

Методический материал

Создаём приложение «Штрафы за превышение скорости» на Java

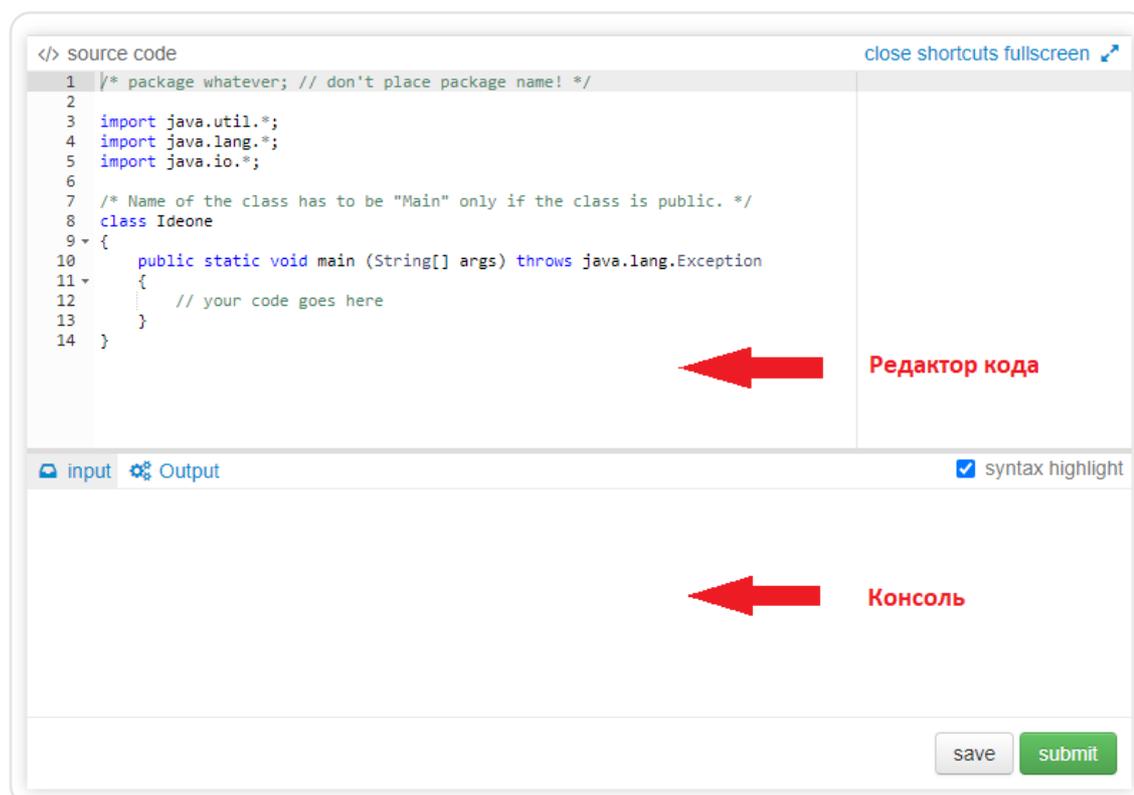
Java — язык программирования, на котором можно создавать приложения практически для любых устройств и операционных систем. Но основное, что пишут на Java — это веб-приложения. Точнее, их серверную часть — backend-составляющую.

Давайте напишем небольшое приложение прямо сейчас! Это будет уменьшенная версия системы, которая определяет штрафы за превышение скорости на дороге. Воспользуемся онлайн-редактором IdeOne — <https://ideone.com/>.

Пишем первый код

В окне редактора уже есть код и комментарий к нему — он выделен символами “//” и зелёным цветом. Комментарий нам не нужен, сотрите его. На остальной код пока не обращайтесь внимания.

Первое, что мы сделаем — напишем код, который будет печатать текст с названием нашей программы в консоль. Консоль — это специальное текстовое поле, которое есть в любой среде разработки. В него программа выводит текст.



```
<> source code close shortcuts fullscreen ↗
1  /* package whatever; // don't place package name! */
2
3  import java.util.*;
4  import java.lang.*;
5  import java.io.*;
6
7  /* Name of the class has to be "Main" only if the class is public. */
8  class Ideone
9  {
10     public static void main (String[] args) throws java.lang.Exception
11     {
12         // your code goes here
13     }
14 }
```

← Редактор кода

← Консоль

save submit

Туда, где раньше был комментарий «// your code goes here», пишем:

```
System.out.println("Система расчёта штрафов");
```

И нажимаем кнопку «Run». Программа компилируется, запускается — и строка текста «Система расчёта штрафов» появляется в консоли.

Конструкция `System.out.println` может показаться сложной. На самом деле, к ней просто нужно привыкнуть. Это самая сложная конструкция, которую вы сегодня увидите. Всё остальное будет проще.

Давайте теперь зададим скорость условного автомобиля, которую зафиксировала камера. Для этого создадим переменную. Переменная — это место в программе, где хранятся данные: числа, строки или объекты. С помощью переменной `carSpeed` мы зададим скорость автомобиля. Пока установим конкретную цифру, но в реальную программу она, конечно, должна попадать с камеры.

Ниже пишем:

```
int carSpeed = 187;
```

Вы можете задать любую скорость, но нам важно, чтобы она превышала разрешённую скорость в населённом пункте. Об этом — ниже.

«Int» означает, что дальше будет задана переменная, в которой хранится целое число, а `carSpeed` — имя переменной. Его принято писать с маленькой буквы, а каждое следующее слово в переменной — с большой. Знак «равно» означает, что мы присваиваем переменной `carSpeed` значение 187. Дальше — точка с запятой, на неё в Java должны оканчиваться все основные строки программного кода.

Мы задали переменную. Теперь её можно использовать в коде. К примеру, мы можем распечатать это число в консоль с помощью уже знакомой конструкции `System.out.println`. Для решения задачи печатать это число в консоль не потребуется, но можете сделать это для себя, чтобы попрактиковаться.

```
System.out.println(carSpeed);
```

Обратите внимание: кавычки внутри скобок нужны в том случае, если нам нужно напечатать обычный текст. Когда мы хотим напечатать значение переменной, кавычки не нужны

Теперь давайте зададим суммы штрафов, которые должны выписываться нашей системой. Штраф по-английски — `fine`, так и назовём переменные.

По правилам дорожного движения при превышении скорости на 20-40 км/ч, штраф должен составлять 500 рублей. На 40 до 60 км/ч — 1000 рублей. И так далее. Создадим 5 переменных для выписки штрафов.

```
int fineFor20to40 = 500;
```

```
int fineFor40to60 = 1000;
```

```
int fineFor60to80 = 2000;
```

```
int fineFor80andMore = 5000;
```

Представим, что отслеживаем скорость в населённом пункте. Значит разрешённая скорость — 60 км/ч. Зададим это условие в виде переменной и назовем её `townSpeed`.

```
int townSpeed = 60;
```

Основная задача программы — проверить, превышена ли скорость автомобилем. Если превышена больше, чем на 20 км/ч — выписать штраф. Давайте напишем код, который будет проверять, насколько превышена скорость: вычитать из скорости автомобиля максимально разрешённую скорость в населённом пункте.

```
int overSpeed = carSpeed - townSpeed;
```

Мы задали все необходимые условия. Чтобы программа работала так, как мы задумали, нужно написать код. Он будет проверять, нарушила машина допустимую скорость или нет. Если нарушила — значит, программа выпишет штраф. Чтобы это проверить, используем оператор `if`: он будет проверять выполнение условия. Само условие пишем в круглых скобках:

```
if(overSpeed < 20) {  
}
```

Далее — фигурные скобки. Если условие, которое мы написали в круглых скобках, выполнится — то код в фигурных скобках тоже выполнится. В случае, если превышение скорости меньше, чем на 20 км/ч, штраф не выписывается. Так и отразим в консоли:

```
if(overSpeed < 20) {  
    System.out.println("Скорость не превышена или превышена  
    незначительно");  
}
```

Пишем следующее условие:

```
if(overSpeed >= 20 && overSpeed < 40) {  
    System.out.println("Штраф: " + fineFor20to40);  
}
```

Оператор && означает логическое «и». То есть, условие целиком выполняется только в том случае, если и первое, и второе условия верны. Что в итоге наше условие означает? **Если скорость больше или равна 20, но меньше 40, штраф составит 500 рублей.** В переменную fineFor20to40 мы уже положили значение 500. Слово «штраф» будет напечатано в консоли, а знак «+» означает объединение строк.

Соответственно, если условие будет соблюдено, в консоли появятся две строки:

Система расчёта штрафов

Штраф: 500

Продолжаем писать условия по тому же принципу.

```
if (overSpeed >= 40 && overSpeed < 60) {  
    System.out.println("Штраф: " + fineFor40to60);  
}  
  
if (overSpeed >= 60 && overSpeed < 80) {  
    System.out.println("Штраф: " + fineFor60to80);  
}  
  
if (overSpeed >= 80) {  
    System.out.println("Штраф: " + fineFor80andMore);  
}  
}  
}
```

Не забываем про синтаксис: точки с запятой и фигурные скобки.

Запускаем, смотрим.

Всё замечательно, но есть проблема: каждое наше условие выполняется всякий раз, когда мы запускаем программу. Даже если скорость не будет превышена, программа всё равно проверит каждое условие. А если бы программа была большая, то работала бы медленно.

Давайте сделаем так, чтобы условия прекращали проверяться, когда будет найдено нужное соответствие. Добавим оператор `else` — «иначе» или «в противном случае». Он будет выполнять код, если условие в операторе `if` не выполнилось. Например:

```
if (overSpeed < 20) {  
  
    System.out.println(«Скорость не превышена или превышена  
    незначительно»);  
  
}  
  
else { //В противном случае  
  
    System.out.println(«Скорость превышена»);  
  
}
```

Написанное в фигурных скобках выполнится только в том случае, если условие в `if` не выполняется. Очевидно, что выполнится именно второе условие — так как мы уже задали автомобилю скорость 187 км/ч, и штраф однозначно будет.

Операторы `else` и `if` можно комбинировать. Например:

```
if (overSpeed < 20) {  
  
    System.out.println(«Скорость не превышена или превышена  
    незначительно»);  
  
}  
  
    else if (overSpeed >= 20 && overSpeed < 40)  
  
    ...
```

В этом случае произойдёт следующее: если условие `overSpeed < 20` не будет выполнено, сработает оператор `else` — и произойдёт проверка следующего условия `overSpeed >= 20 && overSpeed < 40`. И так дальше по цепочке.

А если какое-то из условий выполнится, то следующие условия за ним уже не сработают, и работа программы прекратится.

Допишем код до конца и добавим оператор `else if` к оставшимся условиям.

```
else if (overSpeed >= 60 && overSpeed < 80) {  
    System.out.println("Штраф: " + fineFor60to80);  
}  
  
else if (overSpeed >= 80) {  
    System.out.println("Штраф: " + fineFor80andMore);  
    }  
}  
}
```

Давайте запустим код и убедимся, что всё работает. Если нет — сверьтесь с [ГОТОВЫМ КОДОМ](#).

Поздравляем! Ваше приложение «Штрафы за превышение скорости» готово! Теперь вы знаете, что программировать на Java не так сложно, как вы думали.

Записывайте на профессию «Java-разработчик», научитесь писать код и станьте востребованным специалистом, даже если вы раньше никогда не программировали.

[Записаться на курс](#)