1.说明：

#自定义统一的通用模板

#快捷键：md 说明：user default method

/\*\*

\* $METNOD$

\* @return $RETURN$

\* -----------------------------------------------------------------------------------------------------------------

\*/

public $RETURN$ $METNOD$(){

$END$

}

​

2.说明：

#自定义统一的返回为List的通用模板

#快捷键：ml 说明：user return list method

​/\*\*

\* $METNOD2$

\* @return List<$RETURN1$>

\* -----------------------------------------------------------------------------------------------------------------

\*/

public List<$RETURN1$> $METNOD2$(){

$END$

}

3.说明：

#自定义统一的返回为String的通用模板

#快捷键：ms 说明：user return string method

​/\*\*

\* $METNOD$

\* @return String

\* -----------------------------------------------------------------------------------------------------------------

\*/

public String $METNOD$(){

$END$

}

4.说明：

#自定义统一的返回为int的通用模板

#快捷键：mi 说明：user return int method

​/\*\*

\* $METNOD$

\* @return int

\* -----------------------------------------------------------------------------------------------------------------

\*/

public int $METNOD$(){

$END$

}

​

5.说明：

# 单元测试初始化模板

#快捷键：tinit 说明：test case init

@Before

public void before(){}

@Test

public void $method$() {

$END$

}

6.说明：

# 单元测试通用方法模板

#快捷键：tmd 说明：test case default method

@Test

public void $method$() {

$END$

}

7.

说明：

# 新建对象通用方法模板

#快捷键：no 说明：new object

$val$ name=new $val$();

8.

说明：

# 新建对象通用方法模板

#快捷键：ns 说明：new string

String name=new String();

9.

说明：

# 新建对象通用方法模板

#快捷键：nl 说明：new list

List<$val$> list= Arrays.asList();

10.

说明：

# 新建lambda函数方法模板

#快捷键：lamd 说明：user lambda

()->{};

11.

说明：

# 新建lambda函数方法模板

#快捷键：nclass 说明：user new class

class $val$ {

$END$

}

12.

说明：

# 新建lambda函数方法模板

#快捷键：nitface 说明：user new interface

interface $val$ {

$END$

}

​