



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ ИУ _____

КАФЕДРА _____ Программная инженерия _____

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ

Студент Воякин Алексей Янович

Группа ИУ7-14Б

Тип практики Учебная

Студент _____
_____ *подпись, дата* _____ *фамилия, и.о.*

Руководитель практики _____
_____ *подпись, дата* _____ *фамилия, и.о.*

Оценка _____

Оглавление

1. Условие задачи (стр. 3)
2. Схема программы (стр. 4-7)
3. Описание программы (стр. 8-9)
4. Текст программы (стр. 9-13)
5. Заключение (стр. 13)
6. Список литературы (стр. 14)

Условие задачи:

Стандартный календарь был принят Юлием Цезарем в 45 году до н.э. Каждый год имел 265 дней, и каждый четвёртый год имел лишний день – 29 февраля. Однако этот календарь не очень точно соответствовал реальному солнечному году, и со временем это стало заметно. В 1582 году папа Григорий постановил, что новый стиль календаря вступит в силу. Вековые года (1800, 1900, 2000) были високосными только тогда, когда они делились без остатка на 400. К тому же текущий год нуждался в коррекции, чтобы соответствовать сезонам. Этот новый календарь и необходимое исправление были приняты всеми католическими странами. Тогда после четверга 4 октября 1682 года была пятница 15 октября 1582. Британцы и американцы приняли этот календарь в 1752 году, тогда за средой 2 сентября шёл четверг 14 сентября. Россия не изменяла календарь до 1918 года, а Греция ждала до 1923. Таким образом, был долгий период времени, когда история записывалась в двух различных стилях.

Написать программу, которая будет читать дату, определять в каком она стиле и переводить в другой стиль.

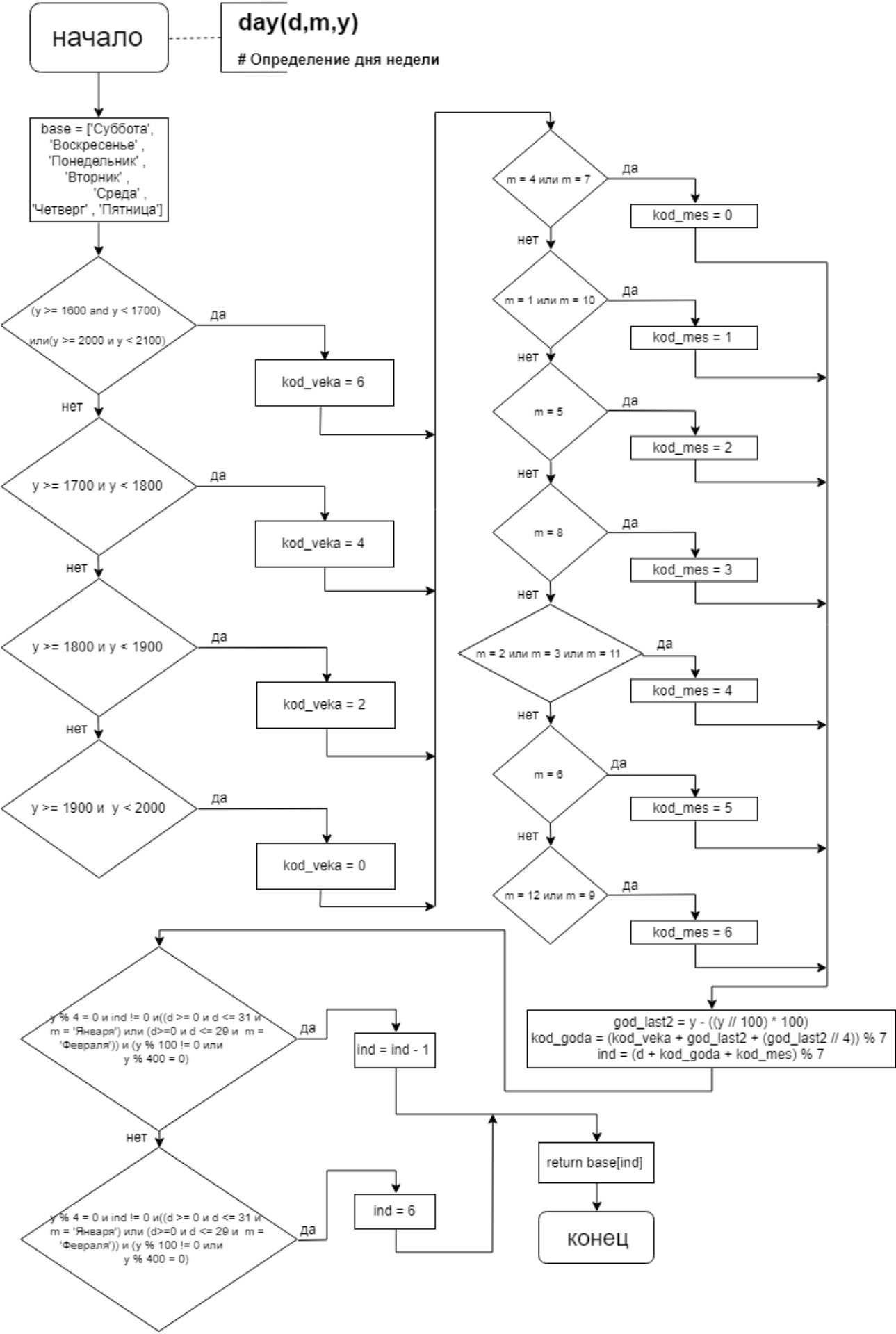
Ввод

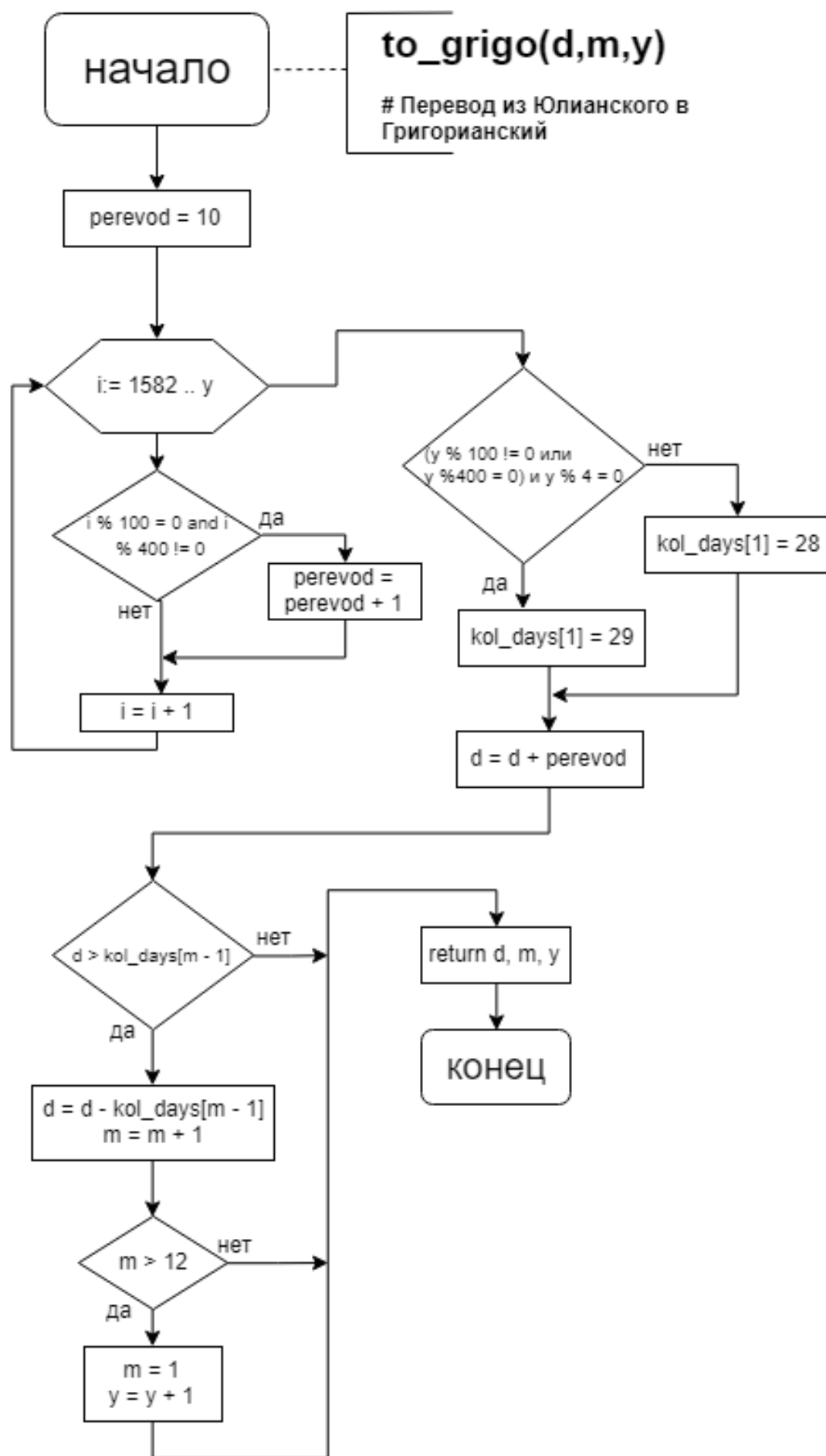
Ввод должен содержать серию строк, каждая строка содержит день недели и дату (например, Пятница 25 Декабря 1992). Даты могут быть в диапазоне от 1 Января 1600 до 31 Декабря 2099 года. Заметьте, что все дни недели и месяцы будут вводиться, как показано в примере, то есть первые буквы будут заглавные, а остальные маленькие. Ввод будет прерван строкой, содержащей символ '#’.

Вывод

Программа должна выводить серию строк, соответствующих введённым. Каждая строка будет содержать дату в другом стиле. Формат вывода такой же, как и для ввода. Даты в старом стиле должны содержать звёздочку ('*') после дня. Заметьте, что это не применимо для ввода.

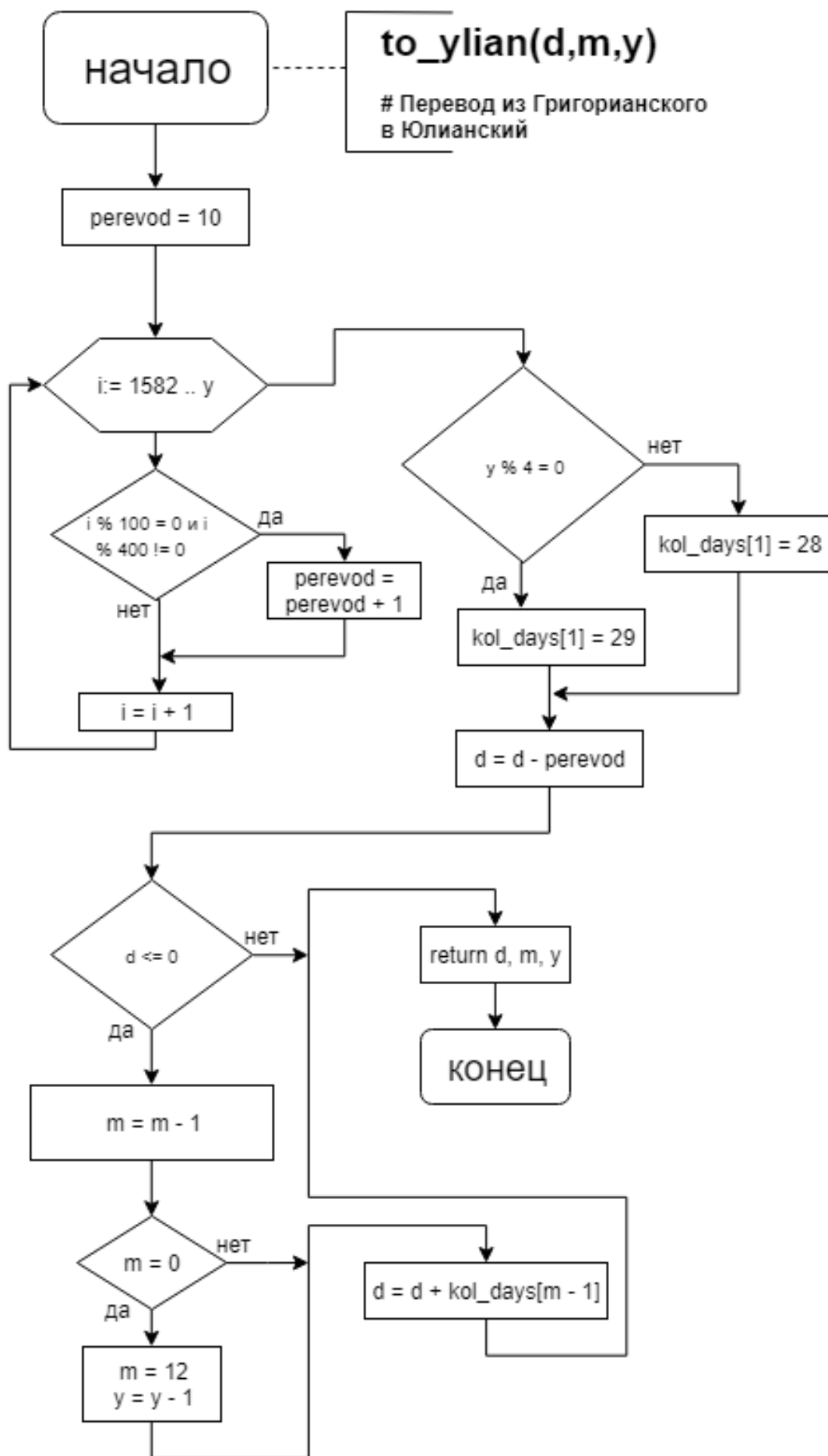
Блок-схема

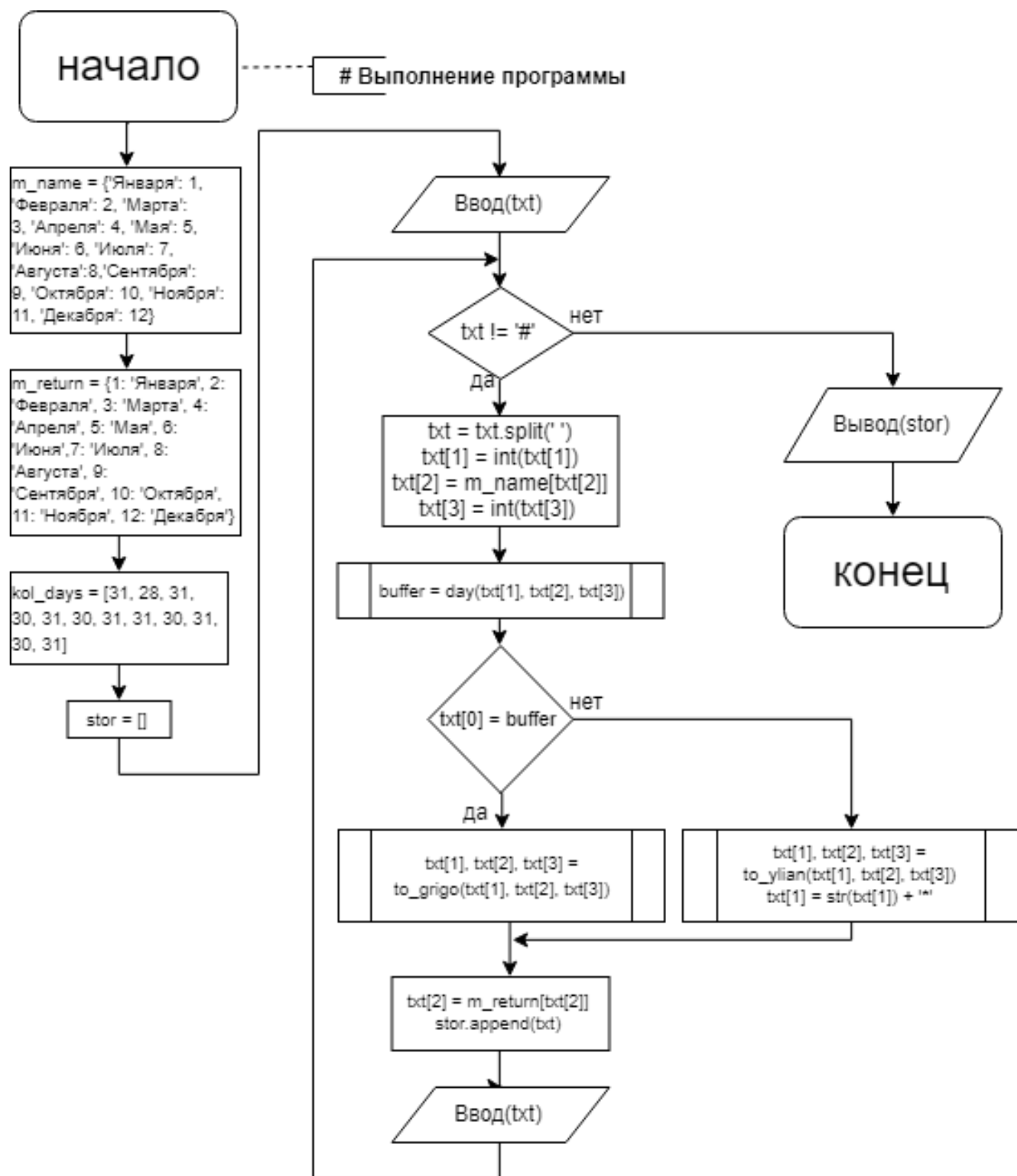




to_ylian(d,m,y)

Перевод из Григорианского
в Юлианский





Описание программы:

Алгоритм выполнения программы:

1. Пока введённая строка не является символом '#', с клавиатуры вводятся строки, содержащие День недели, день, месяц и год.
2. Для каждой строки данных вычисляется день недели по дню, месяцу и году для Григорианского календаря.
3. Если введённый день недели соответствует полученному дню недели, то был выполнен ввод даты в Григорианском календаре. Выполняем перевод введённой даты в Юлианский календарь. Записываем полученную дату в одномерный массив.
4. Если введённый день недели не соответствует полученному дню недели, то был выполнен ввод даты в Юлианском календаре. Выполняем перевод введённой даты в Григорианский календарь. Записываем полученную дату в одномерный массив.
5. при вводе символа '#' выводим полученный одномерный массив на экран.

Входные данные:

txt – строка. Тип str;

Выходные данные:

stor – одномерный массив строк. Тип list;

Вспомогательные данные:

m_name – словарь для перевода текстового формата месяца в цифровой. Тип dict;

m_return – словарь для перевода цифрового формата месяца в текстовый. Тип dict;

kol_days – одномерный массив кол-ва дней в каждом месяце. Тип list;

perevod – изначально кол-во дней смещения календарей. Тип int;

i – счётчик. Тип int;

d, m, y – день месяц год. Тип int;

kod_veka – код века введённой даты. Тип int;

kod_mes – код месяца введённой даты. Тип int;

god_last2 – последний 2 цифры года. Тип int;
kod_goda – код года. Тип int;
base – одномерный массив дней недели. Тип list;
ind – полученный номер дня недели. Тип int;

Текст программы с комментариями:

Определение дня недели по Григорианскому календарю

def day(d,m,y):

base = ['Суббота', 'Воскресенье', 'Понедельник', 'Вторник',
 'Среда', 'Четверг', 'Пятница']

Определение кода века введённого года

if (y >= 1600 and y < 1700) or (y >= 2000 and y < 2100):

kod_veka = 6

elif y >= 1700 and y < 1800:

kod_veka = 4

elif y >= 1800 and y < 1900:

kod_veka = 2

elif y >= 1900 and y < 2000:

kod_veka = 0

Определение кода месяца

if m == 4 or m == 7:

kod_mes = 0

elif m == 1 or m == 10:

kod_mes = 1

elif m == 5:

kod_mes = 2

elif m == 8:

kod_mes = 3

elif m == 2 or m == 3 or m == 11:

```

    kod_mes = 4
elif m == 6:
    kod_mes = 5
elif m == 12 or m == 9:
    kod_mes = 6
god_last2 = y - ((y // 100) * 100)    # Определение последних двух цифр введённого года
kod_goda = (kod_veka + god_last2 + (god_last2 // 4)) % 7    # Определение когда года
ind = (d + kod_goda + kod_mes) % 7    # Определение номера полученного месяца

# Смещение дня недели если год високосный и дата меньше 1 Марта
if y % 4 == 0 and ind != 0 and ((d >= 0 and d <= 31 and m == 'Января') \
                                or (d >= 0 and d <= 29 and m == 'Февраля')) \
    and (y % 100 != 0 or y % 400 == 0):

    ind -= 1
elif y % 4 == 0 and ind == 0 and ((d >= 0 and d <= 31 and m == 'Января') \
                                or (d >= 0 and d <= 29 and m == 'Февраля')) \
    and (y % 100 != 0 or y % 400 == 0):

    ind = 6

# Возвращение дня недел текстом
return base[ind]

# Перевод из Юлианского в Григорианский
def to_grigo(d,m,y):
    perevod = 10    # Начальная разница календарей

# Нахождение разницы календарей для текущего года
for i in range(1582, y + 1):
    if i % 100 == 0 and i % 400 != 0 and y != i:
        perevod += 1

```

```

elif i % 100 == 0 and i % 400 != 0 and y == i and m != 1 and m != 2:
    perevod += 1
# Изменение кол-ва дней Февраля в зависимости от високосности года
if (y % 100 != 0 or y % 400 == 0) and y % 4 == 0:
    kol_days[1] = 29
else:
    kol_days[1] = 28
d += perevod
if d > kol_days[m - 1]:
    d -= kol_days[m - 1]
    m += 1
    if m > 12:
        m = 1
        y += 1
return d, m, y

```

Перевод из Григорианского в Юлианский

```
def to_ylian(d, m, y):
```

```
    perevod = 10    # Начальная разница календарей
```

```
    # Нахождение разницы календарей для текущего года
```

```
    for i in range(1582, y + 1):
```

```
        if i % 100 == 0 and i % 400 != 0 and y != i:
```

```
            perevod += 1
```

```
        elif i % 100 == 0 and i % 400 != 0 and y == i and m != 1 and m != 2:
```

```
            perevod += 1
```

```
    # Изменение кол-ва дней Февраля в зависимости от високосности года
```

```
    if y % 4 == 0:
```

```
        kol_days[1] = 29
```

```
else:
```

```
    kol_days[1] = 28
```

```
d -= perevod
```

```
if d <= 0:
```

```
    m -= 1
```

```
    if m == 0:
```

```
        y -= 1
```

```
        m = 12
```

```
    d += kol_days[m - 1]
```

```
return d, m, y
```

```
# Выполнение программы
```

```
m_name = {'Января': 1, 'Февраля': 2, 'Марта': 3,
```

```
          'Апреля': 4, 'Мая': 5, 'Июня': 6,
```

```
          'Июля': 7, 'Августа': 8, 'Сентября': 9,
```

```
          'Октября': 10, 'Ноября': 11, 'Декабря': 12}
```

```
m_return = {1: 'Января', 2: 'Февраля', 3: 'Марта',
```

```
            4: 'Апреля', 5: 'Мая', 6: 'Июня',
```

```
            7: 'Июля', 8: 'Августа', 9: 'Сентября',
```

```
            10: 'Октября', 11: 'Ноября', 12: 'Декабря'}
```

```
kol_days = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
```

```
stor = []
```

```
print('Введите день недели, день, месяц, год: ')
```

```
print('Символ # чтобы закончить ввод')
```

```
txt = str(input())
```

```
while txt != '#':
```

```
    txt = txt.split(' ')
```

```
    txt[1] = int(txt[1])
```

```
    txt[2] = m_name[txt[2]]    # Перевод текстового месяца в цифровой
```

```
    txt[3] = int(txt[3])
```

```

if txt[3] % 400 != 0:
    # Перевод в другой календарь
    if txt[0] != day(txt[1], txt[2], txt[3]):
        txt[1], txt[2], txt[3] = to_grigo(txt[1], txt[2], txt[3])
    else:
        txt[1], txt[2], txt[3] = to_ylian(txt[1], txt[2], txt[3])
        txt[1] = str(txt[1])+'*'

    # Возврат месяца текстом
    txt[2] = m_return[txt[2]]
    stor.append(txt)
else:
    stor.append(0)
txt = str(input())

# Вывод полученных дат
print('Даты после перевода в другой календарь: ')
for i in range(len(stor)):
    if stor[i] == 0:
        print('Определить календарь для данного года невозможно')
    else:
        print(*stor[i])

```

Заключение:

Воспользовавшись способом быстрого определения дня недели по дню, месяцу и году, мы значительно упрощаем программу. Определение календаря по дню недели является невозможным для годов кратных 400, т.к. определить в каком календаре введена дата невозможно.

Список литературы:

1. Определение дня недели по Григорианскому календарю. [Электронный ресурс]. – URL: <https://lifehacker.ru/kakoj-den-nedeli/>
2. Общая информация про календари и способы перевода даты. [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/%D0%AE%D0%BB%D0%B8%D0%B0%D0%BD%D1%81%D0%BA%D0%B8%D0%B9_%D0%BA%D0%B0%D0%BB%D0%B5%D0%BD%D0%B4%D0%B0%D1%80%D1%8C
3. Автоматизированный перевод дат для проверки работоспособности программы. [Электронный ресурс]. – URL: <http://www.direct-time.ru/%D0%9A%D0%BE%D0%BD%D0%B2%D0%B5%D1%80%D1%82%D0%B5%D1%80%D0%B4%D0%B0%D1%82/>
4. Построение блок-схем по ГОСТу. [Электронный ресурс]. – URL: <https://pro-prof.com/archives/1462>