**Операторы цикла for и for-each. Работа с элементами массива**

Вы уже знаете, что цикл **for** является наиболее распространённым циклом в программировании. Его удобно применять, когда известно количество итераций.

Цикл **for** подходит для перебора элементов массива, так как:

* представляет компактную форму описания цикла **while**
* управление циклом вынесено из тела цикла.

***Создание массива предполагает следующее:***

* Создаются три ссылки на массивы типа **int**, но только две инициализируются: *arrayInt* и *amountDays*.
* Организуется цикл для копирования всех элементов массива *amountDays* в массив *arrayInt*.
* Перед выполнением копирования текущего элемента массива *amountDays* производится изменение его значения, если оно меньше 31.
* После завершения цикла происходит инициализация ссылки *Array*: она тоже станет ссылаться на массив *amountDays*.

На каждой итерации цикла копия значения элемента массива сохраняется в его итерационной переменной (цикл выполняется до тех пор, пока не будут обработаны все элементы набора данных). Однако завершить цикл **for-each** можно и раньше, используя оператор **break**.

*Достоинства цикла****for-each****:*

* Упрощает синтаксис оператора.
* Исключает возможность выхода за пределы массива при его переборе.
* Используется исключительно для перебора значений, а не для какого-либо другого вида операций (например, удаления или редактирования элементов массива).

Цикл **for-each** имеет следующие преимущества:

* Упрощенность представления.
* Нет необходимости ведения дополнительной переменной цикла, установки ее начального значения и условия завершения цикла.
* Нет необходимости индексировать массив.

Цикл **for-each** имеет следующие недостатки:

* Отсутствие гибкости при манипулировании итерационной переменной.
* Невозможность изменить значение элементов массива (для примитивных и неизменяемых ссылочных типов данных).