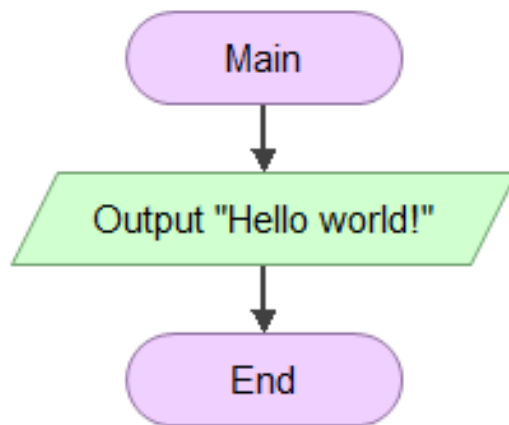




Основы программирования



Первая программа





Первая программа

```
#include <stdio.h>
int main()
{
    printf("Hello World! \n");
    return 0;
}
```

```
// подключаем заголовочный файл
// определяем функцию main
// начало функции
// выводим строку на консоль
// выходим из функции
// конец функции
```



Хранение данных

Тип данных	Назначение
int	Целое число
char	Символ
float	Вещественное число одинарной точности
double	Вещественное число двойной точности

Размер?

MS-DOS
(DOSBox)

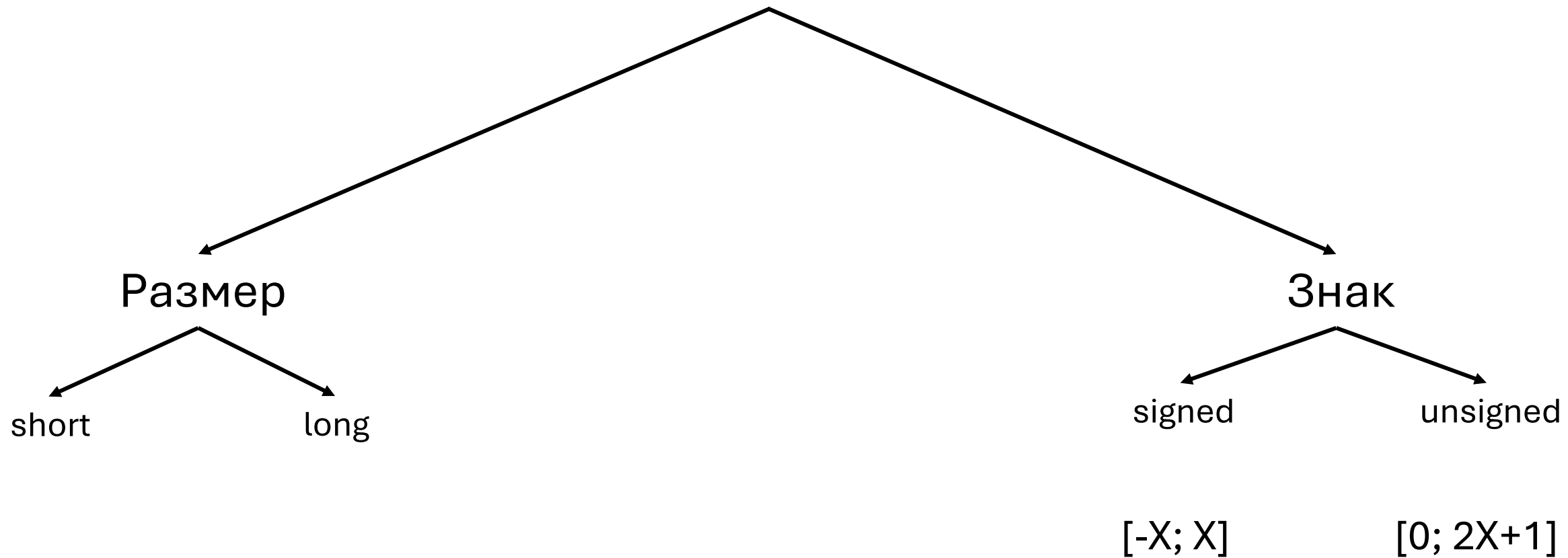
```
C:\PROGRAM>c:\Borlandc\bin\bc.exe  
Sizeof int = 2 bytes  
Sizeof char = 1 bytes  
Sizeof float = 4 bytes  
Sizeof double = 8 bytes
```

Win10

```
C:\Users\79629\Programs>varsize.exe  
Sizeof int = 4 bytes  
Sizeof char = 1 bytes  
Sizeof float = 4 bytes  
Sizeof double = 8 bytes
```



Хранение данных. Модификаторы.





Хранение данных. Модификаторы.

MS-DOS
(DOSBox)

```
C:\PROGRAM>c:\Borlandc\bin\bc.exe  
Sizeof int = 2 bytes  
Sizeof char = 1 bytes  
Sizeof float = 4 bytes  
Sizeof double = 8 bytes
```

With modifiers

```
Sizeof short int = 2 bytes  
Sizeof long int = 4 bytes  
Sizeof long long int = 4 bytes
```

Win10

```
C:\Users\79629\Programs>varsize.exe  
Sizeof int = 4 bytes  
Sizeof char = 1 bytes  
Sizeof float = 4 bytes  
Sizeof double = 8 bytes
```

With modifiers

```
Sizeof short int = 2 bytes  
Sizeof long int = 4 bytes  
Sizeof long long int = 8 bytes
```



Хранение данных

Тип данных	Размер	Минимальный диапазон значений
<code>signed char</code>	минимум 8 бит	от -127 ($= -(2^8-1)$) до 127
<code>unsigned char</code>	минимум 8 бит	от 0 до 255 ($= 2^8-1$)
<code>char</code>	минимум 8 бит	от -127 до 127 или от 0 до 255 в зависимости от компилятора
<code>short int</code>	минимум 16 бит	от -32,767 ($= -(2^{15}-1)$) до 32,767
<code>unsigned short int</code>	минимум 16 бит	от 0 до 65,535 ($= 2^{16}-1$)
<code>int</code>	минимум 16 бит	от -32,767 до 32,767
<code>unsigned int</code>	минимум 16 бит	от 0 до 65,535 ($= 2^{16}-1$)
<code>long int</code>	минимум 32 бита	от -2,147,483,647 до 2,147,483,647
<code>unsigned long int</code>	минимум 32 бита	от 0 до 4,294,967,295 ($= 2^{32}-1$)
<code>long long int</code>	минимум 64 бита	от -9,223,372,036,854,775,807 до 9,223,372,036,854,775,807
<code>unsigned long long int</code>	минимум 64 бита	от 0 до 18,446,744,073,709,551,615 ($= 2^{64}-1$)



Вывод данных

printf()	
Заголовочный файл	stdio.h
Входные данные	(const char *format, ...)
Возвращаемое значение	отрицательное значение — признак ошибки; в случае успеха функции возвращают количество записанных/выведенных байтов
Пример использования	printf("Good %s%c my dear %d students\n","evening",",",4);
	Good evening my, dear 4 students



Вывод данных

Код	Формат
%c	Символ типа char
%d	Десятичное число целого типа со знаком
%i	Десятичное число целого типа со знаком
%e	Научная нотация (е нижнего регистра)
%E	Научная нотация (Е верхнего регистра)
%f	Десятичное число с плавающей точкой
%g	Использует код %e или %f — тот из них, который короче (при использовании %g используется е нижнего регистра)
%G	Использует код %E или %f — тот из них, который короче (при использовании %G используется Е верхнего регистра)
%o	Восьмеричное целое число без знака
%s	Строка символов
%u	Десятичное число целого типа без знака
%x	Шестнадцатиричное целое число без знака (буквы нижнего регистра)
%X	Шестнадцатиричное целое число без знака (буквы верхнего регистра)
%p	Выводит на экран значение указателя
%n	Ассоциированный аргумент — это указатель на переменную целого типа, в которую помещено количество символов, записанных на данный момент
%%	Выводит символ %



Ввод данных

scanf()	
Заголовочный файл	stdio.h
Входные данные	(const char *format, ...)
Возвращаемое значение	отрицательное значение — признак ошибки; в случае успеха функции возвращают количество записанных/выведенных байтов
Пример использования	<pre>int num_of_students; char comma; char line[8]; scanf("%s",line); scanf(" %c",&comma); scanf("%d",&num_of_students); printf("Good %s%c my dear %d students\n",line,comma,num_of_students);</pre>
	Good evening my, dear 4 students

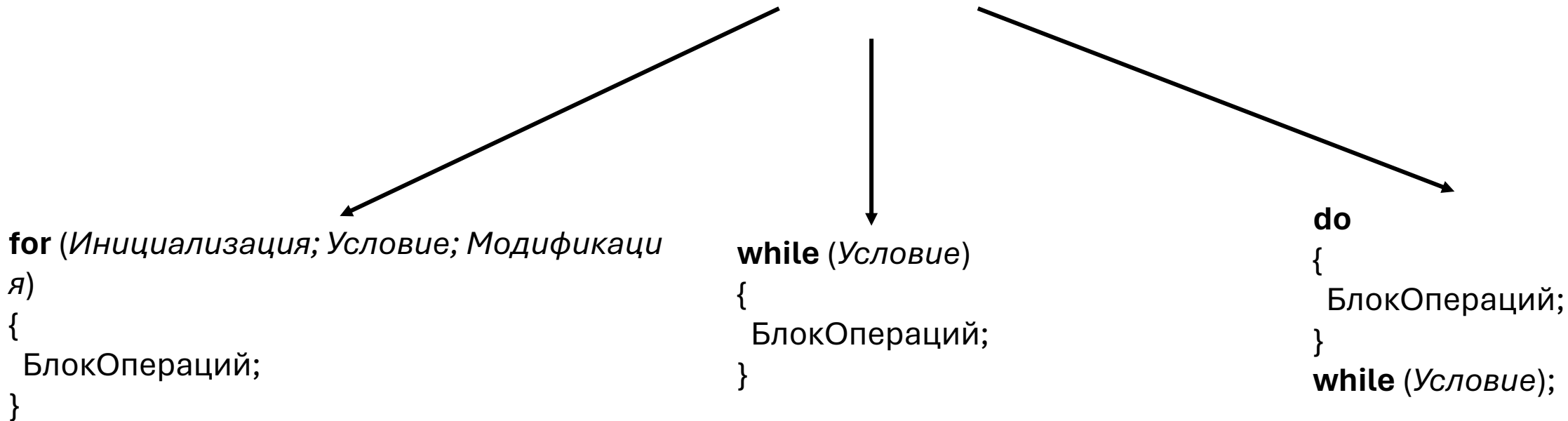


Обработка. Ветвление

Неполная развилка	Полная развилка
<pre>1 if (Условие) 2 { 3 БлокОпераций1; 4 }</pre>	<pre>1 if (Условие) 2 { 3 БлокОпераций1; 4 } 5 else 6 { 7 БлокОпераций2; 8 }</pre>



Обработка. Циклы



```
for (int i=0; i<10; i++)  
{  
    printf("%d. Hello MIET!\n", i);  
}
```

```
int i = 0;  
while (i < 10)  
{  
    printf("%d. Hello MIET!\n", i); i++;  
}
```

```
int i = 0;  
do  
{  
    printf("%d. Hello MIET!\n", i);  
    i++;  
}  
while (i < 10)
```