

NormLab规格说明书

NormLab: 一个基于Python的留学生作业的规范化处理和分析系统

教师在超星系统中发布Labworks（实验作业），每次Labwork的任务书按照“[实验序号]: [实验名称].docx”的格式命名，如：Lab01: Specification-Based Testing.docx、Lab02: Boundary Testing.docx、Lab03: JUnit for Unit Test.docx。

学生每次完成Labwork中规定的任务后，须提交一份实验报告和一份实验代码至超星系统中：是否需要提交实验代码，是由具体的实验任务决定的；但实验报告是必须提交的。为了方便教师评阅学生提交的实验报告，所提交的实验报告和实验代码须遵守如下**提交规范**：

1. 实验报告须以“[实验序号]-[学号]-[学生姓名简称].docx”来命名，如：Lab01-L201926630134-Matin.docx，Lab01-L201926630106-Ronald.docx；由于留学生的名字通常较长，为清晰起见，需要在实验报告的文件名中使用学生姓名的简称，每位学生的全名及所对应的简称详见：[International Student List](#)。
2. 实验代码须以.zip格式进行打包，并且以“Code.zip”来命名；
3. 在所提交的代码包Code.zip中，不允许有.class等二进制文件，也不允许有.idea等集成开发环境（IDE）所生成的工程文件；例如，由IntelliJ IDEA建立的Java项目中，通常会包含.idea、src、target等三个文件夹，在提交代码时，只允许提交src文件夹下的内容。
4. 每个学生只允许提交一份实验报告和一个代码压缩包，不允许提交其它文件和压缩包；
5. 不允许将实验报告和代码压缩包再压缩为一个压缩包提交到系统中；
6. 所提交的文件中不允许有嵌套的压缩包，也就是，不允许.zip文件中还包含.zip文件的情况。

显然，上述提交规范是比较复杂的，在实际执行时，很容易疏漏其中的某些项目，甚至有些学生可能完全不遵守这些提交规范。一个实际的例子详见：[Lab03-JUnit for Unit Test.zip](#)。这个压缩包是教师在评阅学生提交的某次实验作业时，通过超星系统导出的。解压该压缩包后，可以发现系统为每位学生建立了一个独立的.zip压缩包，该压缩包以“[学号]-[学生的全名].zip”来命名。进一步解压该压缩包，可以看到其中包含了两部分内容：

1. 学生自己提交的实验作业，可以看出，大部分学生的提交都不是很规范；
2. 系统还为每位学生建立的一个名为“计算机科学与技术学院-软件工程（留学生）-2019软件工程（留学生）-[学号]-[学生的全名].doc” Word文件，显然，该文件是冗余的，须删除。

为了便于教师进行评阅，现须基于Python为教师开发一个名为 `NormLab` 的软件系统，使得教师能够使用该系统对所提交的实验作业进行规范化处理，并报告雷同作业。雷同作业可以根据如下三个条件进行判断：

1. `same/similar submission size`: 所提交的实验作业是否有相同/相似的大小？
2. `same/similar report name`: 所提交的实验作业中所包含的各文件（尤其是实验报告）的名字是否相同/相似？
3. `same/similar submission structure`: 所提交的实验作业的文件结构是否相同/相似？

使用 `NormLab` 软件系统，教师只须在终端命令行下输入 `python normlab.py [LabXXX.zip]`，即可生成完全遵守上述规范的所有学生的实验作业，并且给出疑似雷同作业的报告清单，详见：[🔗 Similar Works Report](#)。这里，`normlab.py` 为 `NormLab` 的主程序，`LabXXX.zip` 为超星系统所导出的某次实验作业的压缩包的名字（如：`Lab03-JUnit for Unit Test.zip`）。

除了满足提交规范中的6点要求外，经过 `NormLab` 处理后的作业还须满足如下条件：

1. 所有学生提交的实验作业位于 `LabXXX` 目录下；
2. 所有学生提交的实验报告直接放在 `LabXXX` 目录下，以 “[实验序号]-[学号]-[学生姓名简称].docx” 的格式命名；
3. 每个学生提交的实验代码放在一个单独的文件夹中，该文件夹以 “[实验序号]-[学号]-[学生姓名简称]” 的格式来命名，并且，
 - a. 该学生所提交的所有压缩文件（`.zip/.rar`）已经被解压、合并到该文件夹下；
 - b. 该文件夹及其子文件夹下，不存在任何 `.zip/.rar` 压缩包，也不存在任何二进制文件和 IDE 所生成的工程文件；
 - c. 对每个 `.zip/.rar` 压缩包解压后，如果该解压后的文件夹（称为文件夹 `A`）下有且仅有一个与压缩包同名的文件夹（称为文件夹 `A'`），则将 `A'` 下的所有文件都上移一级目录至文件夹 `A` 下，并删除空的文件夹 `A'`；再递归执行这一过程，直到目录 `A` 下不存在一个孤立的与 `A` 同名的文件夹；
 - d. 对每个 `.zip/.rar` 压缩包解压后，如果该解压后的文件夹下有且仅有一个名为 `src` 的文件夹，就将该 `src` 下的所有文件都上移一级目录，并删除空的文件夹 `src`；
 - e. 上述的“一个文件夹 `A` 下有且仅有一个文件夹 `B`”指的是：`A` 下仅有一个文件夹 `B`，并且没有任何其它文件。