|  |  |  |
| --- | --- | --- |
| **Name** | **Key** | **Description** |
| **Physical lines** | lines | 回车数目 |
| **Comment lines** | comment\_lines | Javadoc、多行注释、单行注释的总数。空注释行、头文件中的注释（主要用于定义许可证）以及commented-out行均不会包括在内。   |  | | --- | | /\*\*   \* This is a javadoc block   \*  <- empty comment line considered as a blank line   \*/  <- empty comment line considered as a blank line    /\*   \* This is a multi-comment block   \*/    // This is a single-comment block  // log("Debug information"); <- commented-out line of code is not a comment line | |
| **Commented-out lines of code** | commented\_out\_code\_lines | |  | | --- | | 注释掉的代码行数。Javadoc块不会被扫描  /\*  \* someoneCommentMeOutOneDay();  \* nobodyKnowWhatAmISupposedToDo();  \*/ | |
| **Lines of code** | ncloc | **physical lines - blank lines - comment lines - header file comments - commented-out lines of code** |
| **Density of comment lines** | comment\_lines\_density | comment lines / (lines of code + comments lines) \* 100% |
| **Packages** | packages | 包数目 |
| **Classes** | classes | 类的数目，包括内部类、接口、枚举及注释标签 |
| **Files** | files | 分析的文件数目 |
| **Directories** | directories | 分析的目录数目 |
| **Accessors** | accessors | Getter及setter方法数目，通过get(reading)或set(writing)一个类属性  // Getters  public String getName(){       return this.name;  }  public boolean isParent(){       return this.isParent;  }  // Setters  public void setName(String name){       this.name = name;  }  public void setIsParent(boolean isParent){       this.isParent = isParent;  } |
| **Methods** | Functions | 方法数目，不包括访问器。一个构造函数算作一个方法 |
| **Public API** | public\_api | 公共类、方法（不包括访问器）以及属性（不包括public final static修饰的属性） |
| **Public undocumented API** | public\_undocumented\_api | 不包含javadoc块的公共API数目 |
| **Density of public documented API** | public\_documented\_api\_density | (public API - undocumented public API) / public API \* 100% |
| **Duplicated lines** | duplicated\_lines | 重复的物理行数 |
| **Duplicated blocks** | duplicated\_blocks | 重复的代码块数 |
| **Duplicated files** | duplicated\_files | 涉及到重复代码的文件数目 |
| **Density of duplicated lines** | duplicated\_lines\_density | Duplicated lines / Physical lines \* 100% |
| **Statements** | statements | Java语言规范中没有块定义的语句数目；此数目在遇到含有if, else, while, do, for, switch, break continue, return, throw, synchronized, catch, finally等关键字的语句时增加。例如：  //  i = 0;  if (ok)  if (exit) {  if (3 == 4);  if (4 == 4) { ; }  } else {  try{}  while(true){}  for(...){}  ...  语句数目不会随着以下情况增加，类、方法、字段、注释定义、包以及import定义。 |
| **Complexity** | complexity | 圈复杂度也被称为McCabe度量。它简单归结为一个方法中’if’ , ’for’ , ’while’等块的数目。当一个方法的控制流分割，圈计数器加1.  除不被认为是方法的访问器外，每个方法默认有最小的值1，所以不会增加复杂度。对于以下的每一个java关键字/语句，圈复杂度均会加1：  if  for  while  case  catch  throw  return (当不是一个方法最好一个语句时)  &&  ||  ?  注意else, default及finally不会增加CCN的值。另一方面，一个含switch语句及很大块case语句的简单方法可以拥有一个令人惊讶的高的CCN值（同时，当将switch块转化为等效的if语句时，它具有相同的CCN值）。  例如，下面的方法具有5的复杂度。  public void process(Car myCar){          <- +1      if(myCar.isNotMine()){               <- +1           return;                         <- +1      }      car.paint("red");      car.changeWheel();      while(car.hasGazol() && car.getDriver().isNotStressed()){   <- +2           car.drive();      }      return;  } |
| **Average complexity by method** | function\_complexity | 方法的平均圈复杂度 |