What are “sugar”, “desugar” terms in context of Java 8?

96

15

I hear about 'sugaring' and 'desugaring' more often in Java 8, what does these terms mean ? are they conceptual or syntactical.

Some Example:

> Default iterated loop resugaring to java
>
> Observations about syntactic sugar in compilation.

java semantics java-8

edited Feb 28 '14 at 6:22

Denys Séguret
289k 60 610 618

asked Feb 27 '14 at 6:39

Xelian
7,333 15 65 114

3 Answers

116

sugar, in programming, usually refers to those *sweet* additions, mostly shortcuts, that make some constructs easier to type and to read (the latter being, in practice, the most important during the life cycle of your program).

Wikipedia has a definition of syntactic sugar but you should note that not all sugar is, in essence, syntactical (not all recent sweet additions were just compiler changes).

Here are a few examples :

- the postfix and prefix increment operators (`i++` and `++i`). Their only purpose is to avoid writing an additional statement. They're pure sugar.
- `+=` , `|=` , `&=` , etc. are made of the same kind of sugar.
- Implicit conversion between primitive types and objects is sugar too.

Java is widely seen as not being concise enough, especially compared to modern languages. That's why those additions that help make the code faster to read are welcome.

To finish, I'd just note that while a lack of sugar can make your program fat, an excess of sugar, leading to many different ways to write the same things, can make your language queasy and your program less coherent and harder to maintain. Another kind of sugar, API sugar, is most often a plague which makes the API harder to grasp, especially when it's made of additions (overloading for example).

This being said, *desugaring* refers either to

- the process by which you remove all that is redundant in a language
- the process by which a code processor finds out what's behind a sugared statement (this may for example involves type inference)

edited Mar 28 at 5:45

Hearen
3,332 1 19 32

answered Feb 27 '14 at 6:41

Denys Séguret
289k 60 610 618

6 +1 You can also mention about the desugar part as that would complete the answer which OP has asked! :) – Rahul Tripathi Feb 27 '14 at 6:57

5 @justhalf Think how much thinner your programs will be when you'll be able to remove all those interfaces you create just to be able to pass a function. – Denys Séguret Feb 27 '14 at 7:37

38 "Syntactic sugar causes cancer of the semicolon." -- Alan Perlis – Stuart Marks Feb 27 '14 at 8:14

12 Postfix and prefix operators were once, in the early days of C, not syntactic sugar. Early compilers were dumb, by today's standards, and those operators could be implemented with one machine code instruction, but the equivalent assignment statement could not. So they enabled writing more efficient programs. – Raedwald Feb 27 '14 at 8:32

3 @justhalf This question got a lot of attention. It's not rare in those cases to have admins delete comments which seem not constructive. I don't think your comment had to be deleted but it didn't bring a lot. – Denys Séguret Mar 12 '14 at 7:05

"Desugaring" appears to have a very specific meaning in Java 8. It seems to be a catch-all term to express the various ways a lambda expression may be bound to an actual concrete method call.

14 This document on "Translation of Lambda Expressions" seems to have the real details of what's going on if you're interested in specifics.

A key phrase from the document:

edited Sep 6 '18 at 0:23

answered Mar 5 '14 at 10:02

Shorn

7,685 5 41 84

5

In general "desugaring" in javac allows representing some language features with preexisting ones. This allows representing them in the bytecode without making big changes to the class file format. Also for this reason the back-end of the compiler is more stable than the front-end. This doesn't mean that every new language feature is just syntactic sugar, as is definitely not the case of lambdas and method references. There are more examples of "desugaring" in the compiler:

- for each loops are "desugared" to C style for loops
- assertions are "desugared" to an if sentence
- inner classes are represented as a standalone class

You can investigate also what happen with the String switch, type erasure,...

answered May 26 '14 at 3:05

Vicente Romero

1,113 9 14
