总结：by wyr

一：导出jar类库。选中.java文件，右键，导出为java下的jar即可。

二：使用jar类库。新建文件夹，将jar复制进去，选中右键，构建类路径，构建类路径即可。在编写代码是，可以直接import引入该jar中的类。注：import是为了导入类。

原网址：<http://blog.csdn.net/renfufei/article/details/16545435>

以下为网址内容。

Notes:通过这种方法，我确实实现了jar的添加，感觉很棒。

代码复用是软件开发中很重要的一个原则.将常用的函数构建为类库是一个很好的解决方法,什么时候要用到这些功能,只需要调用就好了.

下面将演示如何导出jar类库与使用jar包:

**1. 导出jar类库**

打开Eclipse,选定工作空间.工作空间一般来说,是用户保存配置信息,以及保存源码的地方.

然后在菜单栏中选择 **File --> New --> Java Project**, 选择以后的界面将如**图2** 所示。

如果没有Java Project这个快捷项,可以选择 Other,然后再进行选择,选择 Other以后的界面如 **图1** 所示:

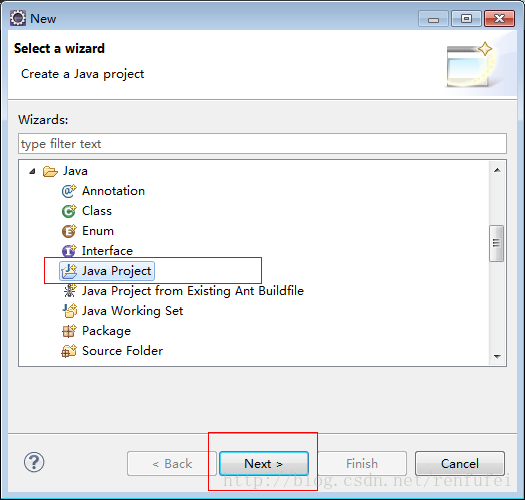


图1

如图1所示,在其中找到 **Java --> Java Project** 以后,选择Next即可进入下一步, 如 图2 所示:

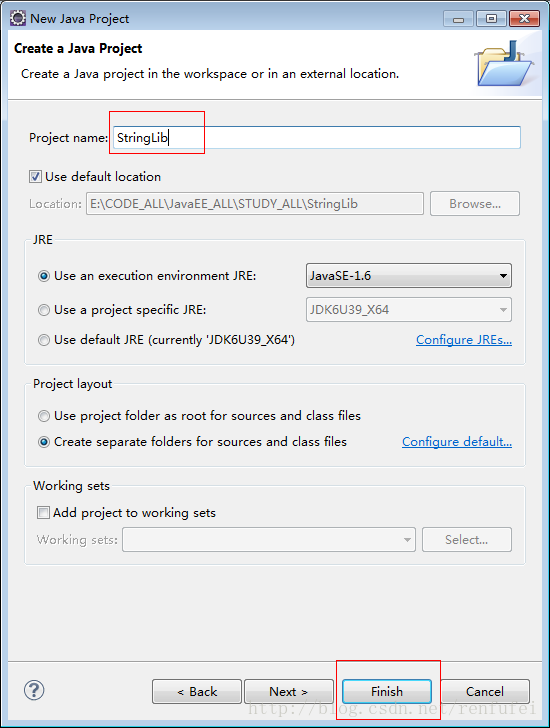


图2

在图2所示的界面中,输入项目的名称,比如: **StringLib**, 使用默认的存储位置,使用默认的JRE, 然后点击 Finish 即可.

项目创建完成以后,关闭不必要的那些View视图界面,(点击X即可).

接着,选择左侧Package视图下的src目录,右键单击, 选择 **New --> Class** , 如**图3** 所示:

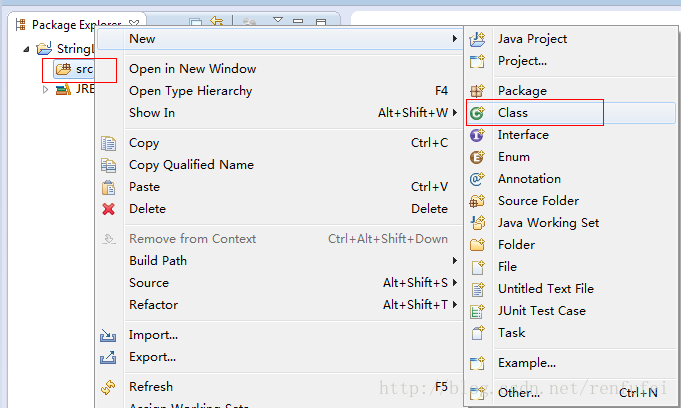


图3

在接下来的向导页面中,输入 类名，比如: **StringUtil** , 选择包名,比如: **com.renfufei.util** ,然后点击完成即可,如**图4** 所示:

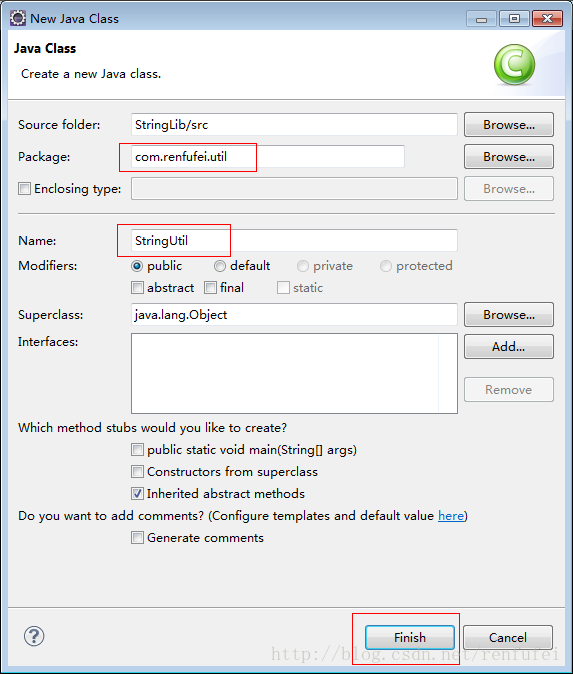


图4

创建类完成以后,根据需要写相关的代码,示例代码如下:

**[java]** [view plaincopyprint?](http://blog.csdn.net/renfufei/article/details/16545435)

1. package com.renfufei.util;
2. import java.util.Random;
3. import java.util.UUID;
4. public class StringUtil {
5. /\*\*
6. \* 判断某个字符串不为空
7. \* @param s
8. \* @return
9. \*/
10. public static boolean notNull(String s){
11. return !isNull(s);
12. }
13. /\*\*
14. \* 是整形正整数
15. \* @param s
16. \* @return
17. \*/
18. public static boolean isLong(String s){
19. if (null != s && s.matches("^[+-]?\\d+$")) {
20. return true;
21. } else {
22. return false;
23. }
24. }
25. /\*\*
26. \* 判断某个字符串是空
27. \* @param s
28. \* @return
29. \*/
30. public static boolean isNull(String s){
31. return null == s || s.trim().length() < 1 || "null".equalsIgnoreCase(s);
32. }
33. /\*\*
34. \* 随机生成一串字符串
35. \*/
36. public static String getUUID(){
37. String uuid=UUID.randomUUID().toString();
38. return uuid;
39. }
40. /\*\*
41. \* 转换为HTML,防止JS注入
42. \* @param s
43. \* @return
44. \*/
45. public static String toHTML(String s){
46. if (isNull(s)) {
47. s = "";
48. }
49. s = s.replaceAll("&", "&");
50. s = s.replaceAll("<", "<");
51. s = s.replaceAll(">", ">");
52. s = s.replaceAll("\"", """);
53. s = s.replaceAll(" ", " ");
54. s = s.replaceAll("\r\n", "\n");
55. s = s.replaceAll("\r", "\n");
56. s = s.replaceAll("\n", "<p>");
57. //
58. return s;
59. }
60. public static String handleSpecialChar(String s){
61. if (isNull(s)) {
62. s = "";
63. }
64. s=s.replaceAll("\r","").replaceAll("\n","").replace("\\","\\\\").replace("\"","\\\"").replace("<", "<").replace(">", ">");
65. //
66. return s;
67. }
68. /\*\*
69. \* 截断。返回截断后的字符串
70. \* @param s 需要阶段的字符串
71. \* @param beginIndex 开始index
72. \* @param endIndex 结束index
73. \* @return
74. \*/
75. public static String truncate(String s,int beginIndex,int endIndex){
76. if (null == s) {
77. return "";
78. }
79. int len = s.length();
80. if (endIndex > len) {
81. return s.substring(beginIndex);
82. }else {
83. return s.substring(beginIndex, endIndex)+"...";
84. }
85. }
86. /\*\*
87. \* 随机生成N位字符串--大写字母和数字
88. \* @param num
89. \* @return
90. \*/
91. public static String getRandStr(int num){
92. String randString = "23456789ABCDEFGHJKMNPQRSTUVWXYZ";
93. return randStr(randString,num);
94. }
95. private static String randStr(String randString,int num){
96. Random rnd = new Random();
97. String rand = "";
98. for(int i=0;i<num;i++){
99. rand =rand + String.valueOf(randString.charAt(rnd.nextInt(randString.length())));
100. }
101. return rand;
102. }
103. // 简单的单元测试
104. public static void main(String[] args) {
105. String s = "--100";
106. System.out.println(isLong(s));
107. }
108. }

package com.renfufei.util;

import java.util.Random;

import java.util.UUID;

public class StringUtil {

/\*\*

\* 判断某个字符串不为空

\* @param s

\* @return

\*/

public static boolean notNull(String s){

return !isNull(s);

}

/\*\*

\* 是整形正整数

\* @param s

\* @return

\*/

public static boolean isLong(String s){

if (null != s && s.matches("^[+-]?\\d+$")) {

return true;

} else {

return false;

}

}

/\*\*

\* 判断某个字符串是空

\* @param s

\* @return

\*/

public static boolean isNull(String s){

return null == s || s.trim().length() < 1 || "null".equalsIgnoreCase(s);

}

/\*\*

\* 随机生成一串字符串

\*/

public static String getUUID(){

String uuid=UUID.randomUUID().toString();

return uuid;

}

/\*\*

\* 转换为HTML,防止JS注入

\* @param s

\* @return

\*/

public static String toHTML(String s){

if (isNull(s)) {

s = "";

}

s = s.replaceAll("&", "&");

s = s.replaceAll("<", "<");

s = s.replaceAll(">", ">");

s = s.replaceAll("\"", """);

s = s.replaceAll(" ", " ");

s = s.replaceAll("\r\n", "\n");

s = s.replaceAll("\r", "\n");

s = s.replaceAll("\n", "<p>");

//

return s;

}

public static String handleSpecialChar(String s){

if (isNull(s)) {

s = "";

}

s=s.replaceAll("\r","").replaceAll("\n","").replace("\\","\\\\").replace("\"","\\\"").replace("<", "<").replace(">", ">");

//

return s;

}

/\*\*

\* 截断。返回截断后的字符串

\* @param s 需要阶段的字符串

\* @param beginIndex 开始index

\* @param endIndex 结束index

\* @return

\*/

public static String truncate(String s,int beginIndex,int endIndex){

if (null == s) {

return "";

}

int len = s.length();

if (endIndex > len) {

return s.substring(beginIndex);

}else {

return s.substring(beginIndex, endIndex)+"...";

}

}

/\*\*

\* 随机生成N位字符串--大写字母和数字

\* @param num

\* @return

\*/

public static String getRandStr(int num){

String randString = "23456789ABCDEFGHJKMNPQRSTUVWXYZ";

return randStr(randString,num);

}

private static String randStr(String randString,int num){

Random rnd = new Random();

String rand = "";

for(int i=0;i<num;i++){

rand =rand + String.valueOf(randString.charAt(rnd.nextInt(randString.length())));

}

return rand;

}

// 简单的单元测试

public static void main(String[] args) {

String s = "--100";

System.out.println(isLong(s));

}

}

书写过程中,记得随时 **CTRL+S** 保存,如果书写错误,可以使用**CTRL+Z** 进行撤销.

保存以后,在左侧包浏览器 视图界面中,选择需要导出的包,或者选择项目, 点击右键,在弹出菜单中选择 Export ,如 **图5**所示:

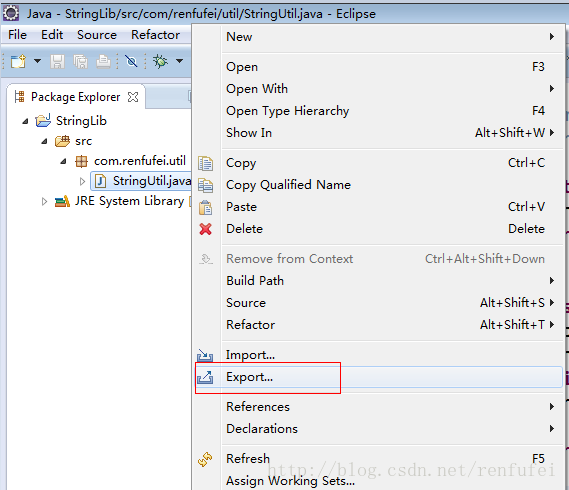


图5

在接下来的向导界面中,找到并选择 **Java --> JAR File** 选项,点击Next,如**图6** 所示:

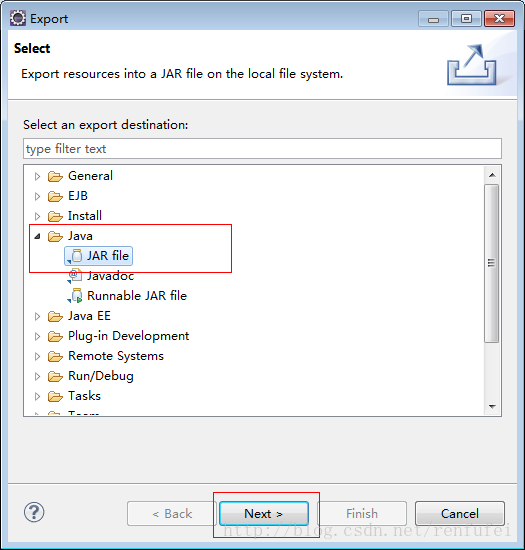


图6

在接下来的界面中，选择要导出的资源，如 **图7** 所示，我们只选择了 **com.renfufei.util** 这个包, 在下方有几个可勾选的选项，其中一个是 导出生成的class文件,还有一个是 导出Java源文件, 图7中都用红色方框标识出来了, 并浏览选择输出文件,此处我们将输出位置选择为桌面,文件名输入:**myutil.jar**. 随后点击 Finish即可.

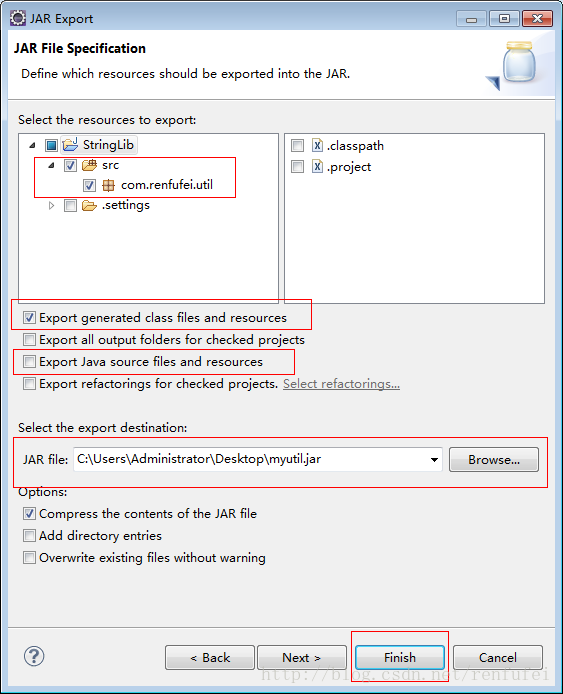


图7

OK，如果没有报错,那么导出完成,请在桌面或者保存位置查看导出的jar类库文件,可以使用 winRAR工具或者zip工具打开查看 jar文件的结构。

**2. 使用 jar 类库**

说明,如果是JavaEE或者JavaWeb项目,在Eclipse中，只需要将 相应的 jar 文件拷贝到 WebRoot下的 WEB\_INF 下 的 lib 目录中即可. 可以略过下面的部分.

新建Java Project, 选择项目名字,比如 **UseLib ,** 然后在项目上单击右键, 在弹出菜单中选择 New --> Folder ,如**图8** 所示:

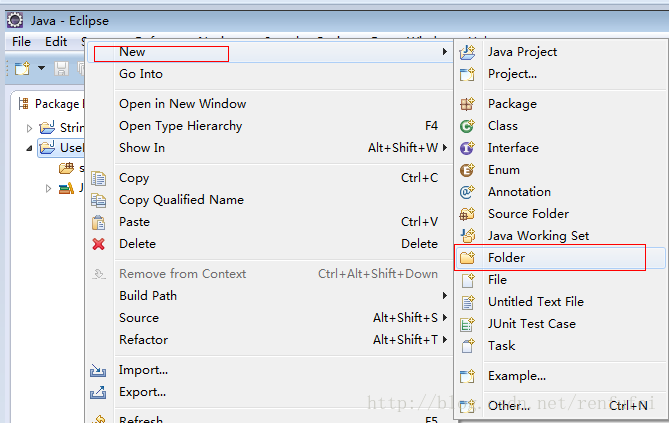


图8

在弹出的向导界面中,输入文件夹名字, 如 **lib** ,然后点击 Finish 完成.如**图9** 所示:

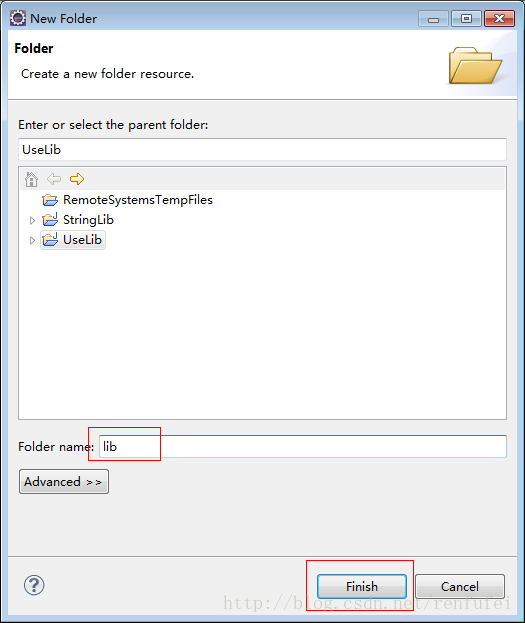


图9

完成以后,将需要导入的jar文件,比如 **myutil.jar** 在文件浏览器中复制( **CTRL+C** ),然后进入Eclipse,点击 lib 文件夹, 粘贴( **CTRL +V** ).

然后在 lib下的 myutil.jar 文件上单击右键，在弹出菜单中选择 Build Path --> Add to Build path ;此时，就将类库文件加入到执行路径中了,如**图10** 所示:

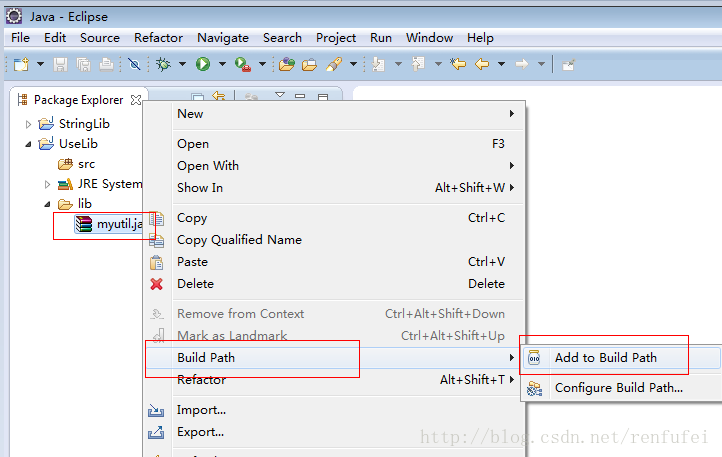


图10

可以在 Referenced Libraries 中看到相应的 jar 包.至此，导入完成.

**3. 测试**

在UseLib项目的 src中, 新建一个类 TestUtil, 包 com.renfufei.test ,可以看到，编译并没有出错,选择右键，Run As --> Java Application. 查看输出结果.

测试代码如下:

**[java]** [view plaincopyprint?](http://blog.csdn.net/renfufei/article/details/16545435)

1. package com.renfufei.test;
2. import com.renfufei.util.StringUtil;
3. public class TestUtil {
4. public static void main(String[] args) {
5. String num = "100";
6. System.out.println(StringUtil.isNull(num));
7. }
8. }