## The Jimple methods used to create statements and expressions in lab4

| API Method Signature | Description |
|---|---|
| `Jimple.newGotoStmt(Unit)` | create new **goto** statement branching to the statement given as argument. Return type of newGotoStmt(): GotoStmt which is a subinterface of Unit. |
| `Jimple.newNopStmt()` | create a new **nop** statement. Return type of newNopStmt(): NopStmt which is a subinterface of Unit. |
| `All the following Jimple.new*Expr()` | The (two) argument(s) needs to be a constant or a local. |
| `Jimple.newEqExpr(Value,Value)` | create a new "==" expression. Return type of newEqExpr(): EqExpr which is a subinterface of Value. |
| `Jimple.newAddExpr(Value,Value)` | create a new "+" expression. Return type of newAddExpr(): AddExpr which is a subinterface of Value. |
| `Jimple.newSubExpr(Value,Value)` | create a new "-" expression. Return type of newSubExpr(): SubExpr which is a subinterface of Value. |
| `Jimple.newMulExpr(Value,Value)` | create a new "*" expression. Return type of newMulExpr(): MulExpr which is a subinterface of Value. |
| `Jimple.newDivExpr(Value,Value)` | create a new "/" expression. Return type of newDivExpr(): DivExpr which is a subinterface of Value. |
| `Jimple.newRemExpr(Value,Value)` | create a new "%" expression. Return type of newRemExpr(): RemExpr which is a subinterface of Value. |
| `Jimple.newNegExpr(Value)` | create a new unary minus expression. Return type of newNegExpr(): NegExpr which is a subinterface of Value. |
| `Jimple.newArrayRef(Value,Value)` | create a new array index expression of the form a[i]. The two arguments need to be locals. Return type of newArrayRef(): ArrayRef which is a subinterface of Value. |
| `IntConstant.v(int)` | create a new integer literal. IntConstant is a class implementing the Value interface. |
| `StringConstant.v(String)` | create a new string literal. StringConstant is a class implementing the Value interface. |