

**Střední Průmyslová Škola Elektrotechnická Ječná**

**IT**

Ječná 30, Praha 2, 120 00

Yechnak

(Discord bot)

**Vojtěch Král**

IT

2022/2023

## Obsah

1 .....	Cíl práce	
.....		3
2 .....	Software	
.....		3
3 .....	Program	
.....		3
3.1 .....	Main	
.....		3
3.2 .....	Commands	
.....		3
3.3 .....	Moderation	
.....		6
3.4 .....	Pom.xml	
.....		7
4 .....	Závěr	
.....		8
5 .....	Zdroje	
.....		8

# 1 Cíl práce

Vytvořit automatický program (bot) na sociální platformě Discord, se základními funkcemi pro údržbu Discord serverů, školní účely a zábavu. Do budoucna bych ho chtěl zlepšit, a rozšířit.

## 2 Software

Pro vytvoření bota jsem používal IDE - IntelliJ IDEA. Pro testování a aplikaci bota jsem použil sociální síť Discord. Použil jsem knihovnu [JDA 4.4.0 352 API](#) a nástroj [Maven](#).

## 3 Program

Program se zatím skládá ze tří tříd: Main, Commands a Moderation.

### 3.1 Main

```
package DiscordBot;

import ...

13 usages  🧑 ChilimanTube *
public class Main {
    13 usages
    public static String prefix = "!";
    🧑 ChilimanTube *
    public static void main(String[] args) throws LoginException{

        // This is the code that connects the bot to the server.
        JDA bot = JDABuilder.createDefault( token: "OTcxMTEwNzQyMDcxODU3MTUy.GDon0y.-LmvjFFWQcd1Do7R6iKWn3mLLzUSWgFDg7Q0So")
            .setActivity(Activity.playing( name: "with Chili's brain"))
            .build();

        bot.addEventListener(new Commands());
    }
}
```

V této třídě se naváže spojení s Discord servery a pomocí unikátního tokenu se nahodí aplikace (bot). Zároveň se inicializují EventListenery, které monitorují chat.

### 3.2 Commands

Ve třídě Commands se nachází klíčová metoda onMessageReceived().

```

1 usage  🧑 ChilimanTube "
public class Commands extends ListenerAdapter {
    4 usages
    ArrayList<String> subjects = new ArrayList<>();
    /ResultOfMethodCallIgnored/
    public void onMessageReceived(@NotNull MessageReceivedEvent event){
        String[] args = event.getMessage().getContentRaw().split( regex: " ");

        //BAD WORD DETECTION
        // It's a method that checks if the message contains a bad word.
        if (Moderation.badWordDetection(args)) {
            event.getChannel().sendTyping().queue();
            event.getChannel().sendMessage( text: "Please, watch your language!").queue();
        }

        //PING
        // It's a command for latency check.
        if(args[0].equalsIgnoreCase( anotherString: Main.prefix + "ping")) {
            event.getChannel().sendMessage( text: "Pong!").queue();
        }

        //PREFIX DISPLAY
        // It's a command that displays the current prefix.
        if (args[0].equalsIgnoreCase( anotherString: Main.prefix + "prefix")){
            event.getChannel().sendTyping().queue();
            event.getChannel().sendMessage( text: "Ayo, my prefix is `!`").queue();
        }
    }
}

```

V této metodě se zatím nachází veškeré funkce programu.

Jedna z takových funkcí je funkce Poll, která umožňuje vytvořit Ano / Ne anketu s možností označení přítomných nebo všech uživatelů serveru.

```

// POLL
// It's a command that allows you to create a poll.
if (args[0].equalsIgnoreCase( anotherString: Main.prefix + "poll")) {
    String ping;
    String pollMessage;
    // Ping @everyone
    if(args[1].equalsIgnoreCase( anotherString: "pe")){
        ping = "@everyone ";
        //YesOrNo Poll
        if(args[2].equalsIgnoreCase( anotherString: "yn")){
            StringBuilder buffer = new StringBuilder();
            for(int i = 3; i < args.length; i++) {
                buffer.append(args[i]).append(" ");
            }
            event.getChannel().sendTyping().queue();
            pollMessage = buffer.toString();
            event.getChannel().sendMessage( text: ping + pollMessage).queue(message -> {
                message.addReaction( s: "✔").queue();
                message.addReaction( s: "✗").queue();
            });
        }
    }
}

// Ping @here
if(args[1].equalsIgnoreCase( anotherString: "ph")){
    ping = "@here ";
    //YesOrNo Poll
    if(args[2].equalsIgnoreCase( anotherString: "yn")){
        StringBuilder buffer = new StringBuilder();
        for(int i = 3; i < args.length; i++) {
            buffer.append(args[i]).append(" ");
        }
        event.getChannel().sendTyping().queue();
        pollMessage = buffer.toString();
        event.getChannel().sendMessage( text: ping + pollMessage).queue(message -> {
            message.addReaction( s: "✔").queue();
            message.addReaction( s: "✗").queue();
        });
    }
}

if(args[1].equalsIgnoreCase( anotherString: "yn")){

```

Část funkce Poll

Následuje funkce Timeout a Remove timeout. Tyto funkce dokáží dát Timeout uživateli, což mu zabrání v psaní zpráv a připojování do hlasových kanálů.

```

//TIMEOUT
// A command that allows you to timeout a user for a day.
if(args[0].equalsIgnoreCase( anotherString: Main.prefix + "timeout")){
    if(args.length == 2){
        event.getChannel().sendTyping().queue();
        String id = args[1];
        id = id.replace( target: "<@", replacement: "");
        id = id.replace( target: ">", replacement: "");
        long userId = Long.parseLong(id);
        RestAction<Member> restaction = event.getGuild().retrieveMemberById(userId);
        restaction.queue(member -> member.timeoutFor( amount: 1, TimeUnit.DAYS));
        event.getChannel().sendMessage( text: "Timeout for <@" + userId + "> has been set.").queue();
    } else {
        event.getChannel().sendMessage( text: "Incorrect command. Use `" + Main.prefix + "timeout @<member>`").queue();
    }
}

//REMOVE TIMEOUT
// Checking if the user is timeouted and if he is, it removes the timeout.
if(args[0].equalsIgnoreCase( anotherString: Main.prefix + "remove") && args[1].equalsIgnoreCase( anotherString: "timeout")){
    if(args.length == 3){
        Member member = event.getGuild().getMemberById(args[2].replace( target: "<@", replacement: "").replace( target: ">", replacement: ""));
        if(member != null && member.isTimedOut()){
            member.removeTimeout();
            event.getChannel().sendMessage( text: "Timeout for user @" + member + " was removed.").queue();
        }else{
            event.getChannel().sendMessage( text: "User @" + member + " is not timeouted.").queue();
        }
    }
}
}

```

Dále se zde nachází detekce anglických vulgarismů, která napomene uživatele, pokud nějaký takový výraz užije.

```

//BAD WORD DETECTION
// It's a method that checks if the message contains a bad word.
if (Moderation.badWordDetection(args)) {
    event.getChannel().sendTyping().queue();
    event.getChannel().sendMessage( text: "Please, watch your language!").queue();
}

```

Metoda badWordDetection() se nachází ve třídě Moderation.

### 3.3 Moderation

Ve třídě Moderation se zatím nachází jediná metoda a to badWordDetection(), která se používá ve třídě Commands. Metoda čerpá z textového dokumentu badwords.txt vulgarismy a následně prohledává chat a porovnává slova se slovy z dokumentu.

```

1 usage  🐞 ChilimanTube
public class Moderation {
    3 usages
    static Scanner scanner;
    /**
     * This function takes in an array of words from a message and checks if any of the words in the array match any of the strings in
     * the badwords.txt file
     *
     * @param words - The words to be checked.
     * @return A boolean value.
     */
1 usage  🐞 ChilimanTube
public static boolean badWordDetection(String @NotNull [] words){
    boolean result = false;
    {
        try {
            File badWords = new File( pathname: "src/badwords.txt");
            scanner = new Scanner(badWords);
            for (String s : words) {
                while (scanner.hasNextLine()) {
                    String word = scanner.nextLine();
                    if (s.equalsIgnoreCase(word)) {
                        result = true;
                    }
                }
            }
        } catch (FileNotFoundException e) {
            System.out.println("An error occurred while loading the file.");
            e.printStackTrace();
            throw new RuntimeException(e);
        }
    }
    return result;
}
}

```

### 3.4 Pom.xml

Soubor na načtení repositářů a dependencí.

```

<?xml version="1.0" encoding="UTF-8"?>
<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>DiscordBot</artifactId>
  <version>1.0-SNAPSHOT</version>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <configuration>
          <source>17</source>
          <target>17</target>
        </configuration>
      </plugin>
    </plugins>
  </build>

  <properties>
    <maven.compiler.source>18</maven.compiler.source>
    <maven.compiler.target>18</maven.compiler.target>
  </properties>

  <dependencies>
    <!-- https://mvnrepository.com/artifact/net.dv8tion/JDA -->
    <dependency>
      <groupId>net.dv8tion</groupId>
      <artifactId>JDA</artifactId>
      <version>5.0.0-alpha.11</version>
    </dependency>

    <dependency>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-clean-plugin</artifactId>
      <version>2.5</version>
    </dependency>
  </dependencies>

```

Část pom.xml souboru

## 4 Závěr

Na svém botu budu pracovat i po odevzdání ročníkové práce. Původním plánem Yechnaka bylo napojení na školní Sharepoint a stáhnutí Archu suplování, a na základě toho posílat denní oznámení o změnách v rozvrhu dané třídy. Avšak v průběhu jsem zjistil, že mé znalosti na to nestačí a mimo jiné na připojení potřebuji unikátní token od školy.

Každopádně s novými znalostmi, které čekají za každým rohem bych chtěl Yechnaka rozvinout třeba až na takovou úroveň, že by se začal používat na všech školních serverech.

## 5 Zdroje

- [1] StackOverflow. StackOverflow [online]. [cit. 2022-06-05]. Dostupné z: <https://stackoverflow.com>



- [2] JDA 4.4.0\_352 API [online]. [cit. 2022-06-05]. Dostupné z:  
<https://ci.dv8tion.net/job/JDA/javadoc/index.html>
- [3] Discord Developers [online]. [cit. 2022-06-05]. Dostupné z:  
<https://discord.com/developers>

Pro citování zdrojů využijte stránku: <https://www.citace.com/vytvorit-citaci>