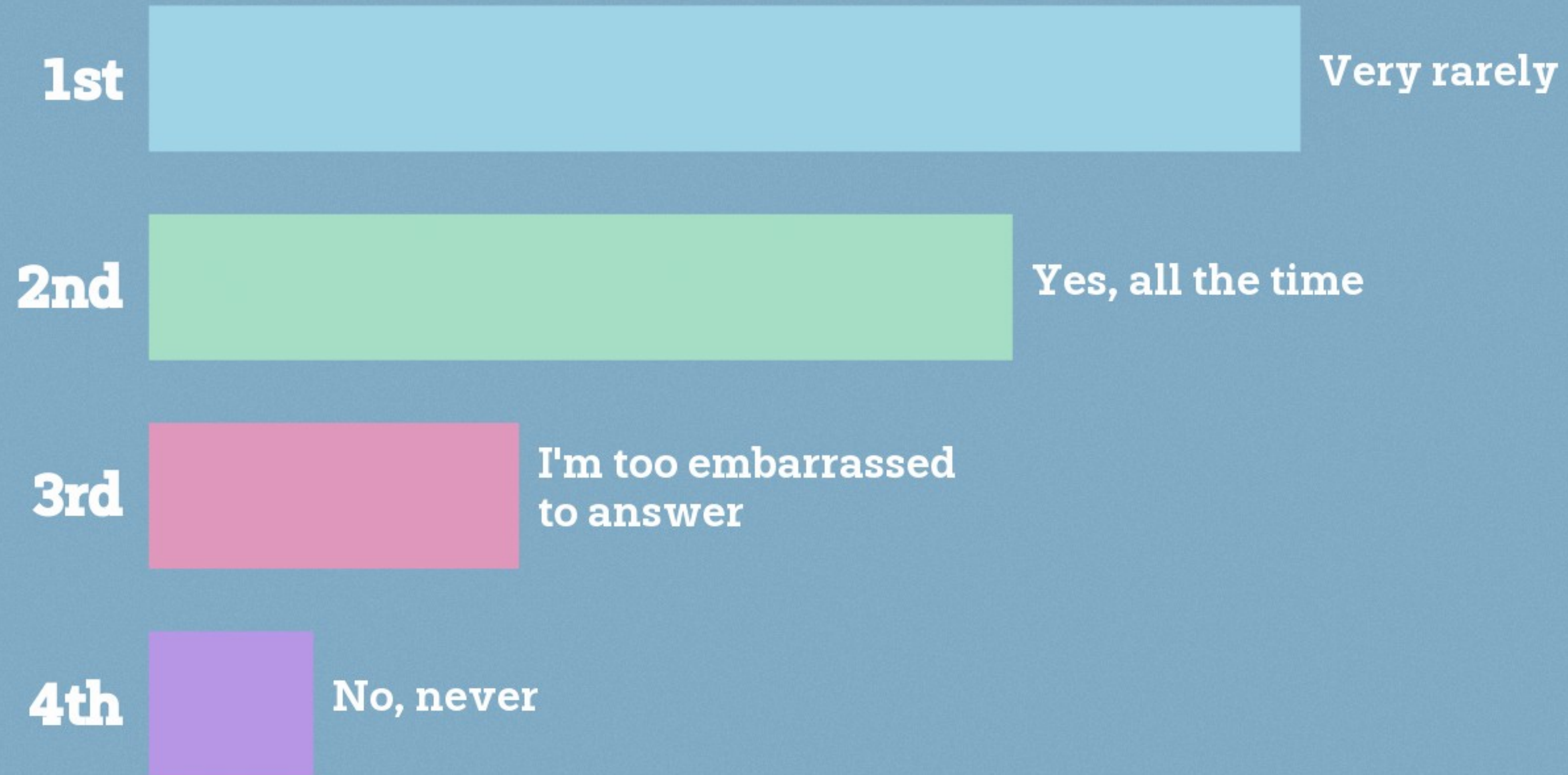


# Ask me anything

**0 questions**

**0 upvotes**

# Do you ever copy-paste code?



# How do you get rid of duplicated code that differs in just a few lines of code?

loops

Inheritance

Functions taking lambdas to do "different operations" on data

Helper Method

constants

make a Funktion with it

by creating new methods or using inheritance

make a methode for what it does

Join the duplicated code and use an if else for the few differing lines

# How do you get rid of duplicated code that differs in just a few lines of code?

Divide the Problem in Subproblems. Trying to solve the Subproblem by the help of the principles of inheritance, Interfaces etc.

Using Interfaces

so they dont have more than one use

write as a method

interface

if else statements (or a switch perhaps?)

to get better readability and avoid confusion

# Why should methods be small?

easier to understand

Readability

Overview

For managing their own responsibility

readability

easier to read

Easy to read and understand.

Easier to understand for others

Readability, maintainability, reusability

# Why should methods be small?

easier to change and test and read

comprehensibility

efficiency

distribution of responsibility

readability and easy to make changes

Because of efficiency

divide and conquer

Should they?

when a method contains static data like a lookup table

# Why should methods be small?

Specially targeted algorithms

Inline documentation on complex parts

limited memory

when a method contains static data like a lookup table

# When does it make sense for methods to be long?

If they are algorithms

If its within the responsibility of the method

If they're still doing just one thing.

Algorithm

for small and simple programmms

If there is no point in divide the Problem which is solved in the long method, such as there exist no subproblem which would make sense.

fancy algorithm's

never

not solvable in any other way, if it, necessary, If I want to annoy the Assis, If i want that it difficult to understand: dark pattern



# When does it make sense for methods to be long?

Inline documentation on complex parts

for procedural programming sometimes

when it contains static data like a lookup table

many different cases in the same hierarchy  
(switch when when , etc)

Dynamic programming

# If you ever developed code for a customer, what was the application?

website

Simulating DNA damage for very old samples.

I helped to code a website , but I bearly did anything

website, reports, data analysis

I programmed a robot but this was for a project that I did with other students

a prototype to configure experimental protocols

# Imagine you are designing a web shopping site. Would “Successful Payment” be a likely class in your design? Why or why not?

no, it would be a state

no a state

No, because it is a state of payment

not such a good idea because of security

Probably not, because it is more of a state than a class.

I think it would just be a method to a class named payment

Probably not, it would be part of a payment class

no. "success" would likely be a property of the class payment. Successful implies that there aren't any responsibilities or tasks left

It would be a state of some payment object

# Imagine you are designing a web shopping site. Would “Successful Payment” be a likely class in your design? Why or why not?

A reason for it could be if you use it in accounting...

only reason would be if you are analysing succesful payments themselves

it could be a class if payments are to be managed themselves

**Last chance for questions**