



Java 语言与系统设计课程实验报告

学生姓名 _____

学生学号 _____

指导教师 _____

专业班级 _____

完成日期 _____

计算机学院

目 录

实验一	2
一、目的与要求.....	2
二、操作环境.....	2
三、实验内容.....	2
四、实验数据.....	8
实验总结	16
参考资料	3

实验一

一、目的与要求

题：用 Java 编程开发“小说人物统计”程序，主要考核字符串处理、IO 操作等。

自行下载自己最喜欢的小说 1 部。存储为文本文档。要求长篇小说，20 万字以上。

事先指定其中 10 个重要人物，考虑他们的姓名、别名等等一系列因素。

(1) 画出每个人在小说中出现位置的可视化图（图可以自行设计）。排序每人出现的次数，并在界面上用柱状图显示。

(2) 统计每个人在小说中出现的篇幅跨度（第一次出现距最后一次出现的篇幅）并排序，并在界面上用柱状图显示，针对每个人物，柱状图中列出该人篇幅跨度起始点到终点位于全文的百分比哪个位置。

(3) 如果两人在较短的一段文字（如 500 字范围，具体可自定）中出现，我们认为两人有联系。有联系的次数越多，关系越紧密。自行设计统计标准，对于 10 个人物：

1) 用表格列出 10 个人物相互关系紧密程度。并对关系紧密程度进行排序，列在界面上。

2) 输入一个人名，列出该人和其他九人关系的紧密程度并排名。

3) 如果多人多次在一段文字中出现，我们认为多人组成小团队。自行设计算法，统计出 10 个人物中，哪些人可能是小团队？

看与实际是否符合？如果不太符合，说明可能的原因。

二、操作环境

硬件： 电脑

软件：Windows, IDEA

三、实验内容

3.1 主类：NovelCharacterStatistics

作用：

该类用于分析小说文本中的角色统计信息，包括每个角色的出现次数、位置、别名映射等。具体功能包括角色出现次数统计、角色位置记录、角色别名映射、角色出现频率排序等。

- 方法 1: processNovel

- 参数: novelText: 小说文本字符串

- 作用: 该方法用于处理小说文本，统计每个角色在文本中的出现次数和位置，并更新相应的数据结构。

- 方法 2: countCharacterOccurrences、

- 参数: text: 小说文本字符串

- character: 角色名称 string

- 作用: 该方法计算指定角色在文本中的总出现次数，包括角色本名和其别名。使用基础方法 calculate 对每个别名进行处理，最终返回总的出现次数。

- 方法 3: findCharacterPositions

- 参数: text: 小说文本字符串

- character: 角色名称

- 作用: 该方法查找指定角色在文本中的所有位置，包括角色本名和其别名。使用基础方法 findPositions 进行处理，最终返回位置列表。

- 方法 4: calculate

- 参数: text: 小说文本字符串

- character: 角色名称

- count: 当前角色出现次数

- 作用: 该方法是一个基础方法，用于计算指定角色在文本中的出现次数。通过在文本中查找指定角色的位置，并在找到一个位置后继续查找下一个位置，累加出现次数。最终返回更新后的出现次数。

- 方法 5: getSortedCharactersByCount

- 参数: 无

- 作用: 该方法返回一个按照角色在文本中出现次数排序的角色列表。使用 Java 的 Map.Entry 和 Comparator 进行排序，将排序后的角色列表返回。

- 方法 6: `getSortedCharactersBySpan`

- 作用：该方法返回一个按照角色在文本中位置范围排序的角色列表。通过计算每个角色在文本中位置的范围（最后一次出现的位置减去第一次出现的位置），使用 `Comparator` 进行排序，将排序后的角色列表返回。

- 方法 7: `readTxtFile`

- 参数：

- `filePath`: 文本文件路径

- 作用：该方法用于读取文本文件内容，返回文件中的所有文本内容字符串。

- 方法 8: `countContact`

- 参数：

- `text`: 小说文本字符串

- `character1`: 第一个角色名称

- `character2`: 第二个角色名称

- `count`: 当前两个角色之间的接触次数

- 作用：该方法计算两个角色在文本中的接触次数。通过在文本中查找两个角色的位置，并判断它们之间的距离是否在规定范围内，累加接触次数。最终返回更新后的接触次数。

- 方法 9: `countContactOccurrences`

- 参数：

- `text`: 小说文本字符串

- `character1`: 第一个角色名称

- `character2`: 第二个角色名称

- 作用：该方法调用基础方法 `countContact`，计算两个角色在文本中的接触次数，包括角色本名和其别名。最终返回两个角色之间的总接触次数。

- 方法 10: `initializeCharacterAliases`

- 作用：该方法用于初始化角色别名映射，直接返回包含全部本名和对应别名列表的映射。

- 方法 11: findPossibleTeams

- 参数:

- text: 小说文本字符串

- windowSize: 窗口大小

- 作用: 该方法用于在文本中查找可能的小团队。通过滑动窗口的方式, 统计每个窗口内各角色的出现次数, 找出频繁出现的人物作为小团队的一部分。最终返回包含小团队的集合。

3.2 类: BarChartEE

作用: BarChartEE 类是用于创建并显示带有渐变色和自定义颜色绘制器的立体条形统计图的 Swing 组件。它接受一个标题和一个包含角色名称及其对应计数的映射, 然后使用 JFreeChart 库生成相应的图表。用户可以通过实例化 BarChartEE 类并传递相应的标题和角色计数映射来轻松地集成此功能到他们的 Swing 应用程序中。

- 方法 1: BarChartEE

- 参数: String title, Map<String, Integer> characterCount

- 作用: 构造方法, 接受图表的标题和包含角色名称及其计数的映射, 然后使用这些信息创建带有渐变色和自定义颜色绘制器的立体条形统计图, 并将其添加到 Swing 组件中显示。

- 方法 2: createDataset

- 参数: Map<String, Integer> characterCount

- 作用: 私有方法, 用于根据角色名称及其计数的映射创建 JFreeChart 数据集 (DefaultCategoryDataset)。遍历映射中的每个条目, 将角色名称作为类别添加到数据集中, 对应的计数作为值添加到 "重要角色" 系列中。

3.3 类: RelationTableEE

作用: RelationTableEE 类用于创建一个包含表格、图片、文本框和按钮的 Swing 窗口, 支持根据用户输入的角色名, 显示该角色与其他角色关系的紧密排序的立体条形统计图。

- 方法 1:

- 名称: RelationTableEE

- 参数: String title, int[][] data

- 作用: 构造方法, 接受窗口标题和一个二维数组作为参数, 用于初始化 RelationTableEE 实例。在构造方法中, 创建包含表格、图片、文本框和按钮的 Swing 窗口, 并设置窗口大小为 800x600。

- 方法 2:

- 名称: createTableModel

- 参数: int[][] data

- 作用: 私有方法, 用于创建并返回一个 DefaultTableModel 对象, 该对象包含表格的标题行和标题列以及与输入二维数组中的数据对应的数据行。

- 方法 3:

- 名称: change

- 参数: int x, int[][] data

- 作用: 私有方法, 接受一个整数 x 和一个二维数组作为参数, 根据 x 指定的角色索引, 生成该角色与其他角色关系的紧密排序的映射, 并返回该映射。在这个方法中, 根据角色索引排除自身, 然后通过流式处理和排序, 生成一个按关系紧密度降序排列的映射。最后, 输出有序映射, 并返回该映射。

3.4 类: ScatterPlot

作用:

ScatterPlot 类用于创建并显示一个散点图, 表示小说中不同角色的出现位置。它接受一个标题和一个包含角色名称及其对应位置列表的映射, 然后使用 JFreeChart 库生成相应的散点图。散点图以角色为横轴, 位置为纵轴, 展示了不同角色在小说中的分布情况。

- 方法 1:

- 名称: ScatterPlot

- 参 数 : String title, Map<String, List<Integer>>

characterPositions

- 作用：构造方法，接受散点图的标题和包含角色名称及其位置列表的映射。在构造方法中，创建并自定义散点图，将角色位置数据添加到图表中，并显示在 Swing 窗口中。

- 方法 2：

- 名称：createDataset

- 参数：Map<String, List<Integer>> characterPositions

- 作用：私有方法，用于创建并返回一个 XYDataset 对象，该对象包含了要在散点图中显示的数据。遍历角色位置的映射，将每个角色的位置添加到 XY 数据集中，用于图表的生成。

- 方法 3：

- 名称：getCharacterNames

- 参数：Map<String, List<Integer>> characterPositions

- 作用：私有方法，用于获取并返回包含所有角色名称的字符串。遍历角色位置的映射，将每个角色的名称添加到字符串中，并返回该字符串用于显示在图表下方。

3.5 类：percentChart

作用：

percentChart 类是一个 JavaFX 应用程序，用于生成并显示人物出现位置散点图、关系紧密程度表、以及人物篇幅跨度示意图。该类继承自 Application 类，并实现了 JavaFX 的 GUI 界面。它依赖 NovelCharacterStatistics 类进行小说分析和数据处理。

该方法因为是 Application 类的子类，所有逻辑都在 start 的方法，下面说明大致编写过程：

- 构造方法和初始化： 该类包含一个构造方法 percentChart()，但在 JavaFX 应用程序中，通常使用 start(Stage primaryStage) 方法作为入口点。构造方法主要用于初始化一些变量，例如 double[][] data，该数组用于存储人物篇幅跨度的百分比数据。

- JavaFX GUI 展示： 在 start(Stage primaryStage) 方法中，创建了一



图 1 第一部：《我的天才女友》海报



图 2 莉拉高光片段

4.2 人物出场次数可视化统计图

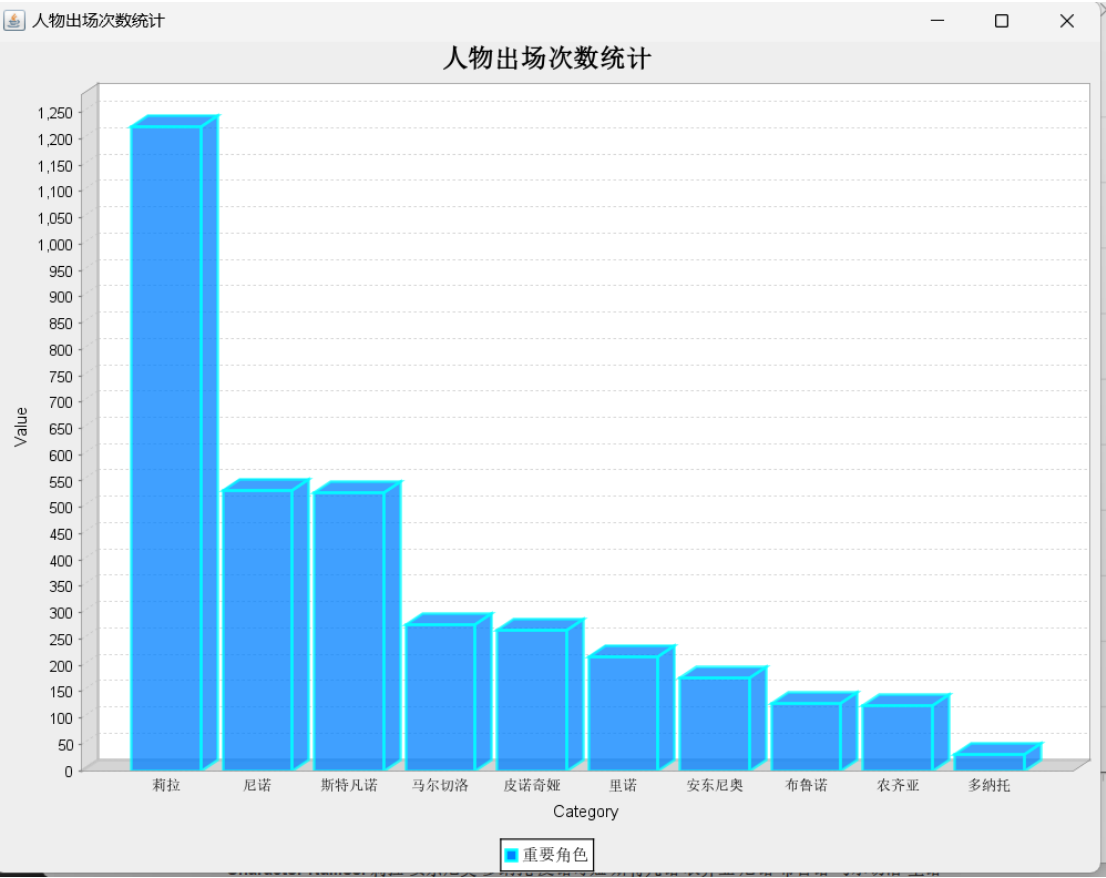


图 3 人物出场次数可视化统计图

说明：该图中，横坐标是角色名，纵坐标是人物出场次数。

4.3 人物出现位置可视化图

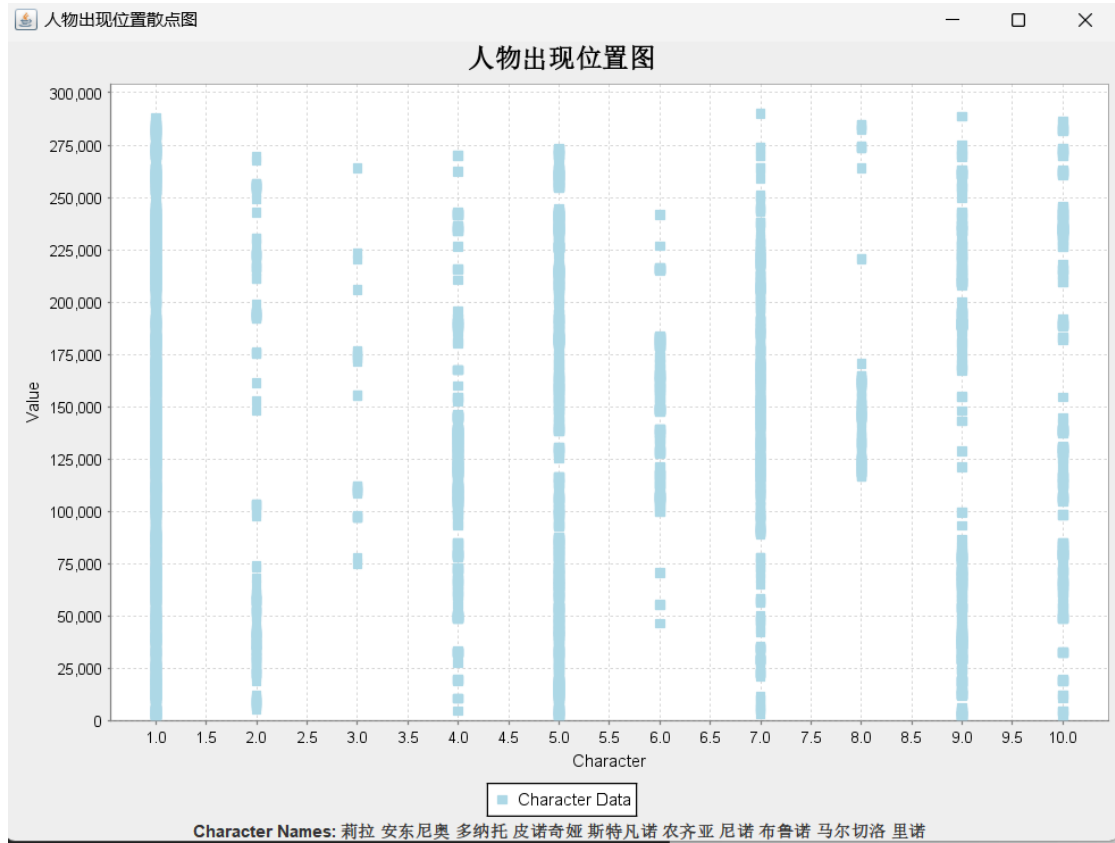


图 4：人物出现位置可视化图

说明：人物出场位置示意图，横坐标无含义，下面给出了图中出场位置的顺序角色名，纵坐标是文章字符串的字数（从 0 到大约 30 万字）；举例说明第一列是主角莉拉，散点过于密集连成一条线，倒数第三列是小配角布鲁诺，可以结合下一个篇幅统计图进行对照查看：布鲁诺出现在 40%的篇幅左右，在一段中密集（和尼诺、埃莱娜、莉拉一起玩耍），并在最后以黑心商人的身份出现几次。

4.4 人物篇幅跨度可视化图

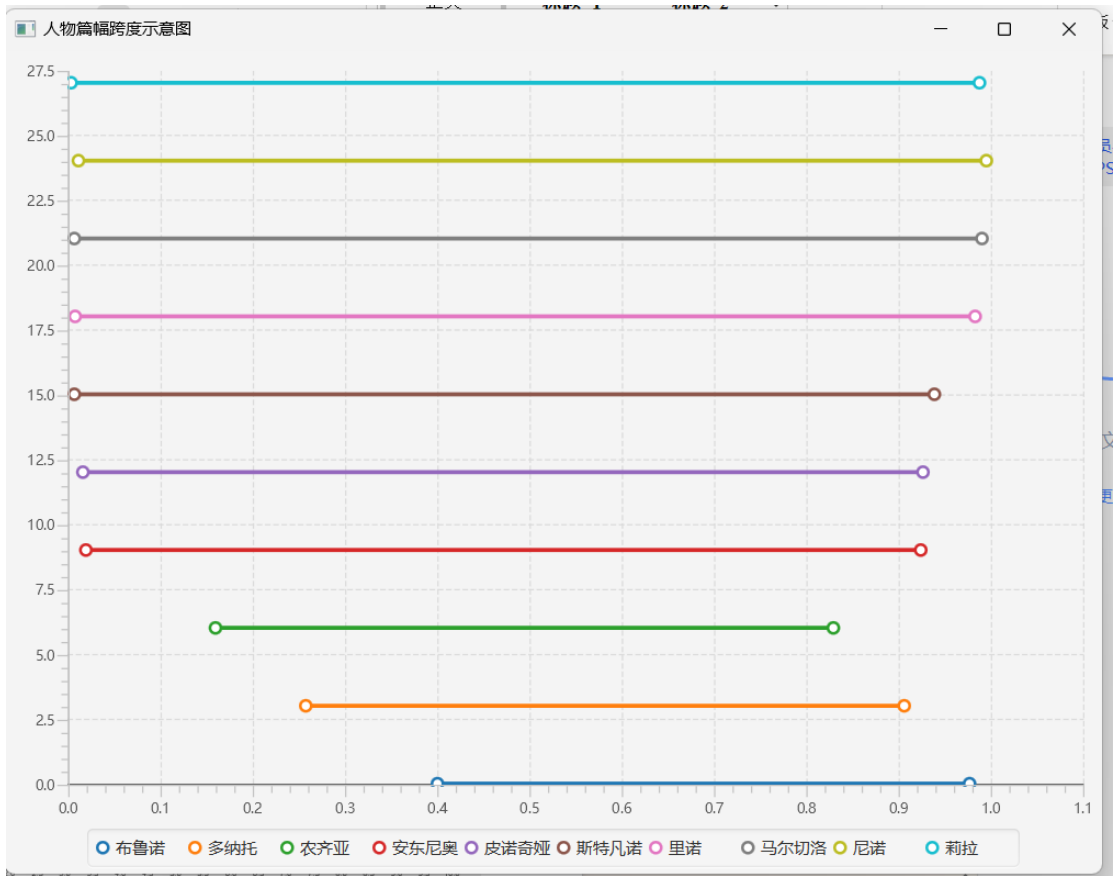


图 5：人物篇幅跨度可视化图

说明：该图是人物出场的篇幅跨度示意图，横坐标的 0-1 表示的是 0%-100%，举例说明，莉拉是主角，其出现篇幅为 0-99%（左右），而布鲁诺只是尼诺的朋友，和主角莉拉和埃莱娜一起玩，在 40%左右的时候第一次出现，一直到最后。

4.5 人物关系紧密表 & 任意人物的关系紧密度排序

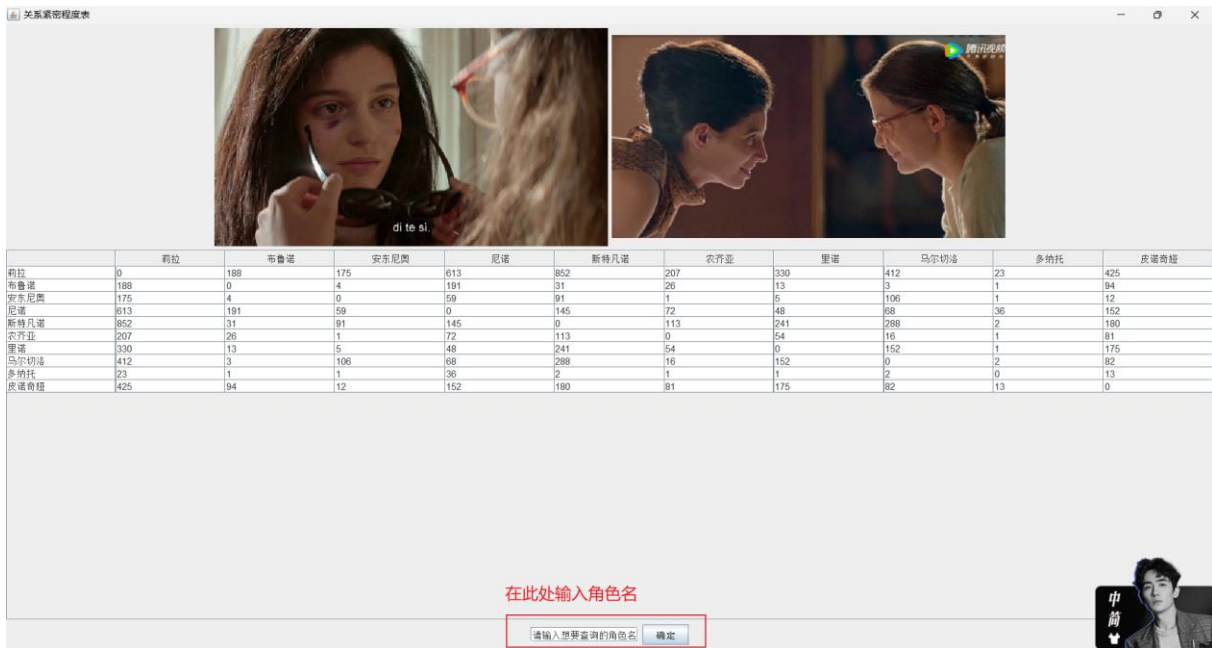


图 6：人物关系紧密表

说明：人物关系紧密程度表。在表格最下面可以输入角色名查询其个人和其他角色的亲密程度条形图，下面以主角莉拉为例：

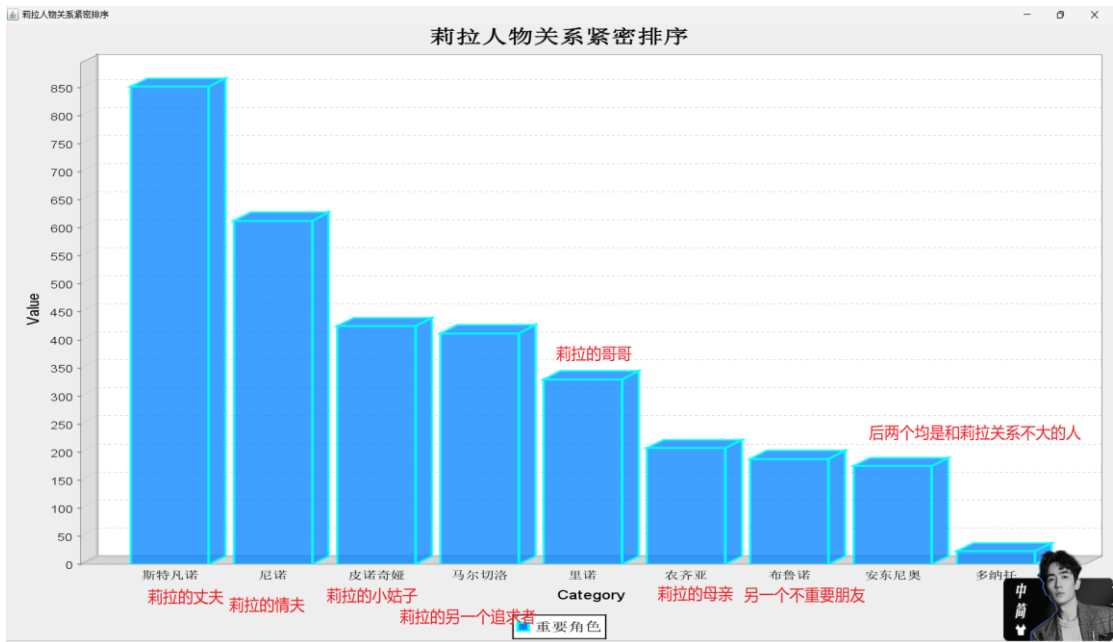
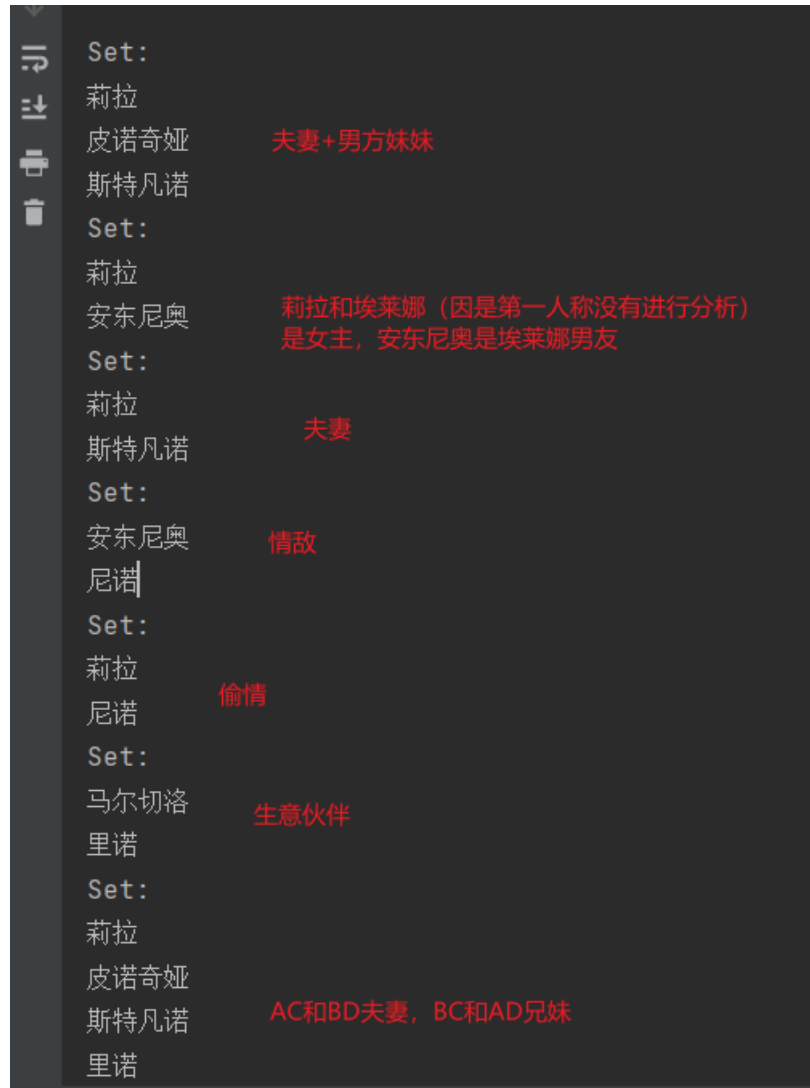


图 7：莉拉的人物关系紧密排序图

说明：其中人物关系我已在图中标出。

4.6 小团队分析功能

小团队分析，通过设置不同的 windowSize 查看小说人物在该大小的字符串中的出现情况，得到不同的结果，现以 200 字符为 windowSize 例，分析小团体结果良好，其中人物关系我已在图中注释写出：



percentage

Set:

莉拉

尼诺 玩伴 (以及埃莱娜) 四人组

布鲁诺

Set:

莉拉

斯特凡诺 夫妻+女方妈妈

农齐亚

Set:

莉拉 母女

农齐亚

Set:

莉拉

皮诺奇娅 妯娌

Set:

皮诺奇娅 兄妹

斯特凡诺

Set:

斯特凡诺 其丈夫和其哥哥 (后结为生意伙伴)

里诺

Set:

斯特凡诺 生意伙伴三人组

马尔切洛

里诺

Set:

莉拉 妯娌

皮诺奇娅

Set:

斯特凡诺

马尔切洛 生意伙伴

Set:

莉拉

布鲁诺 玩伴

Set:

莉拉 追求/被追求

马尔切洛

Set:

尼诺 好兄弟

布鲁诺

五、实验源代码

另外打包，此处不贴代码了

六、实验总结

在本次实验中，我主要学习了如何进行文本分析、字符串处理、文件 IO 以及用 JFreeChart 和 JavaFX 库进行数据可视化和图表展示。以下是我从实验中学到的关键点和总结：

6.1 数据分析与处理：

在本次实验中，我首先了解了如何使用 Java 对文本数据进行处理和分析。通过构建 NovelCharacterStatistics 类，我能够读取小说文本文件，统计人物出现次数、计算人物篇幅跨度以及分析人物关系。

6.2 JavaFX 和 JFreeChart 的使用：

通过实验，我学到了如何使用 JavaFX 构建图形用户界面（GUI）。我创建了一个 JavaFX 应用程序 percentChart，用于展示小说分析的结果。在这个过程中，我学到了如何使用 JavaFX 的图表组件，如散点图、条形图和折线图。

6.3 图表的生成与展示：

实验中，我使用了 JFreeChart 和 JavaFX 的图表组件，分别生成了条形图、散点图和折线图。对于每种图表，我都学到了如何设置标题、坐标轴标签、图例等属性，以及如何自定义图表的外观和样式。

6.4 图表的交互性与用户友好：

通过 JavaFX，我实现了图表的交互性。用户可以通过窗口中的图表进行交互，查看小说中人物的紧密关系示意图。这为用户提供了一种直观的方式来理解和分析小说中的数据。

参考资料

无