

Здравствуй, курс джава начинается здесь

Начнём с супер простых штук, банально чтобы натренировать мышечную память на лексику языка и в будущем делать всё быстрее. Да и в целом сложный код всегда состоит из более простых частей (именно эта проблема зачастую мешает делать сложные задания).

Первым заданием будет вот такое:

1) Ввод числа с консоли, проверка его на чётность/нечётность и последующий вывод числа с дополнением о том, чётное оно или нет.

И так нам здесь по-сути надо просто создать if конструкцию, которая будет проверять чётность с помощью остатка от деления числа на два (помним о том, что если при таком делении есть остаток, число будет нечётным).

Попробуй сама, все ответы выгружу на [github](#) и дам тебе ссылку конечно же.

И так перейдём дальше

2) Проверка числа на положительность/отрицательность либо равенство нулю. Для этого нужно ввести число через консоль и через конструкции if/else делать проверку. Также надо помнить про проработку ошибок в конце, так как человек может ввести неправильный тип данных или сделать что-то ещё не так.

Следующее

3) Теперь для закрепления if/else можно сделать задание на дни недели, где нужно принять с ввода число от 1 до 7 и вывести день недели, соответствующий этому числу (1 - понедельник и т.д.).

Ты наверное можешь сказать, что для такого есть оператор switch, но как раз для того, чтобы в полной мере понимать его значимость, нужно сначала сделать это задание через if/else.

Дальше

4) Теперь делаем то же самое задание с днями недели, но уже через switch. Операторы, принимающие в себя переменные, довольно сложные для понимания поначалу, поэтому в случае чего, всегда спрашивай, так как например у меня были с этим проблемы, а спросить было особо не у кого.

Сделав это задание, можно сравнить предыдущий код и этот, после чего появится вопрос "Действительно ли так удобнее?", так как через switch необходимо каждый раз прописывать break в конце. И всё же по-сути основным преимуществом switch является более понятный вид, а удобность проявляется не всегда, поэтому тут уже зависит от условий, знать этот оператор всё равно полезно.

Следующее

5) Цикл while довольно удобный, поэтому теперь с его помощью тебе нужно сделать код, который будет выводить определённое число каких-либо строк (можно даже сделать, чтобы человек сначала ввёл строку, а потом количество повторений, после чего наблюдал за

результатом). Здесь у тебя свобода для реализации, но всё же основное требование - использование `while`, который работает так:

```
while (условие) {  
    действие;  
}
```

Так же не стоит забывать, что после действия, нам надо изменить переменную, используемую в условии, чтобы цикл не повторялся бесконечно (для этого в конце действия добавляем например `i++` или `i--`).

Этот цикл нужен, если ты не знаешь точного количества повторений (хотя при работе, например, с массивами, похожего результата можно добиться с помощью цикла `for`, но к этому мы ещё вернёмся).

6)Так же можно сразу рассмотреть цикл `do-while`, который отличается тем, что он выполнится минимум раз точно, независимо от соблюдения условий, то есть:

```
do {  
    действие;  
}  
while (условие);
```

Можешь попробовать сделать прошлое задание через этот цикл, чтобы просто закрепить знание, в этом случае заданная строка напечатается как минимум один раз в любом случае (даже если введём 0 как количество повторений).

7)Цикл `for` нужен, когда мы заранее знаем, сколько раз нам надо выполнить что-то. Для примера можно написать код, который будет вычислять факториал введённого числа. Попробуй сделать такое. Помни, что факториал это умножение всех чисел, идущих до нашего нужного включительно, то есть $n! = 1*2*3*...*n$.

В программировании часто нужно смотреть нужные формулы где-то и это нормально, главное уметь использовать найденные знания правильно.

Цикл `for` состоит по-сути из двух главных вещей - создание переменной, используемой для проверки условия, и самого условия (ну и инкремент/декремент, но это необходимо в любом цикле). Основной вид такой:

```
for (int i = 0; i < что-то; i++) {  
    действие;  
}
```

Где сначала мы создаём переменную, используемую только внутри этого цикла через `"int i = 0"`, после чего мы задаём условие, например `"i < 10"` или любое другое, в конце мы задаём то, как будет меняться наша переменная после каждого прохода цикла, в нашем случае это `"i++"`, то есть переменная будет увеличиваться на 1 после каждого повторения цикла.

Можно спросить "Зачем нам в целом цикл `for`?". В основном в будущем в работе с массивами он очень нам поможет, так как позволяет перебирать элементы массива по-очереди, но в более простых задачах он является банально более понятной версией `while`, так как в нём сразу в одной строке видно все условия цикла, да и он позволяет сразу легко создать новую переменную, которая не будет потом лежать без дела после выполнения

цикла.

В целом на этом всё по основным операторам, если не брать в расчёт алгебраические и математические операторы, но их, я уверен, ты знаешь, да и в целом они почти во всех языках одинаковые (имею ввиду вот это сложение, вычитание, деление и т.д.).

Для закрепления вот несколько задачек по ним просто чтобы ты вводила это достаточно долго, чтобы тратить время на решение задачи, а не на то, как правильно пишется оператор или что-то ещё. Всегда при решении задачек старайся сначала выделить для себя то, каким способом это нужно сделать и какие структуры использовать.

Простые задачки заключаются в простом - вывод чисел от 1 до 10; вывод каждого второго числа в диапазоне между двумя числами, введёнными с консоли (надо будет так же сделать проверку на то, какое число меньше, заодно закрепишь if/else); ещё можно взять вывод всех чётных/нечётных чисел в диапазоне двух чисел.

На этом в целом можно закончить с условиями и циклами, главное понять суть их работы и попрактиковаться, а дальше уже станет намного проще.