EECS 2030: 实验室2

可在最多三个学生的小组中进行

激励

这个实验室将让你练习以下内容：

- 创建一个实用类
- 回顾Java中的数组和字符串操作
- 实现静态功能
- 编写简单的单元测试
- 使用Javadoc正确记录你的类

第一部分：入门

下载一个包含 Lab 2 Eclipse 项目的 zip 文件。通过以下方

式将该项目导入Eclipse：

1. 在**文件**菜单下选择**导入...**
2. 在 "**常规** "下选择 "**现有项目进入工作区** "并按 "**下一步**"。
3. 点击**选择存档文件**单选按钮，并点击**浏览...**按钮。
4. 在出现的文件浏览器中，导航到你的下载目录（具体位置取决于你工作的电脑；在实验室的电脑上，文件可能会出现在你的主目录中）。
5. 选择文件**Lab2_S23.zip**并点击**确定**
6. 单击 "**完成**"。

探索现有的方法和测试案例，试着理解每一行的目的。例如，期望构造函数采取什么参数，输出或返回值应该是什么，以及为什么某些操作应该被禁止。

文本加密

在这个实验室中，你必须根据指定的加密/解密密钥，实现一个用于加密和解密先前加密的文本（表示为一个字符串）的应用程序。

该文件是使用替代密码[1] 和柱状移位密码[2] 的组合进行加密的。

加密程序如下：

- 读取加密密钥（这是一个对称系统，加密和解密都使用同一个密钥）；我们假设，密钥是 "ABCDEFGH"。

[1] https://en.wikipedia.org/wiki/Substitution_cipher
[2] https://en.wikipedia.org/wiki/Transposition_cipher#Columnar_transposition

- 计算钥匙的哈希值（这是一个看起来很随机的数字，取决于给定的字符串）。在这种情况下，哈希值取自hashCode方法的输出：

```
int hash = key.hashCode();
```

在这种情况下，该值为2042300548。

- 使用该值作为种子来初始化随机对象：

```
random = new Random(hash);

或随机.setSeed(hash);
```

在这一点上，我们有一个随机数生成器；然而，其伪随机数的序列是由上述种子决定的。

- 现在，我们使用这个生成器生成的数字，从以下字母（以空格结尾）"ABCDEFGHIJKLMNOPQRSTUVWXYZ "创建一个*替换模式*。
  - 产生两个介于0和26之间的随机数（种子已设定，对吗？）
  - 将上述字符串中这些位置的字母进行交换[3]
  - 重复100次
  - 迭代100次后应得到 "GCWHAKSXJMDLFUB ITVYRPZENQO"
  - 这意味着，原文中的A字母将被替换成G，B被替换成C，C被替换成W，以此类推：

"abcdefghijklmnopqrstuvwxyz "
"gcwhaksxjmdlfub itvyrpzenqo"

- 同样地，我们应该在转置步骤中为列的顺序创建一个模式：
  - 以{0, 1, 2, 3, 4, 5, 6, 7}模式开始
  - 将随机种子设置为相同的初始值
  - 产生两个0到7之间的随机数
  - 将上述数组中这些位置的数字进行交换
  - 重复100次
  - 经过100次迭代，应该得到[3, 2, 7, 4, 1, 0, 5, 6]。
  - 这意味着，将按3、2、7等顺序读取列，而不是按顺序读取。

- 以64个字符为单位阅读文本

- 说，前64个字符是

"在密码学中，替换密码是一种编码b的方法"

[3] 你可能想用一个StringBuilder类来代替String

- 应用替代法：

  "juowtn ybstg xnoogovrcvyjyryjbuowj xatojvogofayxbhobkoauwbhjusoc"

- 应用换位法（逐行写，然后按3、2、7、...的顺序读列）：

  j u o w t n y b s t g x
  n o g o v r c v y **j** y r
  y **j** b u o w **j** x a t o **j**
  V O G O F A Y X B H
  O B K O A U W B H **J**
  U S O C

- 结果："wgvyxobjotor gohyoyojxuct rjafkuusgyjohbjbojwvbwnxcbtaos nvuoyao"

- 对于解密，需要逆转转置和替换步骤。

对于这个实验室，你需要实现两个方法（目前是空的），根据提供的密钥加密或解密一个文本字符串。

一个输入的例子和它产生的输出（这些也存在于提供的单元测试器中）：原始，"纯文本"

（来自https://en.wikipedia.org/wiki/Substitution_cipher）：

> 在密码学中， 替换密码是一种编码方法，通过这种方法，明文的单位被替换为密码文本， 根据一个固定的系统； "单位 "可以是单个字母 （最常见的） ， 一对字母， 三胞胎字母， 上述的混合体， 等等。 接收者通过进行反置换来解读文本。

加密的：

> wgvyxobjotor gohyoyojxuct rjafkuusgyjohbjbojwvbwnxcbtaos nvuoyao xygggxywzjlolytwokato ojjvjtwoabou y jagnroeazxowouaawetxbyohjyh yafrguyaoeaofjyxonxvaoovbhounsaosjyoovayukvaooloojoltfgvyycavb b ooyyjbwokvaafouvyjkvtxfglovyekoobtllovyooa ooafjayoayobtytbtry chytotaagutotay ooyjjxctboxahvetogbaaxooaokxp anpvowaoykaboawyob spcbooourjooaayoooaavuoojjvyoooofooroooytyooxvjooo

解密后：

在密码学中，替换密码是一种编码方法，通过这种方法，明文的单位被替换成符合固定体系的密码文本。单位可以是单个字母，最常见的是一对字母，三组字母，上述的混合物，等等。

形成反置换

在这里，26个英文字母和空格以外的任何符号都被替换为空格，所有文本都被转换为大写字母。尽管这使得这个过程有损失，但它使实现更简单。请在你的实现中也这样做。如果输入的明文不是64个字符的倍数，请假定它以适当数量的空格结束（使用*填充*）。加密字符串的长度必须是64的倍数，而且很可能解密的结果会包含在加密步骤中添加的填充空间。

单位测试仪

还提供了一个单元测试器类。目前它只包含两个测试案例（一个用于加密，一个用于解密）。

想一想你将如何测试本实验室中列出的要求（例如，如何处理特殊情况）。

笔记

- 尽可能地减少代码的重复
- 适当时使用辅助方法
- 使用你到目前为止学到的最佳实践，关于编码风格、注释等。
- 如果你不能完成一个或多个方法，至少要确保它至少返回一些正确类型的值；这将允许测试人员运行，这将使你的代码更容易评估。例如，如果你在`解密`方法上遇到困难，那么请确保该方法返回一些字符串值。

如果你有问题，不要犹豫，在eClass的课程论坛上发布你的问题，或者在你的实验课上询问助教。

提交

在你的项目中找到所有的`java`文件，并通过eClass提交它们的电子版（不要压缩它们）。

如果以小组形式工作，只需提交一份材料，并包括一个包含**小组**成员姓名和学号的**group.txt**文件。截止日期是确定的。

分级

实验的评分将使用*本科院系的通用评分方案*[4] 。我们看代码是否通过了单元测试，是否满足本文件的要求，是否符合代码风格规则。

学术诚信

不允许直接合作（如跨组分享你的工作成果）（可采用抄袭检测软件）。但是，你可以讨论实验室的要求，你采取的方法等。你不得使用任何来自外部的代码，即使它是你自己为其他作业、项目、爱好等写的代码。

[4] https://secretariat-policies.info.yorku.ca/policies/common-grading-scheme-for-undergraduate-faculties/