

Survey on exploring the adoption of Java version features and their relationship

Objective: This survey aims to capture the experiences and insights of Java developers regarding the adoption of new methods introduced in recent Java versions. Knowledge or experience with SonarQube would be greatly beneficial in contributing to this survey and enhancing the quality of the collected results.

Instructions:

The survey comprises questions related to Java methods introduced in Java 8, Java 11, and Java 17.

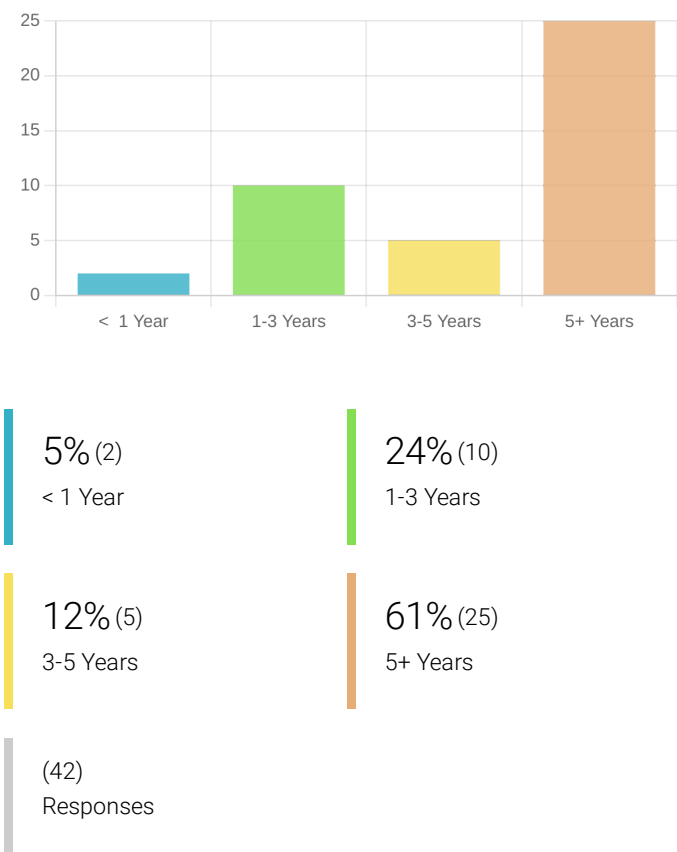
If, for instance, you primarily work with Java 8 and aren't familiar with the methods from Java 11 or Java 17, please skip the corresponding questions. Some questions offer an 'Other' option. For these, you can either leave it blank or provide a custom response as needed.

Questions marked with an asterisk (*) are obligatory, while those without it are optional. Estimated completion time: 5 minutes.

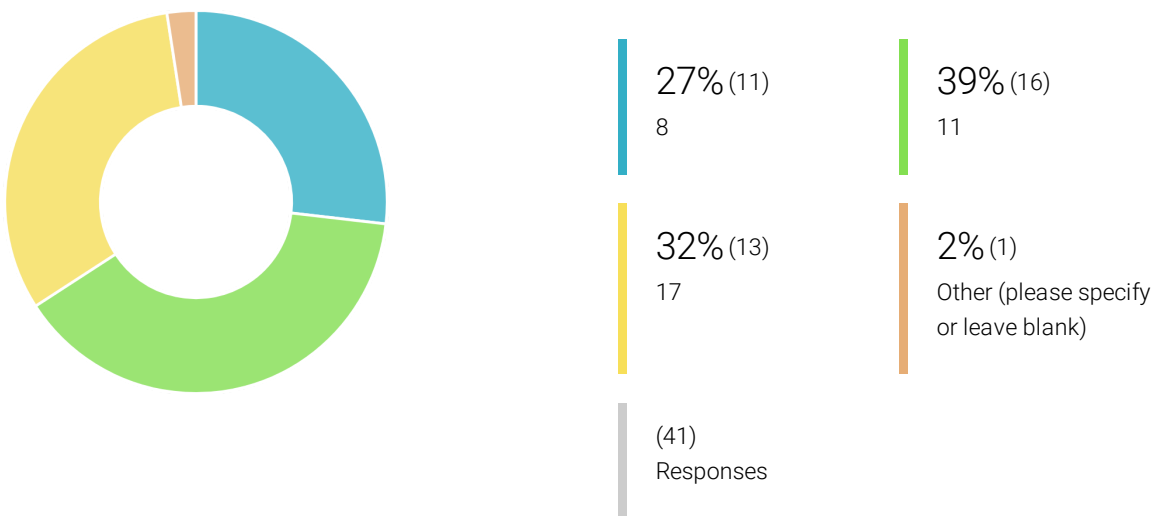
Your participation is voluntary, and all feedback will be kept anonymous. We ensure no identifiable information is stored.

By participating, you are providing valuable insights to further our understanding of Java method adoption and its potential influence on software quality.

1 How many years have you been actively programming in Java?

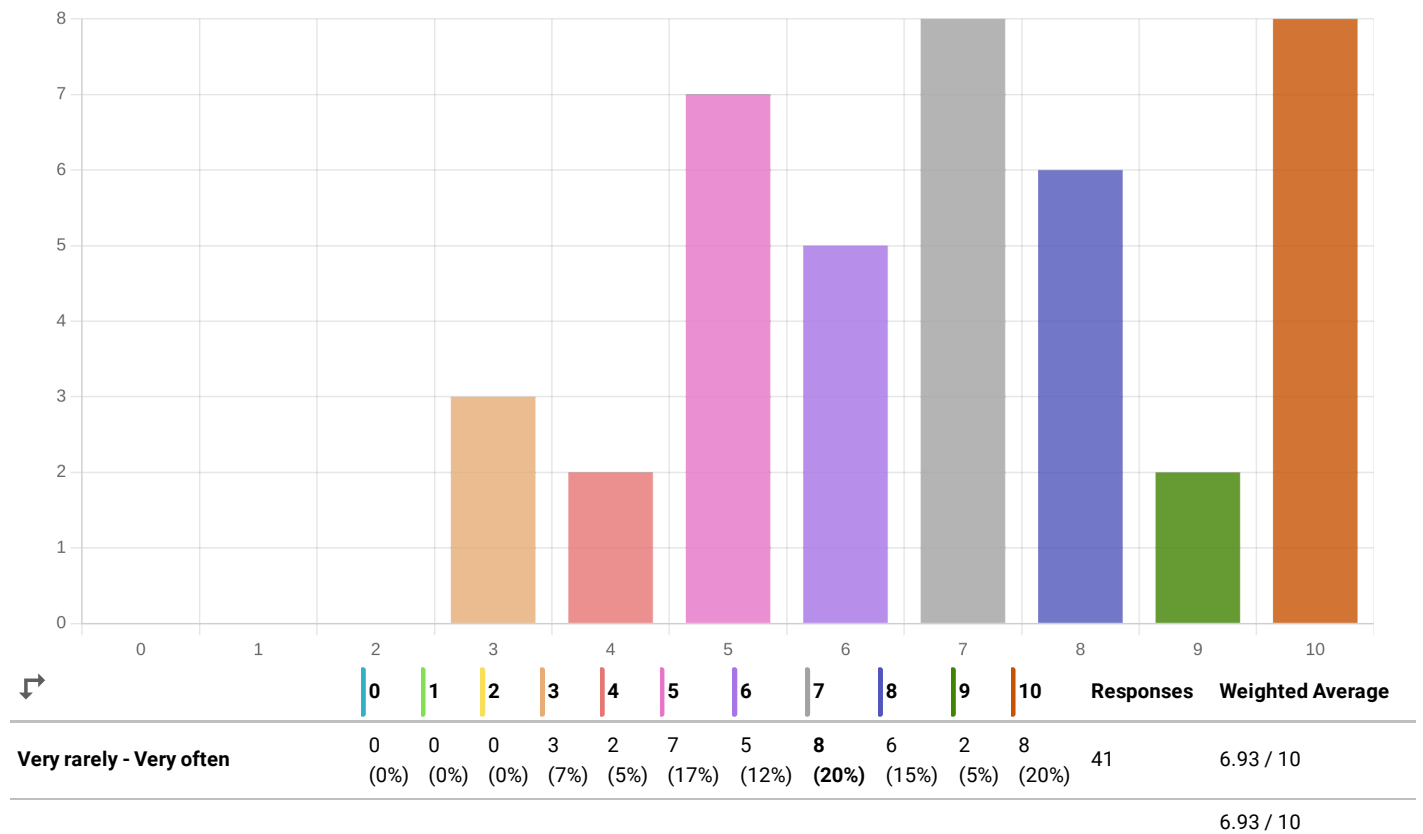


2 Which version of Java do you use the most in your projects?

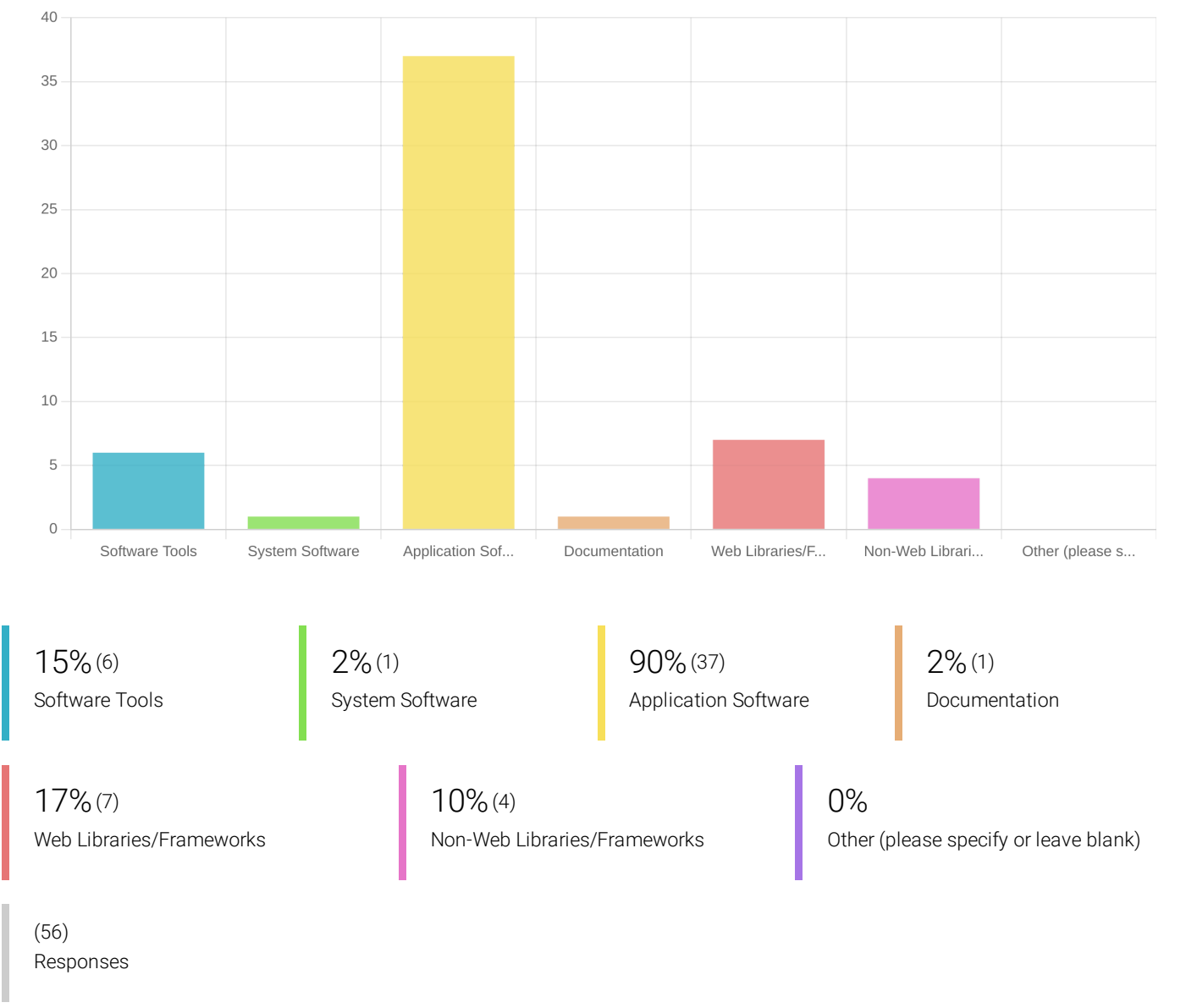


All of those, but cannot say which I used most

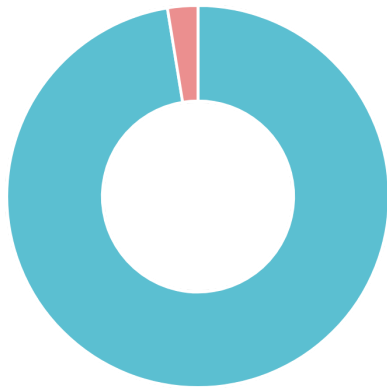
3 How often do you rely on Java built-in methods?



4 In which domain do you work the most?



5 Out of these built-in methods from Java 8, which do you use the most?



97% (38)
stream()

0%
LongStream
concat(LongStream
a, LongStream b)

0%
Calendar.Builder
setDate(int year, int
month, int
dayOfMonth)

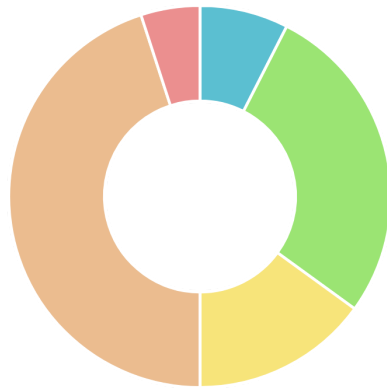
0%
Calendar.Builder
setTimeOfDay(int
hourOfDay, int
minute, int second)

3% (1)
Other (please specify or leave blank)

(39)
Responses

Have not used B,C, or D as they are very niche APIs

6 Out of these built-in methods from Java 8, which do you use the least?



8% (3)
stream()

28% (11)
LongStream
concat(LongStream
a, LongStream b)

15% (6)
Calendar.Builder
setDate(int year, int
month, int
dayOfMonth)

45% (18)
Calendar.Builder
setTimeOfDay(int
hourOfDay, int
minute, int second)

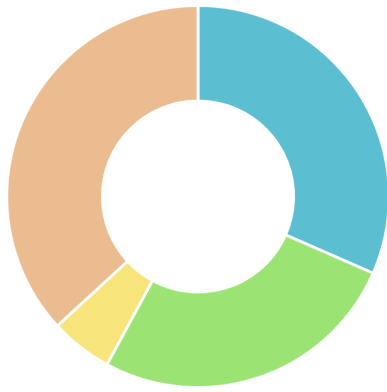
5% (2)
Other (please specify or leave blank)

(40)
Responses

all B, C, D

Have not used B,C, or D as they are very niche APIs

7 Out of these built-in functions from Java 11, which do you use the most?



32% (6)

HttpRequest.Builder
header(String name,
String value)

26% (5)

HttpRequest.Builder
setHeader(String
name, String value)

5% (1)

WebSocket.Builder
header(String name,
String value)

37% (7)

Other (please specify
or leave blank)

(19)

Responses

Files.writeString()

none

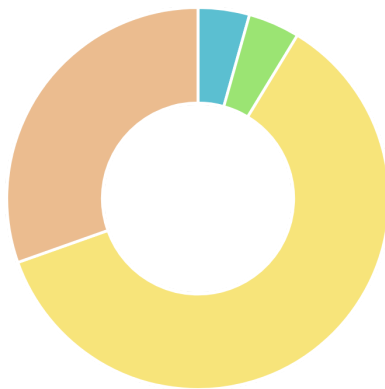
none

never used

Have not use any of them directly as they are abstracted away by most frameworks

toArray method of Collection

8 Out of these built-in functions from Java 11, which do you use the least?



4% (1)

HttpRequest.Builder
header(String name,
String value)

4% (1)

HttpRequest.Builder
setHeader(String
name, String value)

61% (14)

WebSocket.Builder
header(String name,
String value)

30% (7)

Other (please specify
or leave blank)

(23)

Responses

none

none

never used any of those

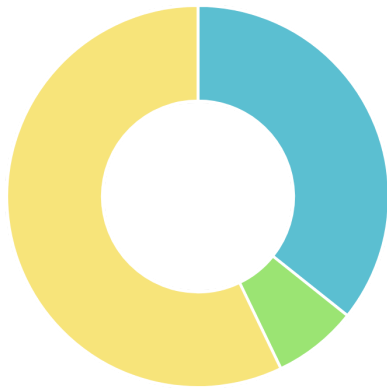
Have not use any of them directly as they are abstracted away by most frameworks

Never used any of them (JAX-RS-Client instead)

String.isBlank(String)

Haven't used any of the above

9 Out of these built-in functions from Java 17, which do you use the most?



36% (5)

ByteBuffer put(int
index, ByteBuffer src,
int offset, int length)

7% (1)

mismatch(Path
path, Path path2)

57% (8)

Other (please specify or leave blank)

(14)

Responses

Pattern matching

None really come to mind

none

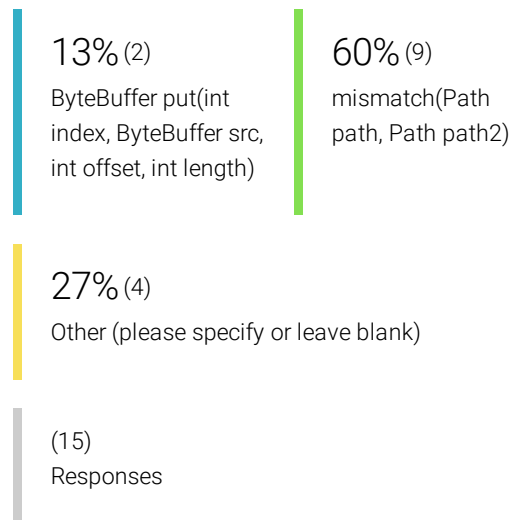
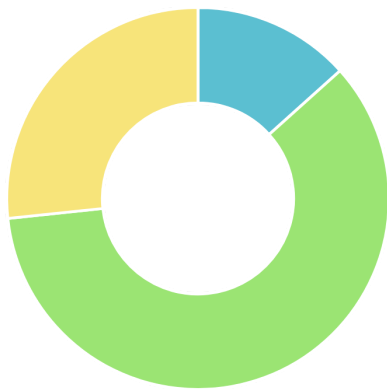
none

Records

Have not used any of them as they are very niche APIs

Haven't used any of the above

10 Out of these built-in functions from Java 17, which do you use the least?



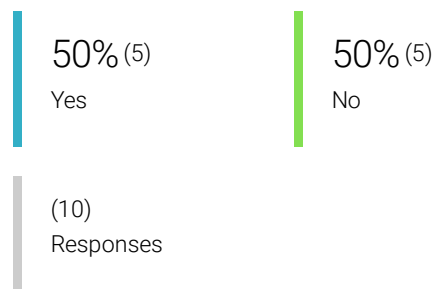
I think I really don't use any of them

none

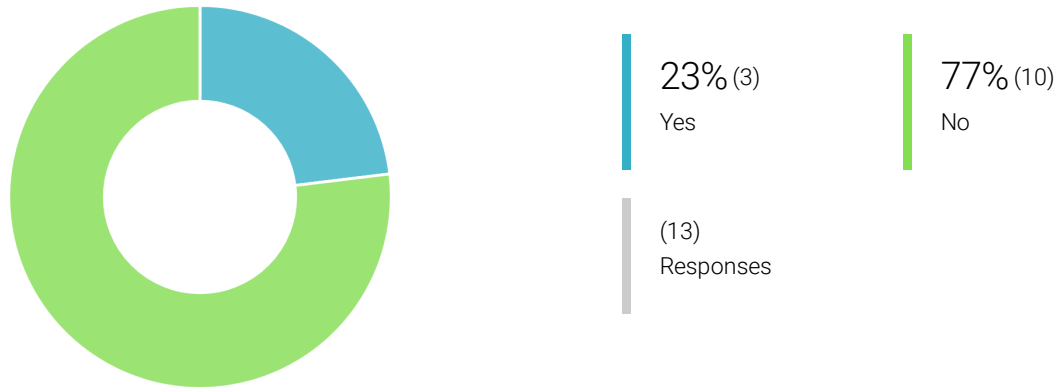
Have not used any of them as they are very niche APIs

Haven't used any of the above

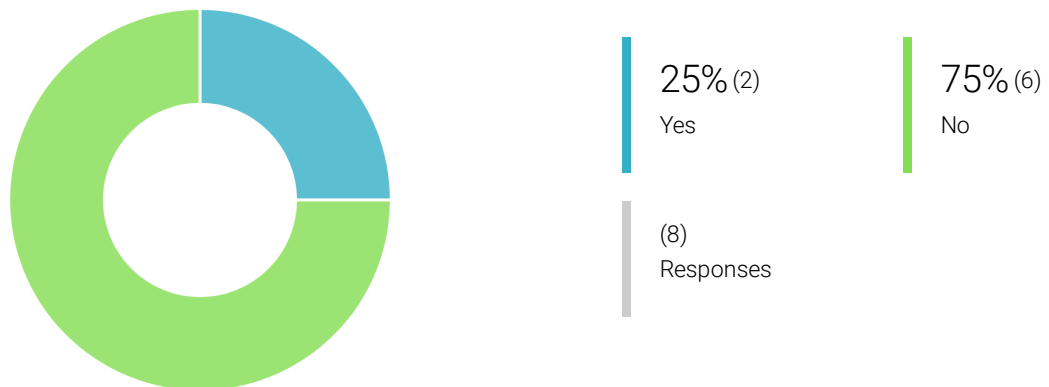
11 Upon adopting Java version 8 in your projects, did you observe an increase in Code Smells identified by SonarQube compared to previous versions of Java?



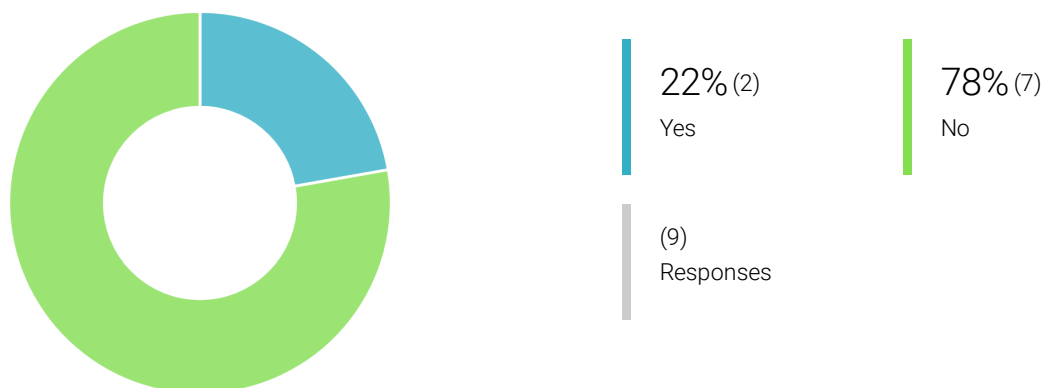
12 Upon adopting Java version 11 in your projects, did you observe an increase in Code Smells identified by SonarQube compared to previous versions of Java?



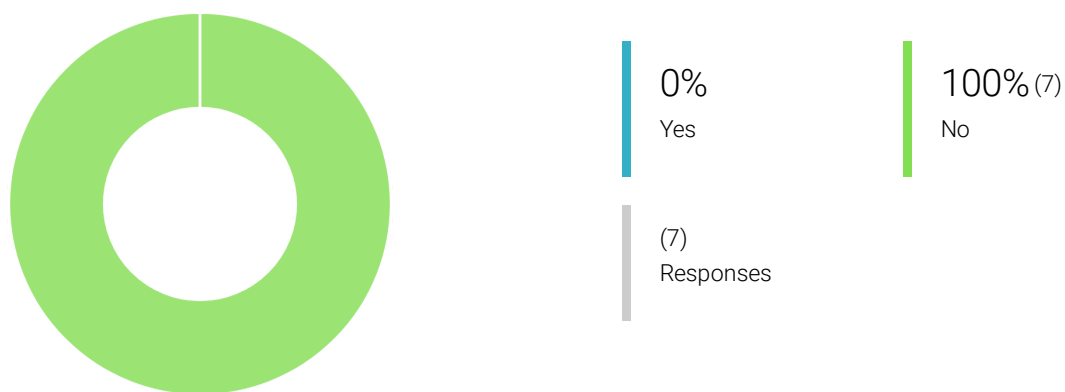
13 Upon adopting Java version 17 in your projects, did you observe an increase in Code Smells identified by SonarQube compared to previous versions of Java?



14 Upon adopting Java version 8 in your projects, did you observe an increase in Bugs identified by SonarQube compared to previous versions of Java?



15 Upon adopting Java version 11 in your projects, did you observe an increase in Bugs identified by SonarQube compared to previous versions of Java?



16 Upon adopting Java version 17 in your projects, did you observe an increase in Bugs identified by SonarQube compared to previous versions of Java?

