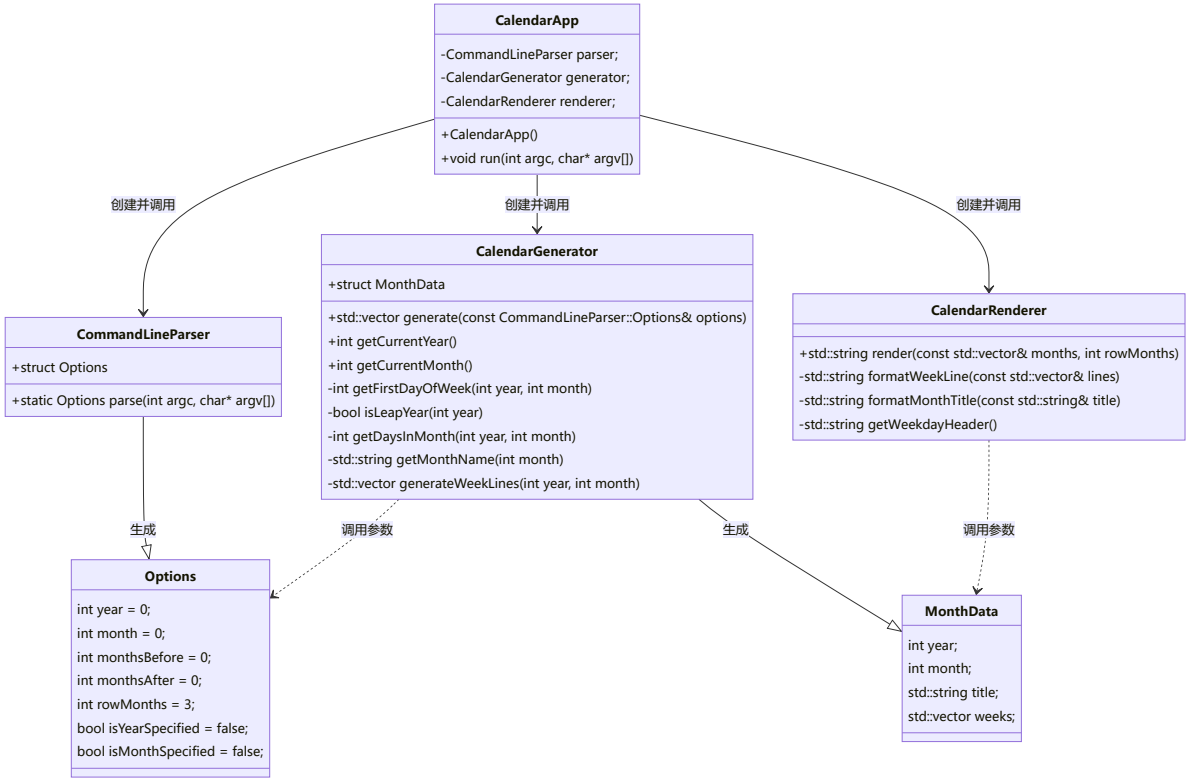
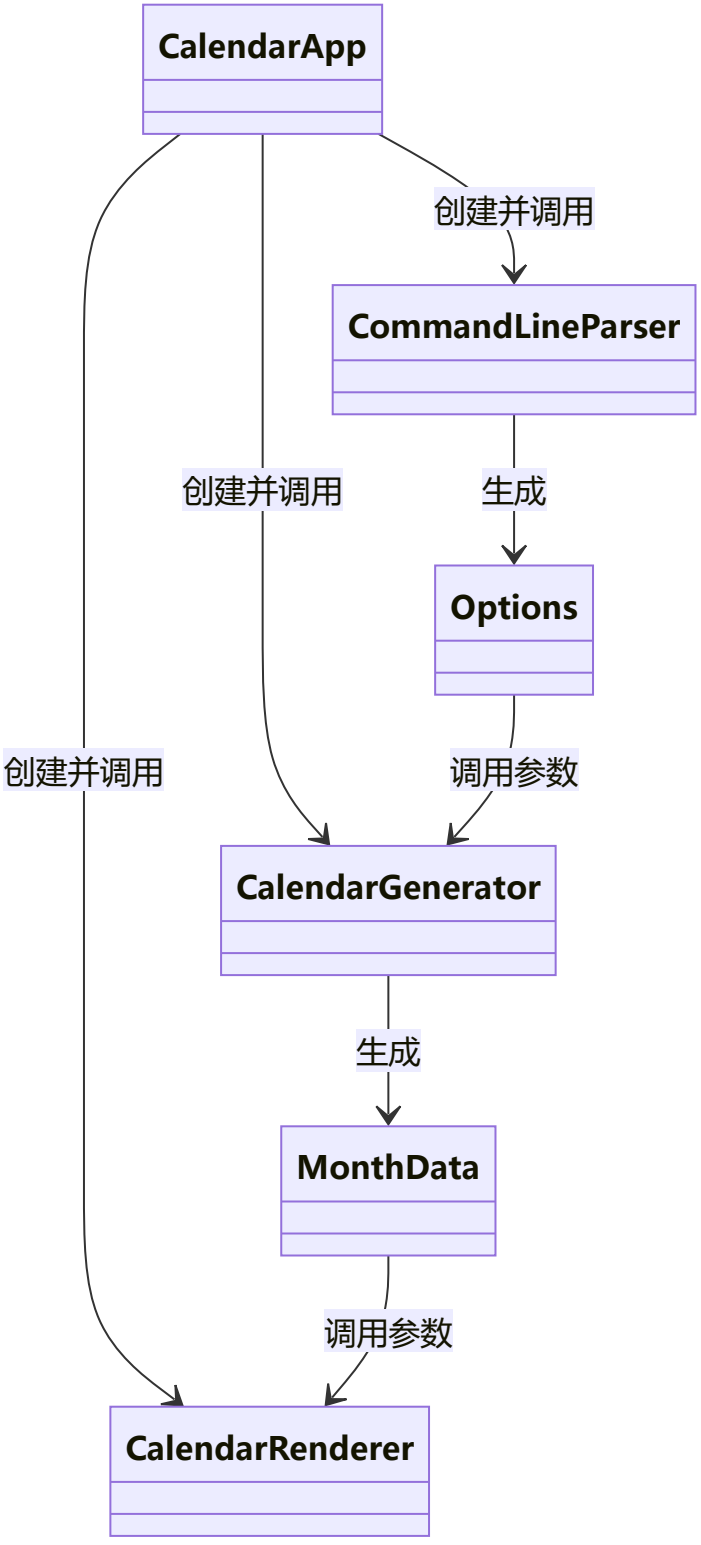
设计说明

23307110426 马颢宸

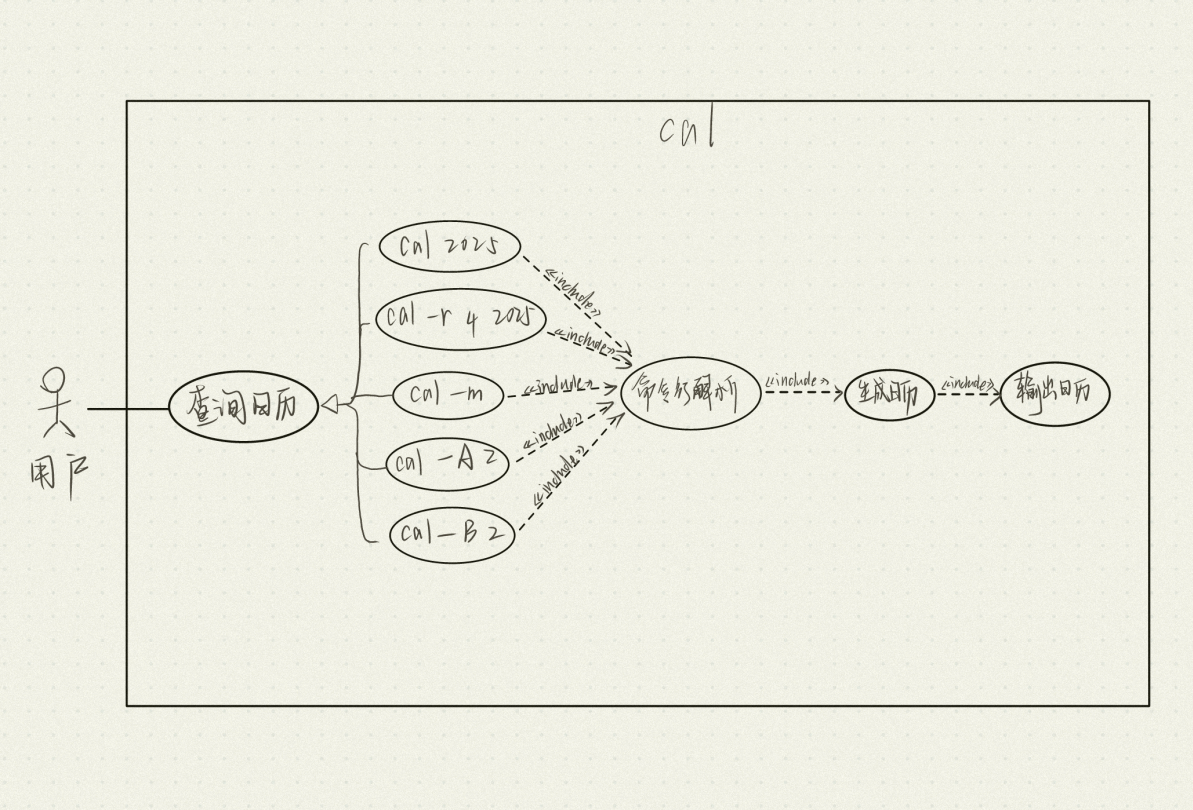
一、类图



二、类关系图


CalendarApp
创建并调用
CommandLineParser
创建并调用
生成
Options
创建并调用
调用参数
CalendarGenerator
生成
MonthData
调用参数
CalendarRenderer

三、用例图



四、测试结果与样例

- 测试结果

```
D:\Archive\subject\two two\OOP\lab\lab7\build>"d:\Archive\subject\two two\OOP\lab\lab7\build\cal_test.exe"
Running unit tests...
Running integration tests...
Test passed: 9/9
```

- 测试样例

```
D:\Archive\subject\two two\OOP\lab\lab7\build>cal 2025
    January 2025          February 2025          March 2025
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                      1                      1
 5  6  7  8  9 10 11    2  3  4  5  6  7  8    2  3  4  5  6  7  8
12 13 14 15 16 17 18    9 10 11 12 13 14 15    9 10 11 12 13 14 15
19 20 21 22 23 24 25   16 17 18 19 20 21 22   16 17 18 19 20 21 22
26 27 28 29 30 31      23 24 25 26 27 28      23 24 25 26 27 28 29
                                              30 31

     April 2025            May 2025              June 2025
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
       1  2  3  4  5                1  2  3    1  2  3  4  5  6  7
 6  7  8  9 10 11 12    4  5  6  7  8  9 10    8  9 10 11 12 13 14
13 14 15 16 17 18 19   11 12 13 14 15 16 17   15 16 17 18 19 20 21
20 21 22 23 24 25 26   18 19 20 21 22 23 24   22 23 24 25 26 27 28
27 28 29 30            25 26 27 28 29 30 31   29 30


      July 2025            August 2025          September 2025
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
       1  2  3  4  5                   1  2       1  2  3  4  5  6
 6  7  8  9 10 11 12    3  4  5  6  7  8  9    7  8  9 10 11 12 13
13 14 15 16 17 18 19   10 11 12 13 14 15 16   14 15 16 17 18 19 20
20 21 22 23 24 25 26   17 18 19 20 21 22 23   21 22 23 24 25 26 27
27 28 29 30 31         24 25 26 27 28 29 30   28 29 30
                       31

    October 2025          November 2025          December 2025
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                      1       1  2  3  4  5  6
 5  6  7  8  9 10 11    2  3  4  5  6  7  8    7  8  9 10 11 12 13
12 13 14 15 16 17 18    9 10 11 12 13 14 15   14 15 16 17 18 19 20
19 20 21 22 23 24 25   16 17 18 19 20 21 22   21 22 23 24 25 26 27
26 27 28 29 30 31      23 24 25 26 27 28 29   28 29 30 31
                       30
```

```
D:\Archive\subject\two two\OOP\lab\lab7\build>cal -r 4 2025
    January 2025          February 2025          March 2025           April 2025
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                      1                      1          1  2  3  4  5
 5  6  7  8  9 10 11    2  3  4  5  6  7  8    2  3  4  5  6  7  8    6  7  8  9 10 11 12
12 13 14 15 16 17 18    9 10 11 12 13 14 15    9 10 11 12 13 14 15   13 14 15 16 17 18 19
19 20 21 22 23 24 25   16 17 18 19 20 21 22   16 17 18 19 20 21 22   20 21 22 23 24 25 26
26 27 28 29 30 31      23 24 25 26 27 28      23 24 25 26 27 28 29   27 28 29 30
                                              30 31

      May 2025              June 2025              July 2025            August 2025
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
             1  2  3    1  2  3  4  5  6  7          1  2  3  4  5                   1  2
 4  5  6  7  8  9 10    8  9 10 11 12 13 14    6  7  8  9 10 11 12    3  4  5  6  7  8  9
11 12 13 14 15 16 17   15 16 17 18 19 20 21   13 14 15 16 17 18 19   10 11 12 13 14 15 16
18 19 20 21 22 23 24   22 23 24 25 26 27 28   20 21 22 23 24 25 26   17 18 19 20 21 22 23
25 26 27 28 29 30 31   29 30                  27 28 29 30 31         24 25 26 27 28 29 30
                                                                     31

   September 2025         October 2025          November 2025         December 2025
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6             1  2  3  4                      1       1  2  3  4  5  6
 7  8  9 10 11 12 13    5  6  7  8  9 10 11    2  3  4  5  6  7  8    7  8  9 10 11 12 13
14 15 16 17 18 19 20   12 13 14 15 16 17 18    9 10 11 12 13 14 15   14 15 16 17 18 19 20
21 22 23 24 25 26 27   19 20 21 22 23 24 25   16 17 18 19 20 21 22   21 22 23 24 25 26 27
28 29 30               26 27 28 29 30 31      23 24 25 26 27 28 29   28 29 30 31
                                              30
```

```
D:\Archive\subject\two two\OOP\lab\lab7\build>cal -m 5
      May 2025
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

```
D:\Archive\subject\two two\OOP\lab\lab7\build>cal -A 2
      May 2025              June 2025              July 2025
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
             1  2  3    1  2  3  4  5  6  7          1  2  3  4  5
 4  5  6  7  8  9 10    8  9 10 11 12 13 14    6  7  8  9 10 11 12
11 12 13 14 15 16 17   15 16 17 18 19 20 21   13 14 15 16 17 18 19
18 19 20 21 22 23 24   22 23 24 25 26 27 28   20 21 22 23 24 25 26
25 26 27 28 29 30 31   29 30                  27 28 29 30 31
```

```
D:\Archive\subject\two two\OOP\lab\lab7\build>cal -B 2
     March 2025             April 2025              May 2025
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                   1          1  2  3  4  5                   1  2  3
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    4  5  6  7  8  9 10
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   11 12 13 14 15 16 17
16 17 18 19 20 21 22   20 21 22 23 24 25 26   18 19 20 21 22 23 24
23 24 25 26 27 28 29   27 28 29 30            25 26 27 28 29 30 31
30 31
```

五、AI设计批评意见

- 测试代码编写忽略实际程序行为

```
bool test_CommandLineParser_parse_months_after() {
    const char* argv[] = {"cal", "2025", "5", "-A", "2"};
    auto opts = CommandLineParser::parse(5, const_cast<char**>(argv));
    ASSERT_TRUE(opts.isYearSpecified);
    ASSERT_EQ(opts.year, 2025);
    ASSERT_TRUE(opts.isMonthSpecified);
    ASSERT_EQ(opts.month, 5);
    ASSERT_EQ(opts.monthsAfter, 2);
    return true;
}

bool test_CommandLineParser_parse_months_before() {
    const char* argv[] = {"cal", "2025", "5", "-B", "2"};
    auto opts = CommandLineParser::parse(5, const_cast<char**>(argv));
    ASSERT_TRUE(opts.isYearSpecified);
    ASSERT_EQ(opts.year, 2025);
    ASSERT_TRUE(opts.isMonthSpecified);
    ASSERT_EQ(opts.month, 5);
    ASSERT_EQ(opts.monthsBefore, 2);
    return true;
}
```

在AI工具编写的实际代码中，出现了测试`cal 2025 5 -A 2`、`cal 2025 5 -B 2`的语句。然而在通过在linux系统中运行这两条命令可知，此为非法命令：

```
root@Ranxiaoxiao:~# cal 2025 5 -A 2
cal: 2025 is neither a month number (1..12) nor a name
root@Ranxiaoxiao:~# cal 2025 5 -B 2
cal: 2025 is neither a month number (1..12) nor a name
```

除此以外，AI工具在测试代码中还编写了例如`cal 2025 5`、`cal -d 2025 5`等非法命令

总而言之，在编写相关代码时，AI工具错误理解了程序实际运行时应该解析的正确指令，并试图验证通过错误指令。

- 忽视了对错误指令的处理与反馈

以错误指令`cal 2025 5`为例，在AI工具编写程序中，运行该指令，程序会返回公元5年的日历

```
D:\Archive\subject\two two\OOP\lab\lab7\build>cal 2025 5
      January 5              February 5               March 5
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                    1             1  2  3  4  5             1  2  3  4  5
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    6  7  8  9 10 11 12
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   13 14 15 16 17 18 19
16 17 18 19 20 21 22   20 21 22 23 24 25 26   20 21 22 23 24 25 26
23 24 25 26 27 28 29   27 28                 27 28 29 30 31
30 31

       April 5                 May 5                  June 5
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                 1  2    1  2  3  4  5  6  7             1  2  3  4
 3  4  5  6  7  8  9    8  9 10 11 12 13 14    5  6  7  8  9 10 11
10 11 12 13 14 15 16   15 16 17 18 19 20 21   12 13 14 15 16 17 18
17 18 19 20 21 22 23   22 23 24 25 26 27 28   19 20 21 22 23 24 25
24 25 26 27 28 29 30   29 30 31              26 27 28 29 30


       July 5                 August 5               September 5
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                 1  2       1  2  3  4  5  6                1  2  3
 3  4  5  6  7  8  9    7  8  9 10 11 12 13    4  5  6  7  8  9 10
10 11 12 13 14 15 16   14 15 16 17 18 19 20   11 12 13 14 15 16 17
17 18 19 20 21 22 23   21 22 23 24 25 26 27   18 19 20 21 22 23 24
24 25 26 27 28 29 30   28 29 30 31            25 26 27 28 29 30
31

      October 5              November 5              December 5
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                    1             1  2  3  4  5                1  2  3
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    4  5  6  7  8  9 10
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   11 12 13 14 15 16 17
16 17 18 19 20 21 22   20 21 22 23 24 25 26   18 19 20 21 22 23 24
23 24 25 26 27 28 29   27 28 29 30            25 26 27 28 29 30 31
30 31
```

而在linux系统中运行该指令，程序会返回错误信息

```
root@Ranxiaoxiao:~# cal 2025 5
cal: 2025 is neither a month number (1..12) nor a name
```

可以看出，AI编写工具忽视了对错误指令的处理及信息反馈机制。这也间接导致了上一问题（试图验证通过错误指令）的产生。

- 忽视了真实历史的日历信息

在测试对比`cal 1`指令时发现，linux系统返回的日历与AI工具编写代码返回的日历有明显区别。但运行`cal 2025`指令，二者相同且符合现实信息。

仔细检查后发现，在**1752年9月**的日历数据中，linux系统缺少了13天的数据信息，即**9/2之后的下一天为9/14**

```
root@Ranxiaoxiao:~# cal 1752
                              1752

      January               February                March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
          1  2  3  4                     1   1  2  3  4  5  6  7
 5  6  7  8  9 10 11   2  3  4  5  6  7  8   8  9 10 11 12 13 14
12 13 14 15 16 17 18   9 10 11 12 13 14 15  15 16 17 18 19 20 21
19 20 21 22 23 24 25  16 17 18 19 20 21 22  22 23 24 25 26 27 28
26 27 28 29 30 31     23 24 25 26 27 28 29  29 30 31
                      30

       April                  May                   June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
          1  2  3  4                  1  2      1  2  3  4  5  6
 5  6  7  8  9 10 11   3  4  5  6  7  8  9   7  8  9 10 11 12 13
12 13 14 15 16 17 18  10 11 12 13 14 15 16  14 15 16 17 18 19 20
19 20 21 22 23 24 25  17 18 19 20 21 22 23  21 22 23 24 25 26 27
26 27 28 29 30        24 25 26 27 28 29 30  28 29 30
                      31

        July                 August              September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
          1  2  3  4                     1         1  2 14 15 16
 5  6  7  8  9 10 11   2  3  4  5  6  7  8  17 18 19 20 21 22 23
12 13 14 15 16 17 18   9 10 11 12 13 14 15  24 25 26 27 28 29 30
19 20 21 22 23 24 25  16 17 18 19 20 21 22
26 27 28 29 30 31     23 24 25 26 27 28 29
                      30 31

      October               November              December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7            1  2  3  4                  1  2
 8  9 10 11 12 13 14   5  6  7  8  9 10 11   3  4  5  6  7  8  9
15 16 17 18 19 20 21  12 13 14 15 16 17 18  10 11 12 13 14 15 16
22 23 24 25 26 27 28  19 20 21 22 23 24 25  17 18 19 20 21 22 23
29 30 31              26 27 28 29 30        24 25 26 27 28 29 30
                                            31
```

而AI工具编写代码中，对应**日期正常递增**

```
D:\Archive\subject\two two\OOP\lab\lab7\build>cal 1752
    January 1752           February 1752           March 1752
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1             1  2  3  4  5             1  2  3  4
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    5  6  7  8  9 10 11
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   12 13 14 15 16 17 18
16 17 18 19 20 21 22   20 21 22 23 24 25 26   19 20 21 22 23 24 25
23 24 25 26 27 28 29   27 28 29              26 27 28 29 30 31
30 31

     April 1752             May 1752              June 1752
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1       1  2  3  4  5  6                1  2  3
 2  3  4  5  6  7  8    7  8  9 10 11 12 13    4  5  6  7  8  9 10
 9 10 11 12 13 14 15   14 15 16 17 18 19 20   11 12 13 14 15 16 17
16 17 18 19 20 21 22   21 22 23 24 25 26 27   18 19 20 21 22 23 24
23 24 25 26 27 28 29   28 29 30 31            25 26 27 28 29 30
30

      July 1752              August 1752           September 1752
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1             1  2  3  4  5                   1  2
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    3  4  5  6  7  8  9
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   10 11 12 13 14 15 16
16 17 18 19 20 21 22   20 21 22 23 24 25 26   17 18 19 20 21 22 23
23 24 25 26 27 28 29   27 28 29 30 31         24 25 26 27 28 29 30
30 31

    October 1752           November 1752          December 1752
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7             1  2  3  4                   1  2
 8  9 10 11 12 13 14    5  6  7  8  9 10 11    3  4  5  6  7  8  9
15 16 17 18 19 20 21   12 13 14 15 16 17 18   10 11 12 13 14 15 16
22 23 24 25 26 27 28   19 20 21 22 23 24 25   17 18 19 20 21 22 23
29 30 31               26 27 28 29 30         24 25 26 27 28 29 30
                                              31
```

查询相关信息后得知，这是由于**日历变换的相关历史**导致，即

1. 历史背景：儒略历 → 格里高利历

- **儒略历（Julian Calendar）**：
  由罗马皇帝儒略·凯撒于公元前 45 年推行，每年 365 天，每 4 年闰 1 天（平均年长 365.25 天）。
  问题：实际太阳年约为 365.2422 天，儒略历每年多算约 11 分钟，长期累积导致日期与天文现象（如春分）逐渐错位。
- **格里高利历（Gregorian Calendar）**：
  1582 年由教皇格里高利十三世推行，修正了闰年规则（能被 100 整除但不能被 400 整除的年份不闰年），平均年长 365.2425 天，更接近真实太阳年。

---

2. 1752 年的历法切换（英国及北美殖民地）

- **切换时间**：
  英国及其殖民地（包括后来的美国）在 **1752 年 9 月** 正式从儒略历切换到格里高利历。
- **日期调整**：
  为了纠正儒略历累积的误差，**1752 年 9 月 2 日（星期三）的次日被定为 9 月 14 日（星期四）**，跳过了 11 天。
  - 这是为了与欧洲大陆（如意大利、西班牙等已使用格里高利历的国家）同步。

3. 为什么 `cal 1752` 显示缺失 11 天?

- Linux 的 `cal` 命令基于历史事实实现，会真实反映 1752 年的历法切换：

```bash
$ cal 1752
       September 1752
Su Mo Tu We Th Fr Sa
       1  2 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

  - 9 月 2 日之后直接跳到 14 日，中间 3-13 日不存在。

---

4. 其他国家/地区的切换时间不同

- **天主教国家**：如意大利、西班牙在 1582 年 10 月首次切换（跳过 10 天）。
- **东正教国家**：如俄罗斯在 1918 年才切换（跳过 13 天）。
- **Linux 的 `cal` 命令**默认以英美历史为基准，因此仅 1752 年 9 月有缺失。

而AI工具在编写相关指令时，未考虑实际历史的日历变换信息，从而在1752年的数据和真实数据有所出入，进而导致1752年及之前 的所有数据都有变动。

- 排版调整不美观

  在最初的AI设计版本中，输出的日历数据在排版上并不美观。

  日期与星期之间无法对齐，且空隙逐列递增；月份title无法正确居中，当一排显示多个月份时，title逐列偏右

  多次询问AI并要求其对齐排版后，仍无法正确输出美观的格式，最终还是依靠手动调整排版。

  总而言之，AI工具虽能写出正确实现程序功能的代码，但对于终端输出的美观与整洁却差强人意。