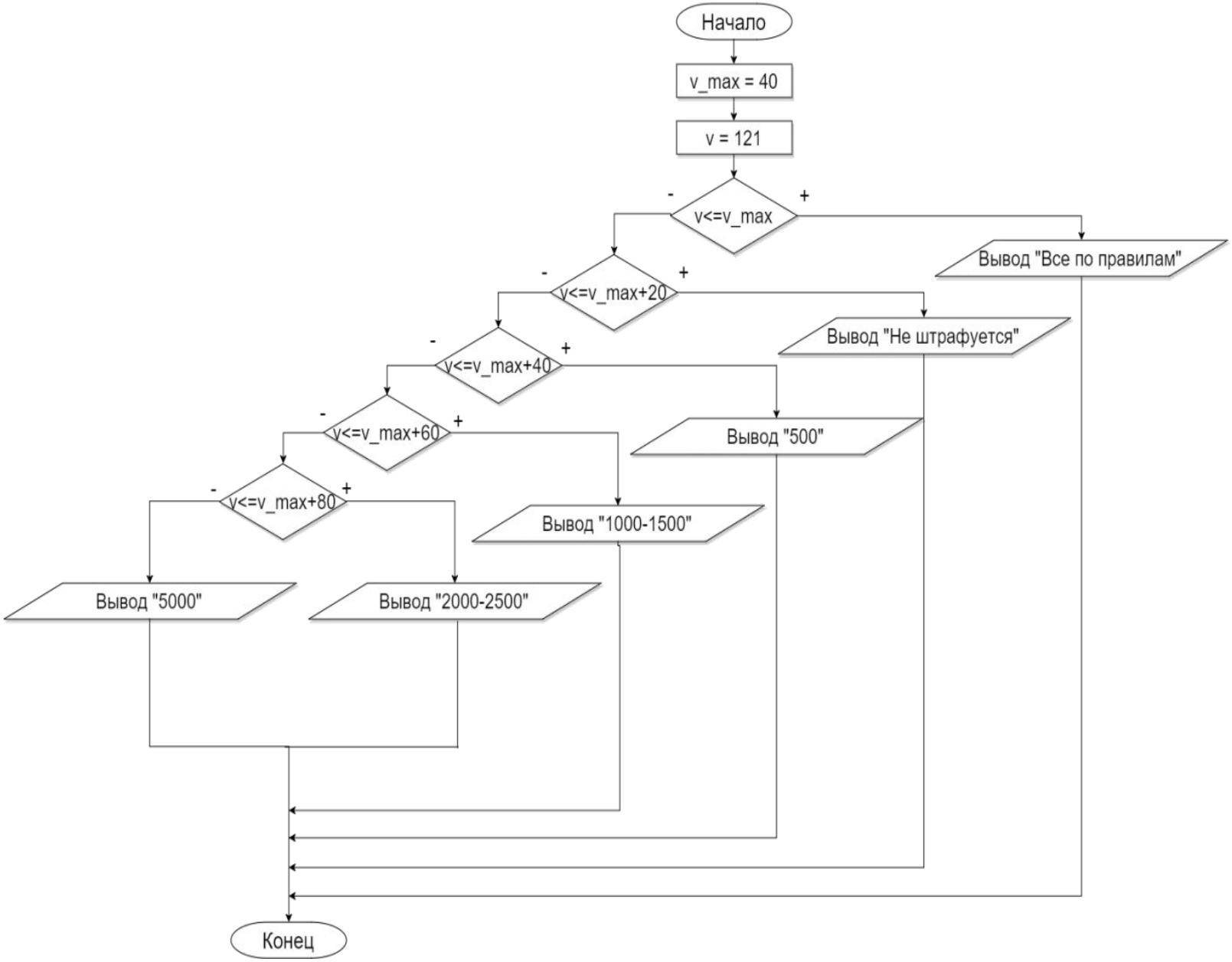
Вложенные развилки




```
if (Условие 1) {
        Действие 1
} else {
        if (Условие 2) {
                Действие 2
        } else {
                Действие 3
        }
}
```

```
if (Условие 1) {
        Действие 1
} else if (Условие 2) {
        Действие 2
} else {
        Действие 3
}
```

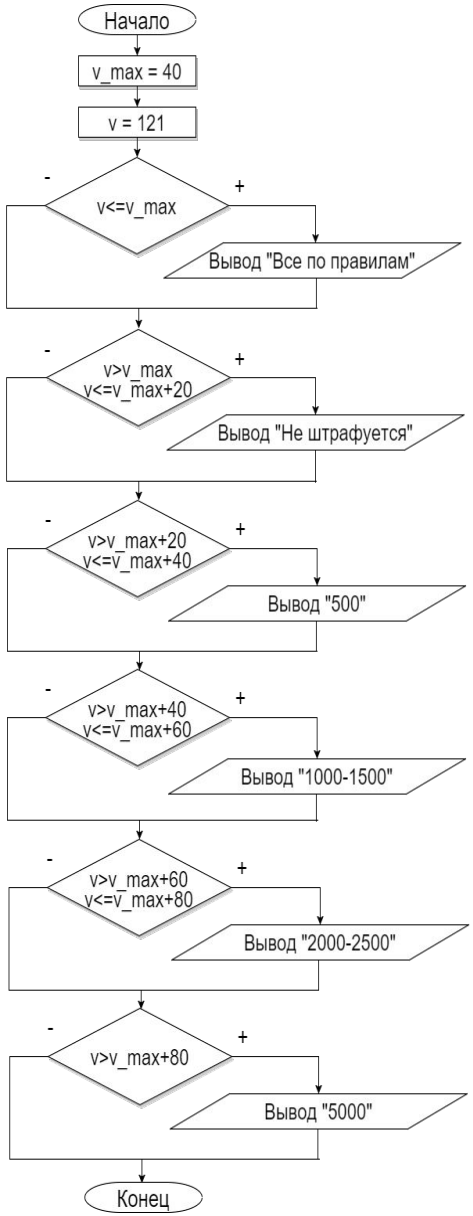
Штраф за превышение скорости



Штраф за превышение скорости – полная развилка

```
void main() {
	int v_max = 40;
	int v = 30;

	if (v <= v_max) {
		printf("All right!");
	} else if (v <= v_max + 20) {
		printf("No $$$");
	} else if (v <= v_max + 40) {
		printf("500");
	} else if (v <= v_max + 60) {
		printf("1000-1500");
	} else if (v <= v_max + 80) {
		printf("2000-2500");
	} else {
		printf("5000");
	}
}
```

Штраф за превышение скорости



Штраф за превышение скорости – усеченная развилка

```
void main() {
  int v_max = 40;
  int v = 70;

  if (v <= v_max) {
    printf("Все по правилам!");
  }
  if ((v > v_max) && (v <= v_max + 20)) {
   printf("не штрафуется");
  }
  if ((v > v_max + 20) && (v <= v_max + 40)) {
    printf("500");
  }
  if ((v > v_max + 40) && (v <= v_max + 60)) {
     printf("1000-1500");
  }
  if ((v > v_max + 60) && (v <= v_max + 80)) {
     printf("2000-2500");
  }
  if (v > v_max + 80) {
      printf("5000");
  }
}
```

Логические операции

Оператор	Описание
&&	Логическое И (AND)
||	Логическое ИЛИ (OR)
!	Логическое унарное НЕ (NOT)

A	!A
0	1
1	0

A	B	A && B	A || B
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

```
if (time < 7.00 || day >= 6)  rest();

if (!closed && money > 1000)  eat();
```

Домашнее задание

1. Дойти до предела long. Найти такую задачу, где нужны целые числа, а возможностей long недостаточно. (В идеале – реализовать её в коде)
2. Дойти до предела double. Найти задачу, где возможностей double недостаточно для вычислений.