Antlr program használata

Ez egy olyan egyszerű program, mely az alábbi szintaxis használatával harcosokat hoz létre, és harcoltatni is lehet őket.

A nyelv az alábbi:

```
grammar FightZone;


start   :   command+ ;

command :   'create' NAME 'with strength' INT    # createCommand
        |   'make' NAME 'and' NAME 'fight'       # fightCommand
        ;

NAME    :   [a-zA-Z]+ ;
INT     :   [0-9]+ ;
WS      :   [ \t\r\n]+ -> skip ;
```

A generált kódot a FightZoneDemo.java osztály használja ki:

```java
public class FightZoneDemo extends FightZoneBaseListener {

    private Map<String, Warrior> warriors = new HashMap<>();

    public static void main(String[] args) throws Exception {
        if (args.length == 0) {
            System.err.println("Please provide an input file.");
            System.exit(1);
        }

        FightZoneLexer lexer = new FightZoneLexer(CharStreams.fromFileName(args[0]));
        CommonTokenStream tokens = new CommonTokenStream(lexer);
        FightZoneParser parser = new FightZoneParser(tokens);
        ParseTree tree = parser.start();

        ParseTreeWalker walker = new ParseTreeWalker();
        FightZoneDemo listener = new FightZoneDemo();
        walker.walk(listener, tree);
    }

    // create command
    @Override
    public void enterCreateCommand(FightZoneParser.CreateCommandContext ctx) {
        String name = ctx.NAME().getText();
        int strength = Integer.parseInt(ctx.INT().getText());

        // create and store the warrior
        Warrior warrior = new Warrior(name, strength);
        warriors.put(name, warrior);
        System.out.println("Created warrior " + name + " with strength " + strength);
    }

    // fight command
    @Override
    public void enterFightCommand(FightZoneParser.FightCommandContext ctx) {
        String name1 = ctx.NAME(0).getText();
        String name2 = ctx.NAME(1).getText();

        Warrior warrior1 = warriors.get(name1);
        Warrior warrior2 = warriors.get(name2);

        if (warrior1 == null || warrior2 == null) {
            System.out.println("Error: One or both warriors not found!");
            return;
        }

        System.out.println(warrior1.fight(warrior2));
    }
}
```

A következő a teszt parancssor:

```
create Superman with strength 5
create Batman with strength 30
make Superman and Batman fight
```

A működés demonstrálása bash-ben maven használatával:

A repo megtekinthető itt:

```
https://github.com/FrighteningFunction/vimiac22_gyak43.git
```

Bashben maven használatával hajtsd végre:

```
mvn clean generate-sources compile
```

Teszt-futtatás:

```
mvn exec:java -Dexec.mainClass="com.fightzone.FightZoneDemo" -
Dexec.args="teszt.txt"
```

Használati esetek:

1. **Create a Warrior**:
   Szintaxis: `create NAME with strength INT`
2. **Make Two Warriors Fight**:
   Szintaxis: `make NAME and NAME fight`
3. **How Strong is a Warrior**:
   Szintaxis: `how strong is NAME?`

   Ez utóbbi nincs implementálva.