Статическая инициализация

Привет!

В одном из прошлых уроков вы изучали статические переменные, значения которых задаются в объектах — например, в конструкторах или методах. В этом уроке мы поговорим о том, как инициализировать статические переменные без создания объектов.

Например, в классе Basket вы хотите однозначно установить время, через которое корзина должна очищаться автоматически. К примеру, это неделя. А время нужно задать в секундах. Можно, конечно, написать так:

```
private static int timeout = 604800;
```

Но, во-первых, чтобы задать такое значение, его нужно еще посчитать, а во-вторых, по такому значению будет непонятно, сколько это времени. Если же вы напишете:

```
private static int timeout = 3600 * 24 * 7;
```

то сразу будет понятно, что это — неделя. Но здесь всё просто: при объявлении статической переменной просто задаём её значение.

Если же у вас более сложная логика, то можно использовать статической инициализации. Например:

```
private static int timeout;
static {
      int secondsInHour = 3600;
      int hoursInDay = 24;
      int daysInWeek = 7;
      int daysInMonth = 30;
timeout = secondsInHour * hoursInDay *
(Period.type == PeriodType.WEEK ?
daysInWeek : daysInMonth);
}
```

Такой блок в программе будет выполняться всего один раз — при любом первом обращении к этому классу. В последнее время блоки статической инициализации в программном коде встречаются всё реже, и лучше вместо них использовать статические методы, это более общепринятая практика:

```
private static int timeout = getTimeout();

private static int getTimeout() {
      int secondsInHour = 3600;
      int hoursInDay = 24;
      int daysInWeek = 7;
      int daysInMonth = 30;
timeout = secondsInHour * hoursInDay *
(Period.type == PeriodType.WEEK ?
daysInWeek : daysInMonth);
}
```

Вот так вы можете инициализировать статические переменные без создания объектов: сразу при их объявлении, в блоках “static” или в статических методах.

Глоссарий

блок статической инициализации