Findbugs功能强大，有多种使用方式，本项目需要用到的基本上就是利用command line进行操作，完整的command line options 如下：Usage: findbugs [general options] -textui [command line options...] [jar/zip/cla

ss files, directories...]

General options:

-gui Use the Graphical UI (default behavior)

-gui1 Use the older Graphical UI

-textui Use the Text UI

-jvmArgs args Pass args to JVM

-maxHeap size Maximum Java heap size in megabytes (default=768)

-javahome <dir> Specify location of JRE

-help Display command line options

-debug Enable debug tracing in FindBugs

Command line options:

General FindBugs options:

-project <project> analyze given project

-home <home directory> specify FindBugs home directory

-pluginList <jar1[;jar2...]> specify list of plugin Jar files to

load

-effort[:min|less|default|more|max] set analysis effort level

-adjustExperimental lower priority of experimental Bug

Patterns

-workHard ensure analysis effort is at least

'default'

-conserveSpace same as -effort:min (for backward c

ompatibility)

-showPlugins show list of available detector plu

gins

Output options:

-timestampNow set timestamp of results to be curr

ent time

-quiet suppress error messages

-longBugCodes report long bug codes

-progress display progress in terminal window

-release <release name> set the release name of the analyze

d application

-experimental report all warnings including exper

imental bug patterns

-low report all warnings

-medium report only medium and high priorit

y warnings [default]

-high report only high priority warnings

-sortByClass sort warnings by class

-xml[:withMessages] XML output (optionally with message

s)

-xdocs xdoc XML output to use with Apache

Maven

-html[:stylesheet] Generate HTML output (default style

sheet is default.xsl)

-emacs Use emacs reporting format

-relaxed Relaxed reporting mode (more false

positives!)

-train[:outputDir] Save training data (experimental);

output dir defaults to '.'

-useTraining[:inputDir] Use training data (experimental); i

nput dir defaults to '.'

-sourceInfo <filename> Specify source info file (line numb

ers for fields/classes)

-projectName <project name> Descriptive name of project

-output <filename> Save output in named file

-nested[:true|false] analyze nested jar/zip archives (de

fault=true)

Output filtering options:

-bugCategories <cat1[,cat2...]> only report bugs in given categorie

s

-onlyAnalyze <classes/packages> only analyze given classes and pack

ages; end with .\* to indicate classes in a package, .- to indicate a package pre

fix

-excludeBugs <baseline bugs> exclude bugs that are also reported

in the baseline xml output

-exclude <filter file> exclude bugs matching given filter

-include <filter file> include only bugs matching given fi

lter

-applySuppression Exclude any bugs that match suppres

sion filter loaded from fbp file

Detector (visitor) configuration options:

-visitors <v1[,v2...]> run only named visitors

-omitVisitors <v1[,v2...]> omit named visitors

-chooseVisitors <+v1,-v2,...> selectively enable/disable detector

s

-choosePlugins <+p1,-p2,...> selectively enable/disable plugins

-adjustPriority <v1=(raise|lower)[,...]> raise/lower priority of warnings fo

r given visitor(s)

Project configuration options:

-auxclasspath <classpath> set aux classpath for analysis

-sourcepath <source path> set source path for analyzed classe

s

-exitcode set exit code of process

-noClassOk output empty warning file if no cla

sses are specified

-xargs get list of classfiles/jarfiles fro

m standard input rather than command line

基本的操作我总结了一下：Usage: findbugs [general options] -textui [command line options...] [jar/zip/class files, directories...],由于无法新建一个project，所以要将所有要分析的文件放入同一个文件夹，然后将路径作为参数传入。