Я обычно использую для файловых операций отдельный класс с такими методами:

|  | |
| --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | **public** **static** **boolean** writeExternal(**InputStream** is, **String** path) {          Log.e("FM", "writeExternal " + path);  **if** (!isExternalAvailable()) **return** **false**;  **File** dest = **new** **File**(EXT\_DIR.getAbsolutePath() + "/" + path);  **FileOutputStream** writer = **null**;  **try** {  **if** (!dest.exists()) {                  dest.getParentFile().mkdirs();                  dest.createNewFile();              }              writer = **new** **FileOutputStream**(dest);              writeFromInputToOutput(is, writer);  **return** **true**;          }  **catch** (**IOException** e) {              Log.e("FM", "ERROR " + e);  **return** **false**;          }  **finally** {              close(writer);          }      } | |  |
|  | |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | **public** **static** **int** writeFromInputToOutput(**InputStream** is, **OutputStream** os) {    **byte**[] buffer = **new** **byte**[BUFFER\_SIZE];  **int** bytesRead = EOF\_MARK;  **int** count = 0;  **try** {  **while** ((bytesRead = is.read(buffer)) != EOF\_MARK) {                  os.write(buffer, 0, bytesRead);                  count += bytesRead;              }          }  **catch** (**IOException** e) {  **return** -1;          }  **return** count;      } | |  |

|  | |
| --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | **public** **static** **InputStream** readExternal(**String** path) {  **if** (!isExternalAvailable()) **return** **null**;  **File** source = **new** **File**(EXT\_DIR.getAbsolutePath() + "/" + path);  **InputStream** is = **null**;  **if** (source.exists()) {  **try** {                  is = **new** **FileInputStream**(source);              }  **catch** (**FileNotFoundException** e) {  **return** **null**;              }          }  **return** is;      } | |  |
|  | |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | **public** **static** **boolean** close(**Closeable** closeable) {  **if** (closeable != **null**) {  **try** {                  closeable.close();  **return** **true**;              }  **catch** (**IOException** e) {  *// e.printStackTrace();*  **return** **false**;              }          }  **else** {  **return** **false**;          }      } | |  |

Там есть и другие полезные методы, если интересно, пиши в ЛС чтоб не флудить.