

Необходимо написать:

- Программу на Java, которая получает на вход исходный код программы на Java, трансформирует его путем автоматической замены итеративных циклов на эквивалентные рекурсивные циклы, и в качестве выходных данных получаем трансформированный код программ на Java.
- В частности, необходимо:
 1. Заменять while
 2. Заменять do..while
 3. Заменять for
 4. Заменять foreach
 5. Заменять throws, return, break и continue.

Также необходимо:

- Писать подробные комментарии и хорошо читаемый код
- Избегать коллизии имен с сгенерированным кодом

Необходимо использовать:

- Библиотеку JavaParser.

Также необходимо использовать код ниже:

```
// Имя метода, которое создаём всякий раз как видим while
String methodName = "method_"+count++;

//Создаёт объект Loop для проверки циклов
Loop loop = new While(null, null, whileStmt);

// Объект Loop вычисляет список переменных, объявленных во всем цикле и
//используемых в цикле (n-образное пересечение)
List<Variable> variables = loop.getUsedVariables(methodDeclaration);

// создаем объект LoopVariables, который служит для преобразования списка
переменных в список аргументов и параметров
LoopVariables loopVariables = new LoopVariables(variables);

// Объект LoopVariables вычисляет для нас список аргументов метода
List<Expression> arguments = loopVariables.getArgs();
```

```
// while(**condition**)
Expression condition = whileStmt.getCondition();
```

```
//Object[] result = this.**method_x**()
MethodCallExpr methodCall = new MethodCallExpr();
methodCall.setName(methodName);
methodCall.setArgs(arguments);
```

```
/**Object**[] result = this. method_x()
ClassOrInterfaceType objType = new ClassOrInterfaceType();
objType.setName("Object");
```

```
/**Object[]** result = this. method_x()
ReferenceType refType = new ReferenceType();
refType.setArrayCount(1);
refType.setType(objType);
```

```
//Object[] result **=** this.method_x()
NameExpr resultExpr = new NameExpr("result");
List<VariableDeclarator> vars = new LinkedList<VariableDeclarator>();
VariableDeclarator var = new VariableDeclarator();
VariableDeclaratorId varId = new VariableDeclaratorId();
varId.setName(resultExpr.getName());
vars.add(var);
var.setId(varId);
var.setInit(methodCall);
```

```
/**Object[] result** = this.method_x()
VariableDeclarationExpr varDecExpr = new VariableDeclarationExpr();
varDecExpr.setType(refType);
varDecExpr.setVars(vars);
```

```
/**Object[] result = this.method_x()**  
ExpressionStmt exprStmt = new ExpressionStmt();  
exprStmt.setExpression(varDecExpr);  
  
/**If** condition {  
IfStmt newIf = new IfStmt();  
  
//if **condition** {...}  
newIf.setCondition(condition);  
  
//if cond **{...}**  
BlockStmt ifBody = new BlockStmt();  
newIf.setThenStmt(ifBody);  
List<Statement> ifBodyStmts = new LinkedList<Statement>();  
  
//if cond {**...**}  
ifBodyStmts.add(exprStmt);
```