

Internal distribution code:

- (A) [-] Publication in OJ
- (B) [-] To Chairmen and Members
- (C) [-] To Chairmen
- (D) [X] No distribution

**Datasheet for the decision
of 11 October 2022**

Case Number: T 1105/17 - 3.5.06

Application Number: 06255752.5

Publication Number: 1785852

IPC: G06F9/44

Language of the proceedings: EN

Title of invention:

Supporting method references in the java language

Applicant:

Oracle America, Inc.

Headword:

Compiling programming language constructs/ORACLE

Relevant legal provisions:

EPC 1973 Art. 56

Keyword:

Inventive step - (no)

Decisions cited:

T 1784/06, T 1893/08, T 1539/09, T 1370/11, T 0790/14

Catchword:



Beschwerdekammern
Boards of Appeal
Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 1105/17 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 11 October 2022

Appellant: Oracle America, Inc.
(Applicant) 500 Oracle Parkway
Redwood City, CA 94065 (US)

Representative: D Young & Co LLP
120 Holborn
London EC1N 2DY (GB)

Decision under appeal: **Decision of the Examining Division of the
European Patent Office posted on 22 November
2016 refusing European patent application No.
06255752.5 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman M. Müller
Members: S. Krischer
B. Müller

Summary of Facts and Submissions

- I. The appeal is directed against the decision of the examining division, dated 22 November 2016, to refuse application No. 06255752 for lack of inventive step over common general knowledge alone.
- II. A notice of appeal was received on 30 January 2017. The appeal fee was paid on the same day. A statement of grounds of appeal was received on 31 March 2017.
- III. In its summons to oral proceedings, the board gave reasons why the claims lacked an inventive step.
- IV. In a letter dated 9 September 2022, the appellant submitted further arguments.
- V. In a telephone call on 7 October 2022, the representative announced that no one would be attending the oral proceedings on behalf of the appellant.
- VI. Oral proceedings were held on 11 October 2022 without the appellant.
- VII. The appellant requested in writing that the decision under appeal be set aside and that a patent be granted on the basis of claims 1-12 filed with a letter of 26 September 2016 (the sole request being the subject of the decision under appeal; see the decision, section 13).

The appellant also "specifically ask[ed]" the board to contact it prior to arranging oral proceedings if the board were minded to refuse the application for reasons

other than those set out in the appealed decision (see the grounds of appeal, section 1.3).

VIII. Claim 1 reads as follows:

"1. A machine-implemented method for generating program code, the method comprising a compiler performing a compilation by:

locating (202), within source code, a particular reference that is being passed as a parameter;

in response to locating the reference, determining whether the particular reference is a reference to a programmatic method; and

in response to a determination that the particular reference is a reference to the programmatic method, automatically generating (208) compiled code that automatically would be generated by a compiler of the source code if the source code comprised, instead of the reference to the programmatic method, a reference to an object that was an instance of a class that implemented a method that invoked the programmatic method;

wherein the programmatic method is a first method, wherein locating the particular reference comprises locating the particular reference in an invocation of a second method that accepts, as a parameter, a reference to an object, and wherein the invocation passes, as a parameter to the second method, the reference to the first method."

Reasons for the Decision

1. *The appellant's request to be contacted prior to arranging the oral proceedings*

As explained in the summons, the board saw no reason to comply with the appellant's request to be contacted ahead of the arrangement of oral proceedings. The appellant was afforded sufficient time to reconsider its position in view of the board's preliminary opinion and in preparation of the oral proceedings. This was the case even more because the board, already in its preliminary opinion, substantially agreed with the reasons given in the decision under appeal.

2. *Summary of the invention*

- 2.1 The application relates to a compiler program which translates source code written in a modified variant of the Java programming language into "compiled code" (as it is called in the claims, e.g. byte code; see original description, page 6, first paragraph; figure 2: 212 and 214). The Java variant allows the programmer to enter source code in a more concise syntax (e.g. for event-handling; page 3, last paragraph), instead of having to "re-enter, again and again, new class definitions that are highly similar and essentially necessary only because of the syntactical requirements of the JAVA Language" (page 3, last but one paragraph).

- 2.2 The specific form of the new, more concise syntax (page 6, lines 2-6; figure 1: 106), defined by its equivalent in the unmodified programming language (lines 9-12; figure 1: 110, together with the

automatically generated so-called "bridge class" called MYBoxer 108), can be found in the description (pages 6-8). For the reasons given below, the details of the new Java syntax and the corresponding syntax in standard Java are not relevant for assessing the patentability of the invention.

3. *Inventiveness*

3.1 It has consistently been decided by the boards of appeal (see, e.g. T 1539/09, 4.3, cited in the appealed decision, section 26, and T 790/14, 2.3) that the design or provision of programming language constructs per se does not contribute to the solution of a technical problem and cannot therefore contribute to the presence of an inventive step.

3.2 In the present case, the alleged effect of the new programming language construct is to "allow[] a programmer to specify event-handling mechanisms in JAVA using more concise, less verbose syntax" (page 3, last paragraph; grounds of appeal, 4.4) in order "to make programmers more efficient" by "reducing the amount of code that programmers need to write" (also 4.4).

3.3 It is true that having to write "less verbose" source code may "spare" the programmer some "burden", namely the mental burden of having to conceive the more verbose syntax or the "mechanical" effort of inputting that code into a computer. The compiled code generated is - and is defined to be - the same as if the more verbose syntax had been used (see claim 1, lines 7-12). Hence, the invention has no effect on the compiled code eventually carried out.

3.4 Moreover, the mentioned advantages are only relative to a programming language with a "more verbose" syntax. Without that point of reference, the proposed programming language would have to be considered as just any programming language with its individual properties and requirements on the programmer. Providing such a programming language would not, according to the case law cited above, be considered to solve a technical problem. This conclusion cannot be called into question by making reference to a different programming language (or one of its constructs) which is somehow inferior. In the board's view, this argument is analogous to the established one that making a computer program more efficient only contributes to the solution of a technical problem if the computer program solves a technical problem independent of its efficiency (see e.g. T 1370/11, reasons 10; T 1784/06, reasons 3.1.2).

3.5 As just stated, the choice of the "less verbose" programming language cannot, for the purposes of inventive step, be distinguished from the choice of any programming language. Programmers may make this choice according to one or several of the following reasons: (1) according to subjective preferences (as the examining division correctly points out), (2) according to circumstances such as which programming language has already been chosen in a given project or (3) for which the compiler happens to be available on the available hardware, but, indeed, also (4) according to which programming language provides certain commands (as the appellant points out in its grounds of appeal, 4.2). At least the first three are non-technical reasons for the choice. Since the claim language does not exclude these, it can be left open whether consideration (4)

(contributed by the appellant) might, in certain circumstances, be acceptable as technical.

- 3.6 Notwithstanding the preceding analysis, the board considers that sparing the programmer some mental burden during programming is not, in itself, a technical problem. This is also the case because it cannot be determined objectively: programmers may differ as to which programming constructs they find simpler to understand and deal with.
- 3.7 The board considers the fact that the invention might spare the programmer keystrokes to be merely incidental to the syntactic variant in question.
- 3.8 Furthermore, the appellant argues that the adaptation of an existing compiler to the new construct is not "straightforward" (grounds of appeal, section 5.3), contrary to what is stated in the decision (section 22, last two sentences).
- 3.9 Consistently with this, the appellant stated in its letter of 9 September 2022 (section 2.5) in reply to the summons to oral proceedings that the technical effect was "to provide a method of compiling which provides improved functionality allowing for the use of a new (concise) syntax". The board takes this to mean "how to compile a new (concise) programming language construct".
- 3.10 The board notes that this effect, irrespective of whether it is technical or not, does not depend on whether the new construct is or is not concise.

- 3.11 However, the claim language does not contain any details of the compilation process. It merely states that the new construct is detected and that "compiled code" (i.e. bytecode) is generated that corresponds to the equivalent standard Java source code, i.e. to the definition of an object of the "bridge" class (figure 1: 108) and the call of its method (110). Noting that the meaning of the concise construct is defined by its expansion into the equivalent standard Java version, this does not go beyond stating that the compiler satisfies the fundamental requirement of being correct with respect to the given operational semantics of the language variant.
- 3.12 According to section 7 of the grounds of appeal, the invention has the advantage that the "new functionality" (i.e. the new construct, in the board's formulation) "is introduced without sacrificing consistency for existing virtual machines". This means that the desired compiler is intended to produce ordinary bytecode that can be executed by an ordinary Java virtual machine.
- 3.13 However, the board considers it to be obvious to compile a new language construct, the meaning of which is defined by source code expansion, with the existing standard Java output (i.e. ordinary bytecode).
- 3.14 In its letter of 9 September 2022 (section 2.6), the appellant reformulates the technical problem, allegedly by analogy with T 1893/08, section 6.4 as "provid[ing] a more flexible system for compiling a program". Actually, however, in the present case the compiler output is substantially determined by the requirement that it must support the new language construct. Any enhanced "flexibility" is a consequence of the larger

"vocabulary" in the modified programming language, i.e. the provision of the new programming language construct which the board considers to be non-technical. The decision T 1893/08 is, therefore, not pertinent to the case at hand.

3.15 Therefore, the subject-matter of claim 1 is not inventive within the meaning of Article 56 EPC 1973.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:

The Chairman:



L. Stridde

Martin Müller

Decision electronically authenticated