

AI for developers

*how to create responsible value*

Johan Hutting

June 2<sup>nd</sup>, 2023 - Devoxx Poland, Krakow

ING  DEVOXX™



do your thing

# ./whoami

Johan Hutting

Senior software engineer at ING

Main focus on JVM technologies, but also interested in game design and Artificial Intelligence: my chess program participated in the Dutch championships.



# Introduction



# What will you learn

- No doom & gloom
- Modern, promising A.I. technologies to apply
- The benefits you'll gain with them
- Which elements to approach with caution

An aerial photograph of a rural landscape under a blue sky with white clouds. In the foreground, there's a green field with dark, parallel tracks from agricultural machinery. To its right is a narrow, rectangular body of water, likely a canal or irrigation ditch, bordered by a strip of green vegetation. Further right is a large, yellowish-brown field that has been recently harvested, showing distinct rows. The background shows more of the same agricultural land stretching towards the horizon.

Generative pre-trained transformers

AKA: GPT-4, ChatGPT

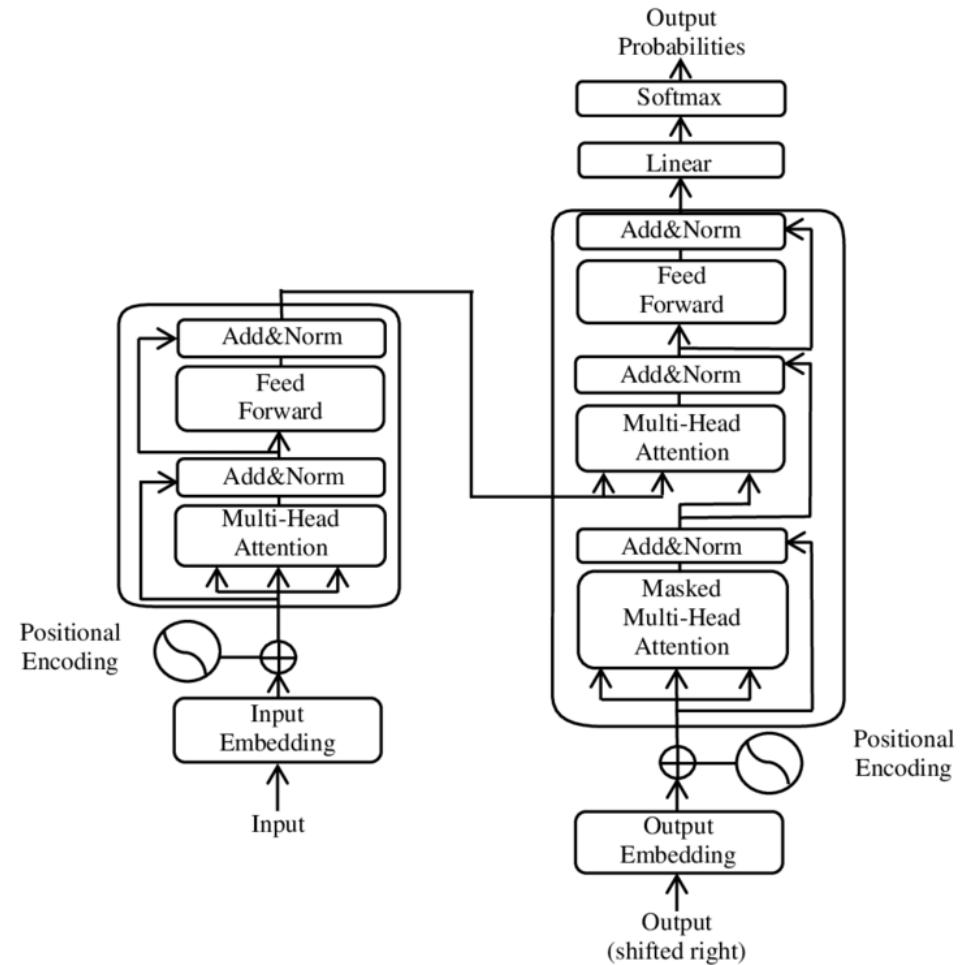
# Transformers

The transformer A.I. architecture was published by **Google Brain** in 2017.

Main benefit: parallel processing of very large datasets while maintaining context.

From 2018 onwards this has increasingly become popular for *Natural Language Processing* and *Computer vision* A.I. tasks.

Typically, these learn in two layers: both the **large language model** (LLM) and the **reinforced learning from human feedback** (RLHF).



# OpenAI's ChatGPT

The screenshot shows a dark-themed chat interface. On the left, a blue square icon contains the letters 'JO'. To its right, the text 'What are you?' is displayed. Below this, another blue square icon contains a white circular swirl logo. To its right, the text 'I am ChatGPT, a large language model developed by OpenAI. I am designed to understand' is followed by a thumbs-up emoji and a reply arrow emoji. Below this, the text 'and respond to natural language queries and generate human-like responses using state-of-the-art machine learning algorithms.' is shown.

ChatGPT can

- Supply the key points of a TLDR; text (or book)
- Explain a subject and allow you to ask for clarifications on specific parts
- Give suggestions for names or concepts
- Support with poetry and writing targeted explanations
- Act as a call centre client or assistant, tracking context of your questions and responses
- Generate plantUML diagrams for your descriptions
- Supply designs for websites
- Generate code examples for your challenges

# Feedback loop on challenges

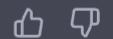


JO

Give me five examples for the coding challenge



Sure, here are five potential coding challenges for a senior Java engineer:



1. Implement a concurrent data structure: Ask the candidate to implement a concurrent data structure such as a concurrent hash map or a concurrent linked list. This challenge is designed to test the candidate's understanding of multithreading, synchronization, and concurrent programming.
2. Build a RESTful API: Ask the candidate to build a RESTful API using Java and a web framework such as Spring or JAX-RS. The API should allow users to create, read,

JO

Give me five examples that can be finished in half an hour



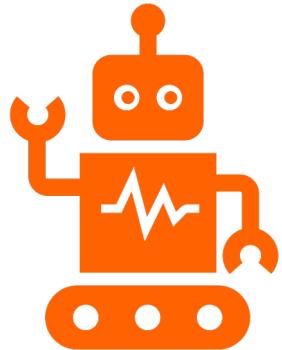
Sure, here are five potential coding challenges that can be finished in half an hour for a senior Java engineer:



1. Reverse a string: Ask the candidate to write a Java method that takes a string as input and returns the reversed string. This challenge is designed to test the candidate's understanding of basic string manipulation and Java syntax.
2. FizzBuzz: Ask the candidate to write a Java program that prints the numbers from 1 to

# ChatBots!

Who loves to talk to a company chatbot when you have a question?



Current chatbots are very restrictive as they are predesigned around a flow-diagram.

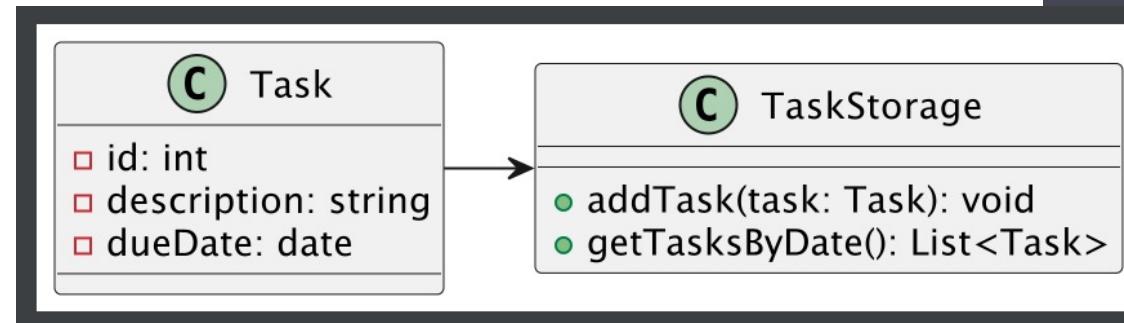


A GPT-AI with the proper domain and customer knowledge will be more effective to come to a solution.

# Generating designs

PlantUML through ChatGPT is easy to achieve by asking for the code and supplying:

- A topic
- Description of the model



JO

I want to generate a UML diagram for the following, please provide the plantUML code for this:  
topic: simple task storage  
description: user can add tasks to the database and can retrieve them, ordered by date



Sure! Here's the PlantUML code for a UML class diagram that represents the simple task storage system you described:

python

```
@startuml
class Task {
    -id: int
    -description: string
    -dueDate: date
}

class TaskStorage {
    +addTask(task: Task): void
    +getTasksByDate(): List<Task>
}

Task --> TaskStorage
@enduml
```

Copy code

In this diagram, there are two classes: `Task` and `TaskStorage`. `Task` has three private attributes (`id`, `description`, and `dueDate`), which represent the unique identifier, description, and due date of a task, respectively. `TaskStorage` has two public methods (`addTask` and `getTasksByDate`), which allow users to add tasks to the database and retrieve them ordered by date, respectively.

# Hallucinating

Returning an answer is more important than the correctness thereof:



Can you give me 10 anagrams of Krakow along with their meaning?



Certainly! Here are ten anagrams of "Krakow" along with their meanings:

1. Warlock: A male witch or sorcerer.
2. Kowtark: A rare term used in architecture to describe a small concave molding.
3. Karok: Referring to the Karok people, an indigenous Native American tribe from California.
4. Okra: A green vegetable with a slimy texture when cooked, often used in soups and stews.
5. Workka: A playful term with no specific meaning.
6. Krowka: The Polish word for "cow," referring to a female bovine animal.
7. Wokark: A nonsensical term with no specific meaning.



What is an anagram?



An anagram is a word or phrase that is formed by rearranging the letters of another word or phrase. The new word or phrase uses all the original letters exactly once. For example, an



## Evaluation

Keep in mind the results are based on a probability matrix, not on logic! You will sometimes get hallucinating results, or even gaslighting responses on longer chats.

You'll still need to correct the output quite often, despite the A.I. responding with confidence.

Do not share internal knowledge if you're not allowed to!

A photograph of a man with light brown hair and a well-groomed beard, wearing black-rimmed glasses and a grey hoodie. He is looking intently at a laptop screen. In the background, two other people are working at their desks in an office setting. One person is wearing white headphones and a blue shirt, while the other is wearing a white t-shirt. The scene is lit with warm, natural light.

A.I. code generation

*Co-pilot, CodeWhisperer, ...*

# Rubber Ducky pair-programming

Consider the A.I. as a pair programming rubber duck.

This is mostly identical to autocomplete in IntelliJ, but with more context regarding your code.

You'll still need to pay attention, quite often wrong solutions are provided. You can even end up with missing braces or other forms of code that doesn't even compile.



## Copilot



GitHub Copilot originally launched in 2021 as an intelligent code completion companion.

The main features:

- Supports multiple languages
- Auto-complete suggestions for lines of code as well as complete methods
- Generate code based on comments

On March 16<sup>th</sup> Microsoft 365 Copilot was announced offering identical solutions for the Office suite, such as Outlook, Word and PowerPoint. During Build '23 it was announced they'll also include it in Windows.

### Concerns

The quality is so-so, you still must correct code quite often.

Keep in mind that you share your local code with GitHub when you don't use the enterprise version of Copilot.

There are lawsuits regarding the use of open-sourced code for training.

## Codewhisperer

Amazon's answer to Microsoft's GitHub Copilot: trained on both open source and curated internal Amazon code.

Free for now, likely licensed in the future.

Quality seems better than Copilot on Java and Kotlin code (fewer mistakes).

The aforementioned concerns with Copilot also apply here.

# Evaluation

Promising support technology, but:

- Are you allowed to share domain knowledge outside your organisation?
- Are you allowed to share code outside your organisation?
- Can you remain focussed on auto-generated boring code?
- How will the juniors learn and improve their craft?

Regarding the use during interviews: Also keep in mind that you don't fully block the usage of these on intake tests if you are ok with people using them.



Generating images

*Your computer as an artist?*

# Stable diffusion

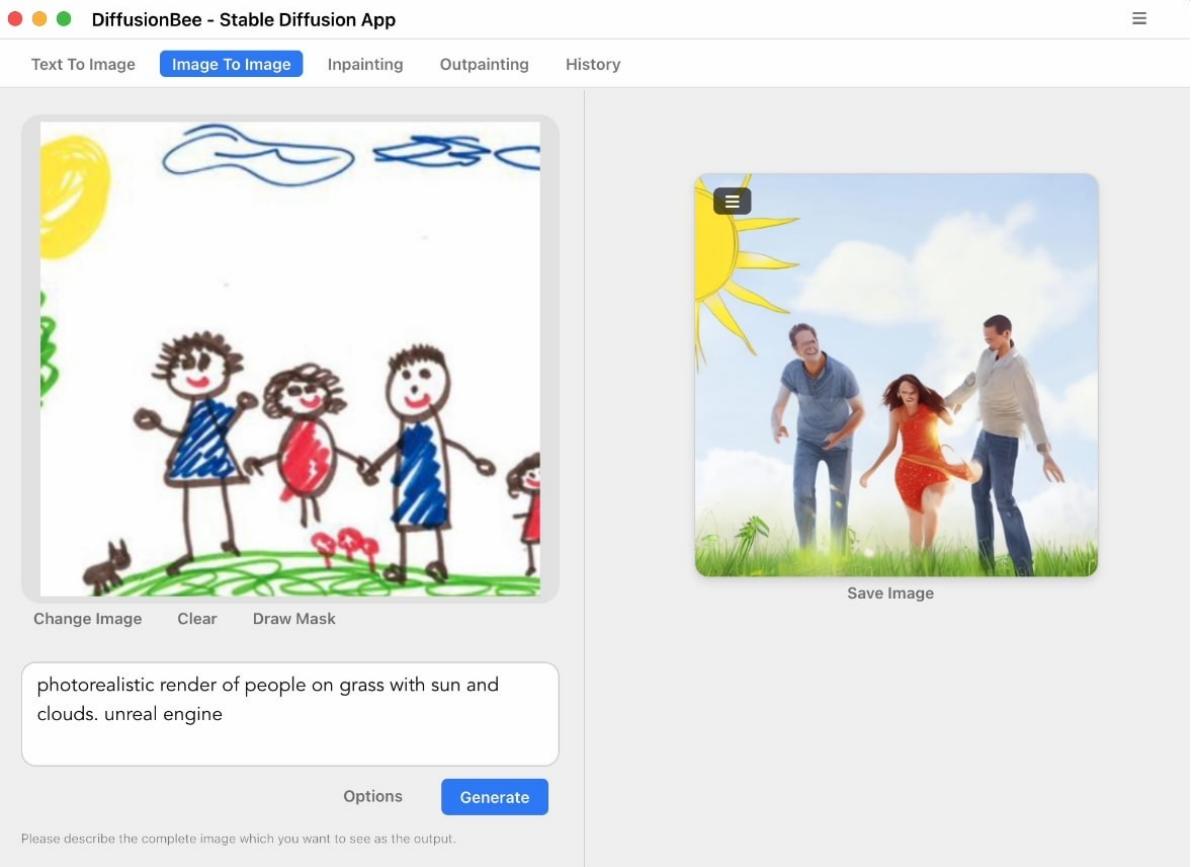
Open sourced A.I. image generation on your local hardware.

It runs as a python script or with a GUI such as Diffusion Bee.

Stable Diffusion also allows you to apply and extend your own trained set, for instance with your own art or stock photo's (assuming you have permission)

If you don't have fast enough hardware, it is also available online:

<https://huggingface.co/spaces/stabilityai/stable-diffusion>





## Dall-E / MidJourney

Closed-source image generation in the cloud.

DALL-E by OpenAI has been available since 2021 and uses GPT-3 under the hood. It is trained with images found on the internet. Initial use is free, paid per month after you reach a threshold

Midjourney creates more “artsy” output. It is closed source but seems to use Stable Diffusion with specific art in its training set. MidJourney v5 which was recently released also delivers photorealistic results and generated the now famous Pope and Trump arrest photographs.

You were able to generate 25 pieces of art for free with prompts on their Discord, this was discontinued after “non-paying users were mishandling the technology”.

AI generated  
(by MidJourney v5)

# Evaluation

Copyright and usage rights is the main issue. You risk expensive court cases when you use someone else's art or photographs without permission.

Can we use this for developers?!  
To design logos and user interfaces or  
should we stick to ChatGPT?

The screenshot shows a ChatGPT conversation. The user asks, "Can you generate a view to add tasks?" ChatGPT responds with a simple example of a view for adding tasks, presented as markdown code:

```
markdown
-----
|       Add New Task      |
|-----|
| Task Name:              |
| [ ]                      |
| Description:             |
| [ ]                      |
| Due Date:                |
| [ ]                      |
| Priority:                |
| ( ) Low                  |
| ( ) Medium                |
| ( ) High                  |
| [Add Task]               |
-----
```

At the top right of the code block, there is a "Copy code" button. The interface has a dark theme with light-colored text and icons.



Pitfalls



*Bias and other unwanted side-effects*

# Bias

As the GPT models are scored by probability bias will be introduced by the dataset. People are biased too, but unlike an algorithm they can overcome this by themselves.

Applying nuance and rational logic still requires a ‘human touch’.

Correcting the dataset (LLM) or applying reinforced learning (RLHF) can potentially create new bias as well as censorship complaints afterwards. Before you extend a set and release it, test it well!

## Other unwanted side-effects

Stack overflow banned A.I. submissions as it is considered harmful.

Education and academics are running into challenges.

The uncanny valley: is the contents of the mail you received real or generated?

Responsibilities of generated text and code: **you** are the owner.

Copyrighting generated art is (at the moment) not possible. Several lawsuits have been started regarding the input of the models, both text and image-based.

# Takeaways



# Takeaways



Be vigilant



Experiment



Create value

Questions?





**do your thing**

<https://ing.jobs>