**package** grootan;

**import** java.io.File;

**import** java.io.IOException;

**import** java.net.URL;

**import** org.jsonschema2pojo.DefaultGenerationConfig;

**import** org.jsonschema2pojo.GenerationConfig;

**import** org.jsonschema2pojo.Jackson2Annotator;

**import** org.jsonschema2pojo.SchemaGenerator;

**import** org.jsonschema2pojo.SchemaMapper;

**import** org.jsonschema2pojo.SchemaStore;

**import** org.jsonschema2pojo.SourceType;

**import** org.jsonschema2pojo.rules.RuleFactory;

**import** com.sun.codemodel.JCodeModel;

**public** **class** Groot {

**public** **static** **void** main(String[] args) {

String packageName="grootan";

File inputJson= **new** File("."+File.***separator***+"temp.json");

File outputPojoDirectory=**new** File("."+File.***separator***+"convertedPojo");

outputPojoDirectory.mkdirs();

**try** {

**new** groot().convert2JSON(inputJson.toURI().toURL(), outputPojoDirectory, packageName, inputJson.getName().replace(".json", ""));

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

System.***out***.println("Encountered issue while converting to pojo: "+e.getMessage());

e.printStackTrace();

}

}

**public** **void** convert2JSON(URL inputJson, File outputPojoDirectory, String packageName, String className) **throws** IOException{

JCodeModel codeModel = **new** JCodeModel();

URL source = inputJson;

GenerationConfig config = **new** DefaultGenerationConfig() {

@Override

**public** **boolean** isGenerateBuilders() { // set config option by overriding method

**return** **true**;

}

**public** SourceType getSourceType(){

**return** SourceType.JSON;

}

};

SchemaMapper mapper = **new** SchemaMapper(**new** RuleFactory(config, **new** Jackson2Annotator(config), **new** SchemaStore()), **new** SchemaGenerator());

mapper.generate(codeModel, className, packageName, source);

codeModel.build(outputPojoDirectory);

}

}