

ЛЕКЦИЯ 05

ОПЕРАТОРЫ В C++

АЛГОРИТМИЗАЦИЯ И
ПРОГРАММИРОВАНИЕ



ОСНОВНЫЕ ПОНЯТИЯ

Операторы управляют процессом выполнения программы. Набор операторов языка C++ содержит все управляющие конструкции структурного программирования.

Составной оператор ограничивается фигурными скобками. Все другие операторы заканчиваются точкой с запятой.

ПУСТОЙ ОПЕРАТОР

Пустой оператор – это оператор, состоящий только из точки с запятой. Он может появиться в любом месте программы, где по синтаксису требуется оператор. Выполнение пустого оператора не меняет состояния программы.

ПУСТОЙ ОПЕРАТОР

```
int main() {  
    ; ; ; ;  
    return 0;  
}
```

СОСТАВНОЙ ОПЕРАТОР

Действие составного оператора состоит в последовательном выполнении содержащихся в нем операторов, за исключением тех случаев, когда какой-либо оператор явно передает управление в другое место программы.

```
{ ... }
```

ОПЕРАТОРЫ СРАВНЕНИЯ

Все операторы сравнения являются бинарными (имеют два операнда) и возвращают всегда значения типа **bool**.

- Оператор больше >
- Оператор меньше <
- Оператор не больше <=
- Оператор не меньше >=
- Оператор равно ==
- Оператор не равно !=

ОПЕРАТОРЫ МАТЕМАТИЧЕСКИХ ДЕЙСТВИЙ

- Бинарный оператор плюс +
- Унарный оператор плюс +
- Бинарный оператор минус -
- Унарный оператор минус -
- Бинарный оператор умножить *
- Бинарный оператор делить /
- Бинарный оператор делить по модулю %
- Унарный оператор преинкремента ++
- Унарный оператор постинкремента ++
- Унарный оператор предекремента --
- Унарный оператор постинкремента --

ОПЕРАТОРЫ ПРИСВАИВАНИЯ

Все операторы присваивания являются бинарными и возвращают ссылку на левый операнд

- Оператор присваивания =
- Оператор составного присваивания и сложения +=
- Оператор составного присваивания и вычитания -=
- Оператор составного присваивания и умножения *=
- Оператор составного присваивания и деления /=
- Оператор составного присваивания и деления по модулю %=

ЛОГИЧЕСКИЕ ОПЕРАТОРЫ

Все логические операторы возвращают значение типа `bool`

- Бинарный оператор логического И `&&`
- Бинарный оператор логического ИЛИ `||`
- Оператор логического НЕ `!`

БИТОВЫЕ ОПЕРАТОРЫ

- Бинарный оператор побитового И (конъюнкция) &
- Бинарный оператор побитового ИЛИ (дизъюнкция) |
- Унарный оператор побитового НЕ ~
- Бинарный оператор побитового XOR ^
- Бинарный оператор побитового сдвига влево <<
- Бинарный оператор побитового сдвига вправо >>

ИНЫЕ ОПЕРАТОРЫ

- Оператор разрешения контекста ::
- Оператор запятая ,
- Оператор круглые скобочки ()
- Оператор квадратные скобочки []
- Оператор кавычки “”
- Тернарный условный оператор ?:

ИНЫЕ ОПЕРАТОРЫ

- Оператор разрешения контекста ::
- Оператор запятая ,
- Оператор круглые скобочки ()
- Оператор квадратные скобочки []
- Оператор кавычки “”
- Тернарный условный оператор ?:

ОСНОВНЫЕ ПОНЯТИЯ

Выражение – это последовательность операторов и их операндов, которые определяют вычисление.

Выражения могут порождать результат и иметь побочные эффекты. Например, выражение $2 + 2$ порождает результат 4, а выражение `std::printf(“%d”, 6)` в качестве побочного эффекта имеет вывод на экран 6.

Для корректного вычисления все операторы имеют свои приоритеты и ассоциативность.

ПРИОРИТЕТЫ ОПЕРАТОРОВ

Выражение – это последовательность операторов и их операндов, которые определяют вычисление.

Выражения могут порождать результат и иметь побочные эффекты. Например, выражение `2 + 2` порождает результат 4, а выражение `std::printf(“%d”, 6)` в качестве побочного эффекта имеет вывод на экран 6.

Для корректного вычисления все операторы имеют свои приоритеты и ассоциативность.

https://en.cppreference.com/w/cpp/language/operator_precedence

УТОЧНЕНИЕ

Побочный эффект — это то, что не может быть оптимизировано компилятором.

ПОТРЕНИРУЕМСЯ

Расставьте порядок выполнения операторов в следующем выражении:

```
int sum = 12 + 6;
```


ПОТРЕНИРУЕМСЯ

Расставьте порядок выполнения операторов в следующем выражении:

(2) (1)

```
int sum = 12 + 6;
```

ПОТРЕНИРУЕМСЯ

Расставьте порядок выполнения операторов в следующем выражении:

(2) (1)

```
int sum = 12 + 6;
```

```
bool b = a && b || !a
```

ПОТРЕНИРУЕМСЯ

Расставьте порядок выполнения операторов в следующем выражении:

(2) (1)

```
int sum = 12 + 6;
```

(4) (2) (3) (1)

```
bool b = a && b || !a
```

ПОТРЕНИРУЕМСЯ

Расставьте порядок выполнения операторов в следующем выражении:

(2) (1)

```
int sum = 12 + 6;
```

(4) (2) (3) (1)

```
bool b = a && b || !a
```

```
something::value += ++a - b * c
```

ПОТРЕНИРУЕМСЯ

Расставьте порядок выполнения операторов в следующем выражении:

(2) (1)

```
int sum = 12 + 6;
```

(4) (2) (3) (1)

```
bool b = a && b || !a
```

(1) (5) (2) (4) (3)

```
something::value += ++a - b * c
```

ЧАСТНОЕ И ОСТАТОК

Пусть X делится Y , тогда $X = p \cdot Y + q$, где

p – частное

q – остаток.

ЧАСТНОЕ И ОСТАТОК

Пусть X делится Y , тогда $X = p \cdot Y + q$, $q < Y$ где

p – частное

q – остаток.

Например, $X = 13$, $Y = 4$. Найти частное и остаток от X/Y

ЧАСТНОЕ И ОСТАТОК

Пусть X делится Y , тогда $X = p \cdot Y + q$, где

p – частное

q – остаток.

Например, $X = 13$, $Y = 4$. Найти частное и остаток от X/Y

$p =$

$q =$

ЧАСТНОЕ И ОСТАТОК

Пусть X делится Y , тогда $X = p \cdot Y + q$, где

p – частное

q – остаток.

Например, $X = 13$, $Y = 4$. Найти частное и остаток от X/Y

$$p = 3$$

$$q = 1$$

$$13 = 3 \cdot 4 + 1$$

ЧАСТНОЕ И ОСТАТОК

Пусть X делится Y , тогда $X = p \cdot Y + q$, где

p – частное

q – остаток.

Например, $X = 13$, $Y = -4$. Найти частное и остаток от X/Y

$p =$

$q =$

ЧАСТНОЕ И ОСТАТОК

Пусть X делится Y , тогда $X = p \cdot Y + q$, где

p – частное

q – остаток.

Например, $X = 13$, $Y = -4$. Найти частное и остаток от X/Y

$p = -3$

$q = 1$

$$13 = -3 \cdot (-4) + 1$$

ЧАСТНОЕ И ОСТАТОК

Пусть X делится Y , тогда $X = p \cdot Y + q$, где

p – частное

q – остаток.

Например, $X = -13$, $Y = -4$. Найти частное и остаток от X/Y

$p =$

$q =$

ЧАСТНОЕ И ОСТАТОК

Пусть X делится Y , тогда $X = p \cdot Y + q$, где

p – частное

q – остаток.

Например, $X = -13$, $Y = -4$. Найти частное и остаток от X/Y

$$p = 3$$

$$q = -1$$

$$-13 = 3 \cdot (-4) + (-1)$$

ЧАСТНОЕ И ОСТАТОК

Нахождение частного a и b возможно с помощью оператора $/$

$$p = a / b;$$

Нахождение остатка от деления a на b возможно с помощью $\%$

$$q = a \% b$$

Нахождение остатка от деления a на b еще называют делением a по модулю b и записывают так: $a \bmod b$

Например,

$$13 \bmod 2 = 1$$

$$11 \bmod 3 = 2$$

$$29 \bmod 8 = 5$$

ПОТРЕНИРУЕМСЯ

$$18 \bmod 5 =$$

$$10 \bmod 2 =$$

$$91 \bmod 6 =$$

$$22 \bmod 7 =$$

$$35 \bmod 4 =$$

$$1024 \bmod 10 =$$

$$5087 \bmod 100 =$$

$$32 \bmod 97 =$$

$$41 \bmod 5 =$$

ПОТРЕНИРУЕМСЯ

$$18 \bmod 5 = 3$$

$$10 \bmod 2 = 0$$

$$91 \bmod 6 = 1$$

$$22 \bmod 7 = 1$$

$$35 \bmod 4 = 3$$

$$1024 \bmod 10 = 4$$

$$5087 \bmod 100 = 87$$

$$32 \bmod 97 = 32$$

$$41 \bmod 5 = 1$$

ДОМАШНЕЕ ЗАДАНИЕ

1. Изучить функцию `sizeof()`
2. Написать программу, которая выводит таблицу, содержащую в себе наименование типа и его размер в байтах.
3. Написать программу, которая принимает от пользователя два числа, вычисляет и выводит на экран частное и остаток от деления двух целых чисел
4. Изобразить блок-схемы для ваших программ
5. Выучить типы данных, их размеры и диапазоны значений

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Дорохова Т.Ю., Основы алгоритмизации и программирования : учебное пособие для СПО / Т.Ю. Дорохова, И.Е. Ильина. — Саратов, Москва : Профобразование, Ай, Пи Ар Медиа, 2022. — 139 с.
2. Кудинов Ю.И., Основы алгоритмизации и программирования : учебное пособие для СПО / Ю.И. Кудинов, А.Ю. Келина. — 2-е изд. — Липецк, Саратов: Липецкий государственный технический университет, Профообразование, 2020. — 71 с.
3. Дональд Кнут, Искусство программирования. Том 1. Основные алгоритмы / Ю.В. Козаченко. - 3-е изд — Москва, Санкт-Петербург: ВИЛЬЯМС, 2018. — 721 с.