

# Операторы

- ▶ Операторы - Получают один/много значений и на их основе создают новое значение
- ▶ Работают с операндами и выдают результат
- ▶ Побочный эффект - оператор изменяет значение операнда
- ▶ Почти все (кроме присвоения сравнения) работают с примитивными типами (исключение - строки)
- ▶ Существует приоритет выражений
- ▶ Присваивание - = - взять значение и скопировать его в выражение

Нельзя присвоить ничего константе, для примитивных типов присваивание есть копирование значение, для объектов - присваивание - есть копирование ссылки

По скольку ссылка копируется, два объекта указывают на одну область памяти, тогда изменение одного порождает изменение другого

## ▶ Мат. Операции

Стандартные операции, при делении - не округление, а усечение

Унарный минус - меняет знак, Унарный плюс - чтобы было

++ -- - инкремент, декремент

Префиксные - изменение до результата, постфиксные - после

## ▶ Операторы сравнение < > == != <= >=

Если мы упакуем примитив (int -> integer), то будет сравнение ссылок и false

▶ Equals - сравнивает ссылки, можно переопределить для правильного сравнение объектов

▶ && || ! - только к булеан

▶ Ускоренное вычисление - логические операции проводятся до тех пор, пока точно не истина или ложь -> части логического сравнения могут опускаться

## ▶ Литералы

Помогают определить тип переменных (16-ричная, 8-ричная, double, long, float...) e - не exp, а  $10^{\wedge}$

## ▶ Подразрядные операторы

Манипуляция битами в целых числах:

Производится перевод в двоичную и выполняются мат операции с двоичными

& - и, | - или, ^ - xor (по модулю два), - могут комбинироваться с =, ~ - не

Операторы сдвига

сдвиг влево << - сдвигает влево указанное количество битов

сдвиг вправо >> - сдвигает вправо (если положительное значение - то пустые - нули, отрицательное - пустые единицы)

>>> - беззнаковые сдвиг вправо - только заполнение нулями

Все типы для сдвига приводятся к int

- ▶ Тернарный оператор использует три операнда:

Логическое условие : выражение0 ? Выражение1

- ▶ + и += Для строк

Если выражение начинается строкой, то все последующие операнды - строки

- ▶ Ошибки, связанные с | || - предотвращает компилятор

- ▶ Операторы приведения

Явное приведение - `int a = (int)4,6575`

Неявное - `int a = (int)`

Явное приведение используется в ситуациях, где возможна потеря данных (от большего берем меньшее)

`Animal sharick = new Animal();`

`Dog myDog = (Dog)sharick;`

Наоборот оно не нужно

- ▶ Округление и усечение

Усечение - при приведении к целым типам, если нужно округление - `math.round()`

При использовании разных типов результат получается в виде меньшего типа

- ▶ В случае операций с `char`, `short`, `byte` - результат в виде большего типа `int` => необходимо явно приводить все назад
- ▶ Любой примитивный тип может быть приведен к примитивному типу кроме булеан