

КУРСОВА РАБОТА

Тема

Java приложение с графичен интерфейс
за взаимодействие с ChatGPT API

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО
КАТЕДРА „КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ“

Дата на задаване: 08.10.2025

Краен срок за предаване и защита:

Преди изпита

Оценка:.....

Преподавател.....(подпис)

Дата:.....

З А Д А Н И Е

за курсова работа по дисциплината: Програмиране в Интернет среда

1. Тема: Java приложение с графичен интерфейс за взаимодействие с ChatGPT API.

2. Изисквания по съдържанието:

2.1. Теоретична част:

- Програмен достъп до ChatGPT чрез API. Описание на структурата на заявките.
- Реализация на Java приложения с графичен интерфейс чрез Swing.
- Описание на програмния код.
- Експерименти. Изводи.
- Използвана литература.

.....

2.2. Практическо изпълнение:

- Програмен код

2.3. Обем на чертежите:

3. Задължителни консултации:

Студент: Ваня Тодорова Ванева
(име, презиме, фамилия)

Преподавател:
(фамилия и подпис)

курс: 2, група: 1, фак. №: 22472128

доц. Р. Иванов /

1. Теоретична част

1.1. Програмен достъп до ChatGPT чрез API. Описание на структурата на заявките

Програмният достъп до ChatGPT става чрез OpenAI API, който позволява на разработчици и приложения да изпращат заявки към модела и да получават автоматично генерирани отговори.

API използва протокола HTTPS и JSON за обмен на данни между клиента и сървъра. За достъп до услугата се изисква API key, който контролира достъпа до ресурсите

Една заявка към ChatGPT API включва тези елементи:

model	Посочва кой модел ще се използва.
messages	Списък от съобщения. Всяко съобщение съдържа роля ("system", "user", "assistant") и текст ("content").
temperature	Креативността/случайността на отговора. От 0-2.
max_tokens	Максималния брой думи/символи, генерирани в отговора.

Таблица 1

Обикновено се изпраща POST заявка към endpoint. В този случай към <https://api.openai.com/v1/chat/completions>

Примерен JSON формат на заявката

```
{
  "model": "gpt-4o",
  "messages": [
    {"role": "system", "content": "Ти си помощен асистент."},
    {"role": "user", "content": "Обясни как се използва ChatGPT API в Java."}
  ],
  "temperature": 0.7,
  "max_tokens": 200
}
```

Отговорът, който API връща, в JSON формат

```
{
  "id": "chatcmpl-12345",
  "object": "chat.completion",
  "created": 1713456789,
  "model": "gpt-4o",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "За да използвате ChatGPT API в Java, трябва да изпратите HTTP POST заявка..."
      },
      "finish_reason": "stop"
    }
  ]
}
```

1.2. Реализация на Java приложения с графичен интерфейс чрез Swing

Swing е framework (платформа) за изграждане на графичен потребителски интерфейс (GUI - Graphical User Interface) в езика Java, базирана на архитектурния модел Model-View-Controller (MVC), който разделя логическата структура на кода от графичното представяне на интерфейса.

ChatGPTGui.java (View и Controller) - отговорен за всички аспекти на потребителския интерфейс и за управлението на потребителските действия.

ChatGPTApiClient.java (Model) - независим от Swing и е отговорен за комуникацията с външния ChatGPT API.

1.3. Описание на програмния код

1.3.1. Описание на приложението

Проектът е базиран на Maven, който е инструмент за изграждане на Java проекти, използвайки Project Object Model (POM) управлява компилацията, тестването и документирането на проекта.

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО
КАТЕДРА „КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ“

ChatGPTGui.java взима потребителските действия, като натискане на бутони, визуализира елементите (прозорци, полета за текст, менюта), предава данните на модела и изпълнява API повикването в отделна нишка, за да не блокира интерфейса.

ChatGPTApiClient.java съдържа бизнес логиката, напълно независим от Swing. Създава и изпраща HTTP POST заявки към API-то и конструира JSON заявката с въпроса на потребителя, обработва и връща чист текстов отговор.

pom.xml дефинира зависимости (<dependencies>), които декларират външните библиотеки org.json и junit и плъгини (<plugins>), конфигурира ехес-maven-plugin.

.gitignore е файл за контрол. Казва на Git да игнорира автоматично генерирани файлове и директории, като target/ и config.properties, за да се скрие API ключът.

1.3.2. Използвани технологии и библиотеки

- **Java Swing:** Използвано за създаване на всички елементи на ChatGPTGui.java
- **Java Threading (java.lang.Thread):** За асинхронно изпълнение на API повикването, за да се предотврати "freeze" на GUI.
- **Junit:** За писане на модулни тестове за бизнес логиката.
- **org.json.(JSONObject, JSONArray):** Използвано за JSON заявките и отговорите.
- **java.net.(URLConnection, URL):** За осъществяване на мрежовата комуникация.
- **java.util.Properties:** Използвано за зареждане на конфигурационния файл (config.properties).
- **javax.swing.*(вкл. JOptionPane, DefaultCaret):** Основен пакет за всички GUI компоненти, включително pop-up за грешки.
- **java.io.*(вкл. File, InputStream, BufferedReader):** за работа с файлове (конфигурация) и четене на HTTP отговори (streams).
- **java.awt.event.ActionEvent:** Част от модела за обработка на събитията (Event Handling Model) за улавяне на натискане на бутона.

1.4. Експерименти. Изводи

- **Грешен API key** – връща “Incorrect API key provided”
- **Недостатъчно quota/funds в профила** – “You exceeded your current quota check plan for billing details”
- **Грешен URL** – “API error (404)”
- **Висока температура** – връща много „креативен“ отговор, с много думи на различни езици
- **Без интернет връзка** – „No internet connection“
- **Без въпрос** – Излиза pop-up с „Please enter a prompt“
- **Липсва ключ в config.properties** – Излиза pop-up с „API key missing in config.properties“
- **Липсва файлът config.properties** – Излиза pop-up за липсващ config и още един с „ config.properties not found in working directory classpath“

Експеримент с „Коя година е?“

- **gpt-3.5-turbo** връща 2022
- **gpt-4o-mini** и **gpt-4o** връщат 2023

(<https://github.com/vaniavaneva/ChatGPT-API/tree/main/screenshots>)

1.5. Използвана литература

Apache.org. Apache Maven [Линк]

Достъпно от: <https://www.apache.org/>

[Дата на достъп: ??.10.2025]

Bro code. Learn Java threading in 10 minutes! [Видео] 2024

Достъпно от: <https://www.youtube.com/watch?v=SztE5W41on4>

[Дата на достъп: ??.10.2025]

CS CORNER Sunita Rai. First Java Swing GUI Application [Видео] 2022

Достъпно от: <https://youtu.be/vuQdLKq2LaY>

[Дата на достъп: ??.10.2025]

Edureka. Java Swing [Линк]

Достъпен от: <https://www.edureka.co/blog/java-swing/>

[Дата на достъп: ??.10.2025]

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО
КАТЕДРА „КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ“

Gantoin. Пример за проста Java Swing програма [Онлайн код]

Достъпен от: <https://gist.github.com/gantoin/190684c344bb70e5c5f9f2339c7be6ed>

[Дата на достъп: ??.10.2025]

GeeksforGeeks. Introduction to Java Swing [Линк]

Достъпен от: <https://www.geeksforgeeks.org/java/introduction-to-java-swing/>

[Дата на достъп: ??.10.2025]

GeeksforGeeks. Java.net.httpURLConnection [Линк]

Достъпен от: <https://www.geeksforgeeks.org/java/java-net-httpurlconnection-class-java/> [Дата на достъп: ??.10.2025]

Learn.Microsoft. Java.lang.thread [Линк]

Достъпен от: <https://learn.microsoft.com/en-us/dotnet/api/java.lang.thread>

[Дата на достъп: ??.10.2025]

Wikipedia. Swing (Java) [Линк]

Достъпен от: [https://en.wikipedia.org/wiki/Swing_\(Java\)](https://en.wikipedia.org/wiki/Swing_(Java))

[Дата на достъп: ??.11.2025]

2. Практическо изпълнение

2.1. Линк към GitHub

<https://github.com/vaniavaneva/ChatGPT-API>

2.2. Програмен код

```
config.properties
```

```
api.key=...  
base.url=https://api.openai.com/v1/chat/completions  
default.model=gpt-4o-mini  
default.temp=0.7  
default.tokens=512
```

```
ChatGPTApiClient.java
```

```
private final String apiKey;  
private final Properties config = new Properties();  
public ChatGPTApiClient(String apiKey) {  
    this.apiKey = apiKey;  
    try (InputStream input = getClass().getClassLoader()  
        .getResourceAsStream("config.properties")) {  
        if (input != null) {
```

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО
КАТЕДРА „КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ“

```
        config.load(new InputStreamReader(input,
StandardCharsets.UTF_8));
    }
    } catch (IOException e) {
        System.err.println("Warning: could not load
config.properties: " + e.getMessage());
    }
}
public String sendMessage(String model, String prompt, String url,
double temperature, int maxTokens)
throws IOException {
    if (model == null || model.isEmpty())
        model = config.getProperty("default.model", "gpt-4o-mini");
    if (url == null || url.isEmpty())
        url =
config.getProperty("base.url", "https://api.openai.com/v1/chat/complet
ions");
    JSONObject body = new JSONObject();
    JSONArray messages = new JSONArray()
.put(new JSONObject().put("role", "user").put("content", prompt));
    body.put("model", model);
    body.put("messages", messages);
    body.put("temperature", temperature);
    body.put("max_tokens", maxTokens);
    try {
        HttpURLConnection con = (HttpURLConnection) new
URL(url).openConnection();
        con.setRequestMethod("POST");
        con.setRequestProperty("Content-Type", "application/json");
        con.setRequestProperty("Authorization", "Bearer " + apiKey);
        con.setDoOutput(true);
        try (OutputStream os = con.getOutputStream()) {
            os.write(body.toString().getBytes(StandardCharsets.UTF_8));
        }
        int status = con.getResponseCode();
        InputStream inputStream = (status == 200) ?
con.getInputStream() : con.getErrorStream();
        StringBuilder response = new StringBuilder();
        try (BufferedReader reader = new BufferedReader(new
InputStreamReader(inputStream, StandardCharsets.UTF_8))) {
            String line;
            while ((line = reader.readLine()) != null) {
                response.append(line);
            }
        }
        String responseText = response.toString();
        if (status == 200) {
            JSONObject json = new JSONObject(responseText);
            JSONArray choices = json.optJSONArray("choices");
        }
    }
}
```


ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО
КАТЕДРА „КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ“

```
        if (choices == null || choices.isEmpty()) {
            return "(No response from API)";
        }
        JSONObject message =
choices.getJSONObject(0).getJSONObject("message");
        return message.optString("content", "(Empty
message)").trim();
    } else {
        try {
            JSONObject errJson = new JSONObject(responseText);
            JSONObject errorObj = errJson.optJSONObject("error");
            if (errorObj != null && errorObj.has("message")) {
                return errorObj.getString("message");
            }
        } catch (Exception ignored) {}
        return "API error (" + status + ")";
    }
} catch (UnknownHostException e) {
    return "No internet connection.";
} catch (IOException e) {
    if (e.getMessage() != null && (e.getMessage().contains("timed
out") || e.getMessage().contains("refused")))) {
        return "Network error: Unable to reach the API server ("
+ e.getMessage() + ")";
    }
    throw e;
}
}
```

ChatGPTGui.java

```
private JPanel panel1;
private JPasswordField apiKeyField;
private JTextField urlField;
private JTextArea promptText;
private JSpinner tokensSpinner;
private JSpinner tempSpinner;
private JButton sendButton;
private JButton clearButton;
private JLabel apiKeyLabel;
private JLabel modelLabel;
private JLabel urlLabel;
private JLabel ChatGPTLabel;
private JLabel tempLabel;
private JLabel tokLabel;
private JLabel promptLabel;
private JLabel answerLabel;
private JLabel nameFnLabel;
```

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО
КАТЕДРА „КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ“

```
private JComboBox<String> modelComboBox;
private JTextPane responsePane;
private final Properties config = new Properties();
private ChatGPTApiClient apiClient;
public ChatGPTGui() {
    loadConfig();
    initUI();
    initApiClient();
    attachHandlers();
}
public JPanel getRootPanel() {
    return panel1;
}
private void loadConfig() {
    File cfg = new File("config.properties");
    try (InputStream is = cfg.exists() ? new FileInputStream(cfg)
        : getClass().getClassLoader()
        .getResourceAsStream("config.properties")) {
        if (is != null) {
            config.load(new InputStreamReader(is,
                StandardCharsets.UTF_8));
        } else {
            showWarning("config.properties not found in working
                directory or classpath.");
        }
    } catch (IOException e) {
        showWarning("Error loading config.properties: " +
            e.getMessage());
    }
}
private void initUI() {
    apiKeyField.setText(config.getProperty("api.key", ""));
    apiKeyField.setEditable(false);
    apiKeyField.setEchoChar('•');
    urlField.setText(config.getProperty("base.url",
        "https://api.openai.com/v1/chat/completions"));
    modelComboBox.setModel(new DefaultComboBoxModel<>(
        new String[]{"gpt-4o-mini", "gpt-4o", "gpt-3.5-turbo"}));
    modelComboBox.setSelectedItem(config.getProperty("default.model",
        "gpt-4o-mini"));
    double temp = parseDouble(config.getProperty("default.temp",
        "0.7"), 0.7);
    int tokens = parseInt(config.getProperty("default.tokens",
        "512"), 512);
    tempSpinner.setModel(new SpinnerNumberModel(temp, 0.0, 2.0,
        0.1));
    tokensSpinner.setModel(new SpinnerNumberModel(tokens, 1, 4096,
        1));
    responsePane.setEditable(false);
}
```

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО
КАТЕДРА „КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ“

```
        try {
            DefaultCaret dc = (DefaultCaret) responsePane.getCaret();
            dc.setUpdatePolicy(DefaultCaret.ALWAYS_UPDATE);
        } catch (Exception ignored) {}
    }
    private void initApiClient() {
        String key = new String(apiKeyField.getPassword());
        if (key.isEmpty()) {
            showWarning("API key missing in config.properties");
            apiClient = null;
            return;
        }
        try {
            apiClient = new ChatGPTApiClient(key);
        } catch (Exception e) {
            showError("Failed to initialize API client: " +
e.getMessage());
            apiClient = null;
        }
    }
    private void attachHandlers() {
        sendButton.addActionListener(e -> sendMessage());
        clearButton.addActionListener(e -> {
            responsePane.setText("");
            promptText.setText("");
        });
        promptText.getInputMap().put(KeyStroke.getKeyStroke("ctrl
ENTER"), "send");
        promptText.getActionMap().put("send", new AbstractAction() {
            @Override
            public void actionPerformed(ActionEvent e) {
                sendMessage();
            }
        });
    }
    private void sendMessage() {
        if (apiClient == null) {
            initApiClient();
            if (apiClient == null) return;
        }
        String prompt = promptText.getText().trim();
        if (prompt.isEmpty()) {
            showWarning("Please enter a prompt.");
            return;
        }
        String model = modelComboBox.getSelectedItem().toString();
        String url = urlField.getText().trim();
        int tokens = (Integer) tokensSpinner.getValue();
        double temp = (Double) tempSpinner.getValue();
    }
```

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО
КАТЕДРА „КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ“

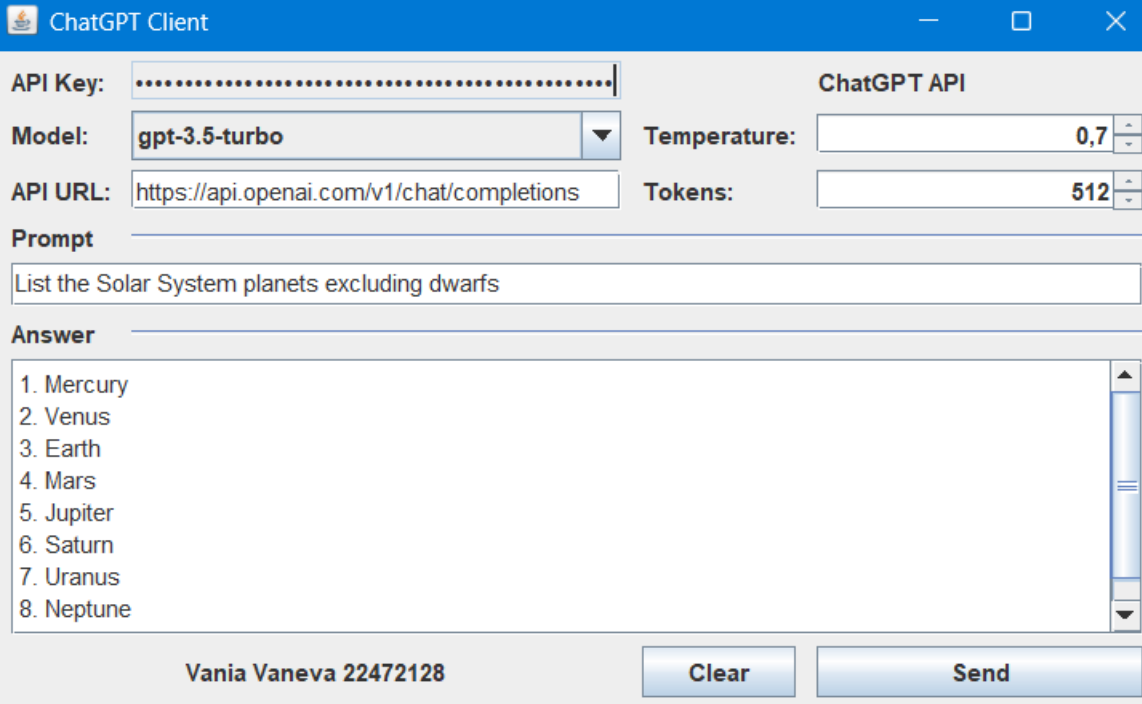
```
        sendButton.setEnabled(false);
        new Thread(() -> {
            try {
                String reply = apiClient.sendMessage(model, prompt, url,
temp, tokens);
                setResponseText(reply + "\n");
            } catch (Exception ex) {
                setResponseText(ex.getMessage() + "\n");
            } finally {
                SwingUtilities.invokeLater(() ->
sendButton.setEnabled(true));
            }
        }).start();
    }
    private void setResponseText(String text) {
        SwingUtilities.invokeLater(() -> responsePane.setText(text));
    }
    private void showWarning(String msg) {
        SwingUtilities.invokeLater(() ->
            JOptionPane.showMessageDialog(null, msg, "Warning",
JOptionPane.WARNING_MESSAGE));
    }
    private void showError(String msg) {
        SwingUtilities.invokeLater(() ->
            JOptionPane.showMessageDialog(null, msg, "Error",
JOptionPane.ERROR_MESSAGE));
    }
    private int parseInt(String s, int def) {
        try { return Integer.parseInt(s.trim()); } catch (Exception e) {
return def; }
    }
    private double parseDouble(String s, double def) {
        try { return Double.parseDouble(s.trim()); } catch (Exception e)
{ return def; }
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            ChatGPTGui gui = new ChatGPTGui();
            JFrame frame = new JFrame("ChatGPT Client");
            frame.setContentPane(gui.getRootPanel());
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.pack();
            frame.setSize(600, 350);
            frame.setLocationRelativeTo(null);
            frame.setVisible(true);
        });
    }
}
```

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО
КАТЕДРА „КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ“

pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>mavenProject</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>mavenProject</name>
  <url>http://maven.apache.org</url>
  <properties>
    <project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.json</groupId>
      <artifactId>json</artifactId>
      <version>20240303</version>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>exec-maven-plugin</artifactId>
        <version>3.6.1</version>
        <configuration>
          <mainClass>org.example.ChatGPTApiClient</mainClass>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО
КАТЕДРА „КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ“



The screenshot shows the ChatGPT Client application window. The title bar is blue with the text "ChatGPT Client". The interface includes fields for "API Key" (masked with dots), "Model" (set to "gpt-3.5-turbo"), "API URL" (set to "https://api.openai.com/v1/chat/completions"), "Temperature" (set to 0.7), and "Tokens" (set to 512). The "Prompt" field contains the text "List the Solar System planets excluding dwarfs". The "Answer" field displays a numbered list of planets: 1. Mercury, 2. Venus, 3. Earth, 4. Mars, 5. Jupiter, 6. Saturn, 7. Uranus, 8. Neptune. At the bottom, the user name "Vania Vaneva 22472128" is shown, along with "Clear" and "Send" buttons.

ChatGPT Client

API Key: ChatGPT API

Model: gpt-3.5-turbo Temperature: 0.7

API URL: https://api.openai.com/v1/chat/completions Tokens: 512

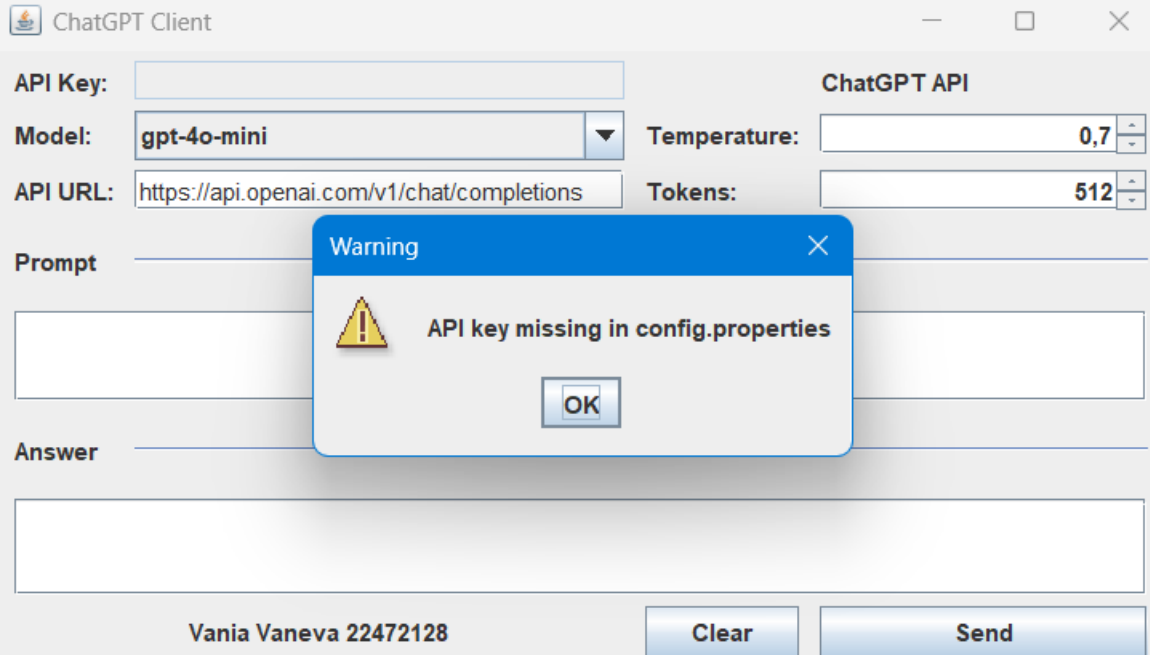
Prompt

List the Solar System planets excluding dwarfs

Answer

1. Mercury
2. Venus
3. Earth
4. Mars
5. Jupiter
6. Saturn
7. Uranus
8. Neptune

Vania Vaneva 22472128 Clear Send



The screenshot shows the ChatGPT Client application window with a configuration error. The "API Key" field is empty. A blue warning dialog box is overlaid on the window, displaying a yellow warning icon and the text "Warning API key missing in config.properties". The dialog has an "OK" button. In the background, the "Model" is set to "gpt-4o-mini", "API URL" is "https://api.openai.com/v1/chat/completions", "Temperature" is 0.7, and "Tokens" is 512. The "Prompt" and "Answer" fields are empty. The user name "Vania Vaneva 22472128" and "Clear" and "Send" buttons are visible at the bottom.

ChatGPT Client

API Key: ChatGPT API

Model: gpt-4o-mini Temperature: 0.7

API URL: https://api.openai.com/v1/chat/completions Tokens: 512

Prompt

Warning

API key missing in config.properties

OK

Answer

Vania Vaneva 22472128 Clear Send