Python**选题**-**笑话生成**

# 作品说明

本作品为“笑话生成与管理系统”，基于 Python 语言开发，实现了自动生成不同风格的笑话、保存喜欢的笑话、统计生成数据等功能。用户可以选择预设风格（1234），也可以通过“custom”自定义选择风格

(如：funny,random等)，满足不同用户的娱乐和学习需求。该系统适用于编程初学者练习 Python 基础、字符串处理、文件操作等知识，也可作为趣味娱乐工具在日常生活中使用。

# 实现过程

1. **功能设计**

支持多种笑话风格（如1234、funny、tech、office、random）。用户可选择风格并生成笑话，支持自定义风格选择。

支持将喜欢的笑话保存到本地文件。

提供笑话数量、总字数、平均字数等统计信息。可查看历史保存的笑话。

1. **主要实现细节**

使用字典 [styles](vscode-file://vscode-app/d:/Program%20Files%20(x86)/vscode/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) 存储不同风格的笑话列表。

styles = {

"1234": [

"为什么程序员喜欢用暗模式？因为亮模式会让他们的代码看起来像是被太阳

晒过。",

"程序员的梦想是什么？代码一次通过，永不改动！",

"为什么程序员总是很冷？因为他们总是待在低温的服务器房间里。", "程序员的座右铭是什么？'没有 bug 的代码是不存在的！'"

],

"custom": ["这是一个自定义笑话示例。"], "funny": [

"为什么计算机喜欢吃饼干？因为它们有很多字节！", "程序员的宠物是什么？Bug。",

"为什么程序员总是很累？因为他们一直在跑循环！"

],

"tech": [

"为什么程序员喜欢用 Linux？因为他们喜欢自由。", "为什么云计算很受欢迎？因为它让人感觉轻松自在。", "为什么硬盘总是很忙？因为它们一直在转圈圈。"

],

"office": [

"为什么老板喜欢 Excel？因为它能让他们看起来很专业。", "为什么会议总是很长？因为没人愿意结束。",

"为什么打印机总是坏？因为它们讨厌工作。"

],

"random": [

"为什么鸡过马路？为了去另一边。",

"为什么鱼不喜欢说话？因为它们总是沉默。", "为什么树喜欢站着？因为它们不喜欢坐下。"

]

}

[generate\_joke](vscode-file://vscode-app/d:/Program%20Files%20(x86)/vscode/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) 方法根据用户输入风格生成笑话，若为 custom，则进一步输入具体风格。

def generate\_joke(self, style="1234", word\_limit=50): styles = {

"1234": [

"为什么程序员喜欢用暗模式？因为亮模式会让他们的代码看起来像是被太阳

晒过。",

"程序员的梦想是什么？代码一次通过，永不改动！",

"为什么程序员总是很冷？因为他们总是待在低温的服务器房间里。", "程序员的座右铭是什么？'没有 bug 的代码是不存在的！'"

],

"custom": ["这是一个自定义笑话示例。"], "funny": [

"为什么计算机喜欢吃饼干？因为它们有很多字节！", "程序员的宠物是什么？Bug。",

"为什么程序员总是很累？因为他们一直在跑循环！"

],

"tech": [

"为什么程序员喜欢用 Linux？因为他们喜欢自由。", "为什么云计算很受欢迎？因为它让人感觉轻松自在。", "为什么硬盘总是很忙？因为它们一直在转圈圈。"

],

"office": [

"为什么老板喜欢 Excel？因为它能让他们看起来很专业。", "为什么会议总是很长？因为没人愿意结束。",

"为什么打印机总是坏？因为它们讨厌工作。"

],

"random": [

"为什么鸡过马路？为了去另一边。",

"为什么鱼不喜欢说话？因为它们总是沉默。", "为什么树喜欢站着？因为它们不喜欢坐下。"

]

}

# 如果用户输入 custom，则进一步选择具体风格 if style == "custom":

sub\_style = input("请输入自定义笑话风格 (例如: funny, tech,

office, random): ")

selected\_style = styles.get(sub\_style, ["该自定义风格不存在，请

检查输入！"])

else:

selected\_style = styles.get(style, styles["custom"])

joke = random.choice(selected\_style) if len(joke) > word\_limit:

joke = joke[:word\_limit] + "..." self.joke\_count += 1 self.total\_word\_count += len(joke) return joke

[save\_joke](vscode-file://vscode-app/d:/Program%20Files%20(x86)/vscode/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) 方法将笑话保存到内存和本地文件。

def save\_joke(self, joke):

self.saved\_jokes.append(joke)

timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S") with open("saved\_jokes.txt", "a", encoding="utf-8") as file:

file.write(f"{timestamp} - {joke}\n")

[get\_statistics](vscode-file://vscode-app/d:/Program%20Files%20(x86)/vscode/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) 方法统计生成笑话的数量和字数。

def get\_statistics(self):

average\_word\_count = self.total\_word\_count / self.joke\_count if self.joke\_count > 0 else 0

return {

"joke\_count": self.joke\_count, "total\_word\_count": self.total\_word\_count, "average\_word\_count": average\_word\_count

}

[show\_saved\_jokes\_from\_file](vscode-file://vscode-app/d:/Program%20Files%20(x86)/vscode/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) 方法读取并展示已保存的笑话。

def show\_saved\_jokes\_from\_file(): try:

with open("saved\_jokes.txt", "r", encoding="utf-8") as file: jokes = file.readlines()

if not jokes:

print("暂无已保存的笑话。") else:

print("历史保存的笑话：") for joke in jokes:

print(joke.strip()) except FileNotFoundError:

print("还没有保存过笑话。")

主循环实现用户交互，输入选择、风格、字数限制等。

if name == " main ": generator = JokeGenerator() while True:

print("\n1. 生成笑话")

print("2. 查看已保存的笑话") print("3. 退出")

choice = input("请选择操作（1/2/3）：")

if choice == "1":

print("请选择笑话风格：1234 或者输入 custom 自定义风格")

style = input("风格: ") try:

word\_limit = int(input("请输入笑话字数限制: "))

except ValueError:

print("字数限制必须是数字，请重新输入!") continue

joke = generator.generate\_joke(style, word\_limit) print(f"生成的笑话: {joke}")

save\_option = input("是否收藏并保存笑话? (y/n): ") if save\_option.lower() == "y":

generator.save\_joke(joke) print("笑话已保存!")

stats = generator.get\_statistics()

print(f"目前已创作笑话数量: {stats['joke\_count']}, 总字数:

{stats['total\_word\_count']}, 平均字数: {stats['average\_word\_count']:.2f}") elif choice == "2":

show\_saved\_jokes\_from\_file() elif choice == "3":

break else:

print("无效选择，请重新输入。")

print("已保存的笑话:")

for saved\_joke in generator.display\_saved\_jokes(): print(saved\_joke)

1. **用到的知识点**

Python 字典、列表、字符串处理文件读写操作

类与对象、方法定义

用户输入与异常处理

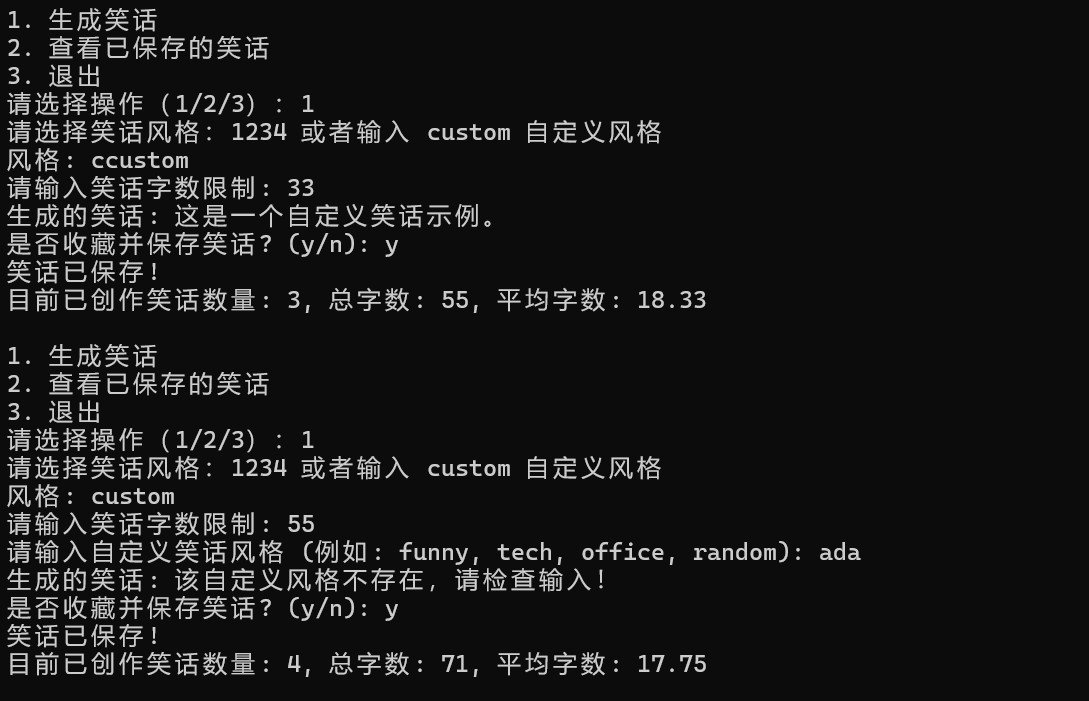
随机选择（random.choice）

# 结果展示

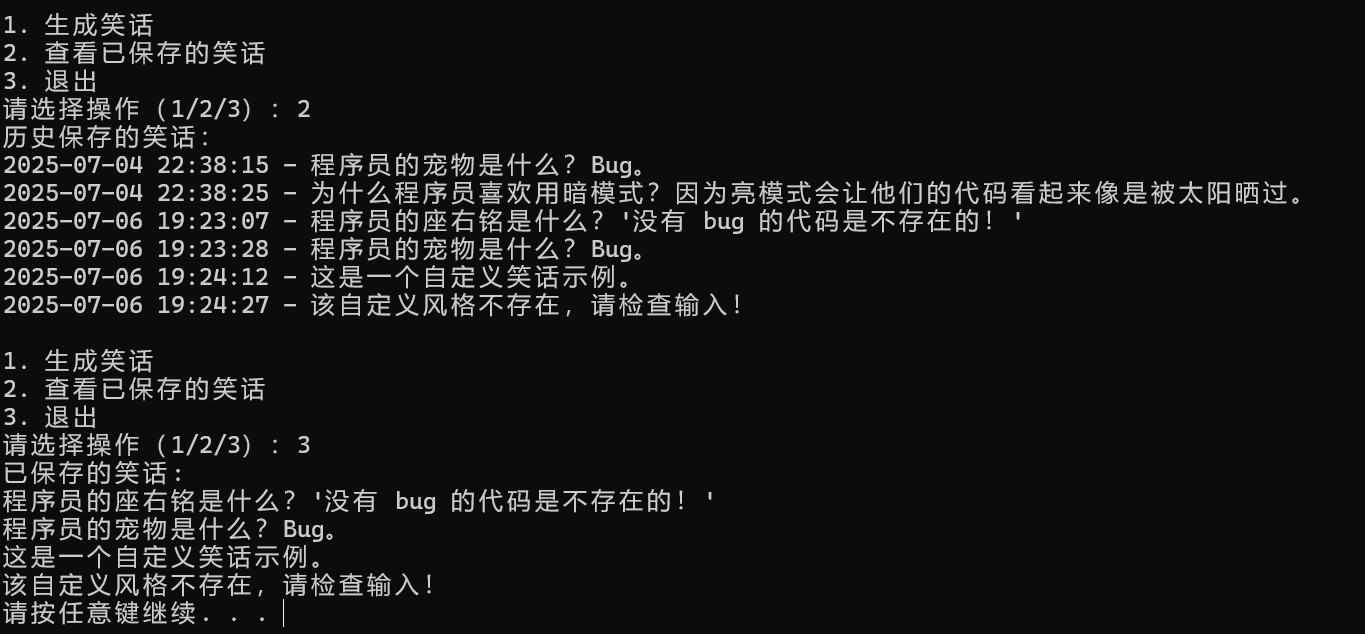
1. **生成笑话界面：**



1. **保存笑话与统计信息：**



1. **查看历史笑话：**



# 总结展望

本作品实现了一个简单实用的笑话生成与管理系统，涵盖了 Python 基础编程的多个知识点，具有一定的趣味性和实用性。

**不足与改进方向：**

当前笑话内容较少，可进一步丰富笑话库，支持在线获取笑话。

用户自定义风格仅限于已有风格，可扩展为用户自定义添加笑话内容。界面为命令行，后续可尝试开发图形界面或网页端，提升用户体验。 可增加笑话点赞、分享等互动功能，增强趣味性和传播性。