



Кирилл Станиславович Ткаченко
Инженер 1-й кат.,
ФГАОУ ВО «Севастопольский государственный университет»

Программная реализация алгоритма Гаусса для расчета даты Православной Пасхи на C++, Pascal, Python, школьном алгоритмическом языке

Предлагается программная реализация алгоритма Гаусса для расчета даты Православной Пасхи на C++, Pascal, Python, школьном алгоритмическом языке. Приводятся полные исходные тексты программ и результаты работы. Программы станут полезны всем изучающим основы программирования на достаточно простых и интересных примерах.

Изучение программирования становится полезным, когда элементы программирования можно применить на практике. Достаточно практичными являются календарные задачи, которые известны издревле. Одной из таких календарных задач является расчет даты Пасхи. Для расчета могут быть применены различные подходы. При этом существует алгоритм, созданный великим математиком – алгоритм Гаусса вычисления даты Пасхи [1]. Он существует в нескольких различных вариациях для старого и нового стилей календарей, для Православной Пасхи и для других. Его непосредственная программная реализация довольно простая, может быть доступна всем тем, кто только приступил к изучению программирования, алгоритмических языков, либо программирует достаточно слабо, испытывает при этом некоторые затруднения.

Поэтому в настоящей работе рассматриваются программы, реализующие алгоритм Гаусса для расчета даты Православной Пасхи по новому стилю на C++, Pascal, Python, школьном алгоритмическом языке.

Алгоритм по большей части, сводится к следующим расчетам (Год – номер года, mod – остаток от целочисленного деления):

1. $a \leftarrow \text{mod}(\text{Год}, 19)$;
2. $b \leftarrow \text{mod}(\text{Год}, 4)$;

3. $c \leftarrow \text{mod}(\text{Год}, 7);$
4. $d \leftarrow \text{mod}(19 * a + 15, 30);$
5. $e \leftarrow \text{mod}(2 * b + 4 * c + 6 * d + 6, 7);$
6. $f \leftarrow d + e;$
7. Если $f \leq 26$, то искомая дата равна $(f + 4)$ апреля, иначе равна $(f - 26)$ мая.

Схема алгоритма приводится на рисунке 1.

Для предыдущего года (Год=2021) расчеты по приведенному алгоритму осуществляются следующим образом:

1. $a \leftarrow \text{mod}(\text{Год}, 19) = \text{mod}(2021, 19) = 7;$
2. $b \leftarrow \text{mod}(\text{Год}, 4) = \text{mod}(2021, 4) = 1;$
3. $c \leftarrow \text{mod}(\text{Год}, 7) = \text{mod}(2021, 7) = 5;$
4. $d \leftarrow \text{mod}(19 * a + 15, 30) = \text{mod}(19 * 7 + 15, 30) = \text{mod}(148, 30) = 28;$
5. $e \leftarrow \text{mod}(2 * b + 4 * c + 6 * d + 6, 7) = \text{mod}(2 * 1 + 4 * 5 + 6 * 28 + 6, 7) = \text{mod}(196, 7) = 0;$
6. $f \leftarrow d + e = 28 + 0 = 28;$
7. Если $f \leq 26$, то искомая дата равна $(f + 4)$ апреля, иначе равна $(f - 26)$ мая. Поскольку $28 > 26$, то искомая дата равна $(28 - 26)$ мая – 2 мая 2021 г.

Для текущего года (Год=2022) расчеты по приведенному алгоритму осуществляются следующим образом:

1. $a \leftarrow \text{mod}(\text{Год}, 19) = \text{mod}(2022, 19) = 8;$
2. $b \leftarrow \text{mod}(\text{Год}, 4) = \text{mod}(2022, 4) = 2;$
3. $c \leftarrow \text{mod}(\text{Год}, 7) = \text{mod}(2022, 7) = 6;$
4. $d \leftarrow \text{mod}(19 * a + 15, 30) = \text{mod}(19 * 8 + 15, 30) = \text{mod}(167, 30) = 17;$
5. $e \leftarrow \text{mod}(2 * b + 4 * c + 6 * d + 6, 7) = \text{mod}(2 * 2 + 4 * 6 + 6 * 17 + 6, 7) = \text{mod}(136, 7) = 3;$
6. $f \leftarrow d + e = 17 + 3 = 20;$
7. Если $f \leq 26$, то искомая дата равна $(f + 4)$ апреля, иначе равна $(f - 26)$ мая. Поскольку $20 \leq 26$, то искомая дата равна $(20 + 4)$ апреля – 24 апреля 2022 г.

Видно, что описанный и изображенный на рисунке 1 алгоритм достаточно простой и сводится к использованию операторов присваивания и ветвления. Поэтому он может быть непосредственно реализован на значительном количестве языков программирования. Далее рассматриваются четыре случая – на C++, Pascal, Python, школьном алгоритмическом языке.

Вначале необходимо произвести некоторые дополнительные действия.

На языке C++ – указать на необходимость включения файлов определений:

```
#include <iostream>
#include <sstream>
#include <string>
```

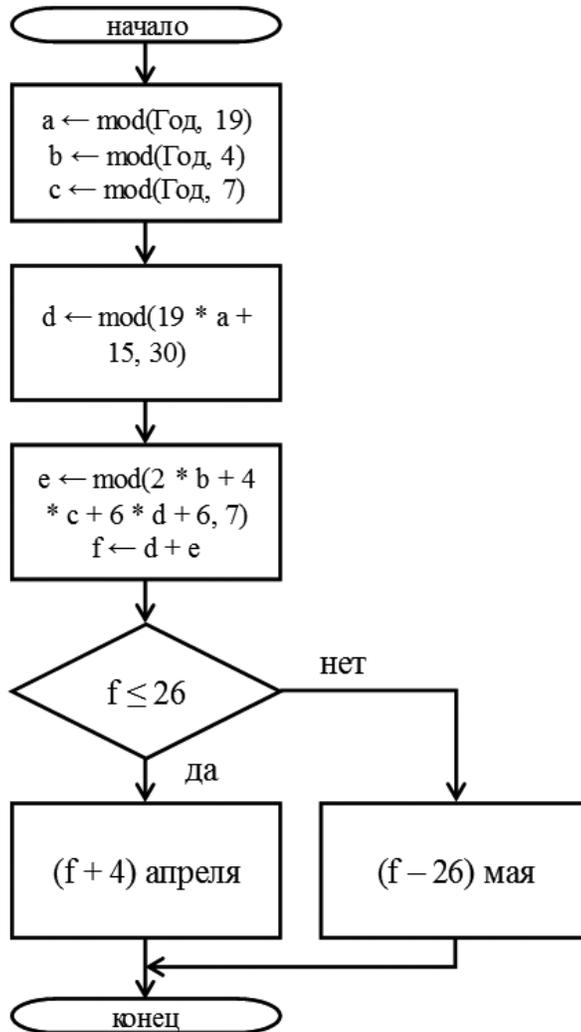


Рис. 1 – Схема алгоритма

На языке Pascal – задать идентификатор программы и определить функцию для перевода числа в строковое представление (в некоторых реализациях и диалектах возможно в дальнейшем непосредственно использовать, например, функцию IntToStr):

```

program Potentsial;
function toStr(x : integer) : string;
begin
  str(x, toStr)
end;

```

Для удобства головной алгоритм на школьном алгоритмическом языке будет рассмотрен вместе с другими головными подпрограммами.

Теперь определяются функции для календарного расчета. Их единственным аргументом является номер года. На языке C++ заголовок определения функции выглядит так:

```
std::string raschet(int god) {
```

На языке Pascal заголовок определения функции и переменных записывается следующим образом:

```
function raschet(god : integer) : string;
var
  a, b, c, d, e, f : integer;
  rez : string;
begin
```

На языке Python:

```
def raschet(god):
```

На школьном алгоритмическом языке:

```
алг лит Расчет(цел Год)
нач
  цел a, b, c, d, e, f;
```

Далее реализуются шаги (1)–(6) алгоритма. На языке C++:

```
int a = god % 19;
int b = god % 4;
int c = god % 7;
int d = (19 * a + 15) % 30;
int e = (2 * b + 4 * c + 6 * d + 6) % 7;
int f = d + e;
```

На языке Pascal:

```
a := god mod 19;
b := god mod 4;
c := god mod 7;
d := (19 * a + 15) mod 30;
e := (2 * b + 4 * c + 6 * d + 6) mod 7;
f := d + e;
```

На языке Python:

```
a = god % 19
b = god % 4
c = god % 7
d = (19 * a + 15) % 30
e = (2 * b + 4 * c + 6 * d + 6) % 7
f = d + e
```

На школьном алгоритмическом языке:

```
a := mod(Год, 19);
b := mod(Год, 4);
c := mod(Год, 7);
d := mod(19 * a + 15, 30);
e := mod(2 * b + 4 * c + 6 * d + 6, 7);
f := d + e;
```

В целом, могут быть произведены некоторые незначительные оптимизации и упрощения. В частности, например, можно заметить, что в шагах (4) и (5) результаты расчетов (1)–(3) фигурируют единственный раз. Поэтому путем подстановки шаги (1)–(6) можно записать иначе, но получившиеся выражения, возможно, будут выглядеть «громоздко».

Теперь следует реализовать шаг (7), а затем вернуть сформированную в строковом виде дату. Этот шаг и формирование даты на языке C++ при использовании объекта класса для удобной в рамках решаемой задачи обработки строк `std::ostringstream` записывается следующим образом:

```
std::ostringstream rez;
if (f <= 26) {
    rez << (f + 4) << ".04";
} else {
    rez << (f - 26) << ".05";
}
rez << "." << god;
return rez.str();
}
```

На языке Pascal:

```
if f <= 26 then
    rez := toStr(f + 4) + '.04'
else
    rez := toStr(f - 26) + '.05';
raschet := rez + '.' + toStr(god)
end;
```

На языке Python:

```
if f <= 26:
    rez = str(f + 4) + ".04"
else:
    rez = str(f 26) + ".05"
return rez + «.» + str(god)
```

На школьном алгоритмическом языке:

```
если f <= 26
    то
        знач := цел_в_лит(f + 4) + «.04»;
    иначе
        знач := цел_в_лит(f 26) + «.05»;
    все
    знач := знач + «.» + цел_в_лит(Год);
кон
```

Основная часть работы выполнена. Демонстрация разработанных функций производится на основе их вызовов из головной функции. На языке C++:

```
int main(int, char**) {
    std::cout << raschet(2021) << std::endl;
    std::cout << raschet(2022) << std::endl;
    return 0;
}
```

На языке Pascal:

```
begin
    writeln(raschet(2021));
    writeln(raschet(2022))
end.
```

На языке Python (для упрощения не производится определение головной функции):

```
print(raschet(2021))
print(raschet(2022))
```

На школьном алгоритмическом языке:

```
алг Потенциал
нач
    вывод Расчет(2021), нс;
    вывод Расчет(2022), нс;
кон
```

Полный исходный текст программы на языке С++ приводится в приложении А, Pascal – Б, Python – В, школьном алгоритмическом языке – Г, результаты работы программ – Д, снимок экрана программы на школьном алгоритмическом языке в среде КуМир – на рисунке 2.

```

1  алг Потенциал
2  нач
3  . вывод Расчет(2021), нс;
4  . вывод Расчет(2022), нс;
5  кон
6
7  алг лит Расчет(цел Год)
8  нач
9  . цел a, b, c, d, e, f;
10 . a := mod(Год, 19);
11 . b := mod(Год, 4);
12 . c := mod(Год, 7);
13 . d := mod(19 * a + 15, 30);
14 . e := mod(2 * b + 4 * c + 6 * d + 6, 7);
15 . f := d + e;
16 . если f <= 26
17 . . то
18 . . . знач := цел_в_лит(f + 4) + ".04";
19 . . иначе
20 . . . знач := цел_в_лит(f - 26) + ".05";
21 . все
22 . знач := знач + "." + цел_в_лит(Год);
23 кон

```

2.05.2021
24.04.2022

Анализ Выполнено шагов: 31

Рис. 2 – Программа на школьном алгоритмическом языке в среде КуМир

Полученные программы найдут применение при начальном обучении программированию и изучении нескольких различных высокоуровневых языков. Интересный и достаточно простой расчет является хорошей демонстрацией перед рассмотрением большого количества практических календарных задач.

Литература

1. Алгоритм Гаусса вычисления даты Пасхи // Википедия. URL: https://ru.wikipedia.org/wiki/Алгоритм_Гаусса_вычисления_даты_Пасхи (дата обращения: 22.03.2021).

Приложение А

Полный исходный текст программы на языке C++

```
#include <iostream>
#include <sstream>
#include <string>

std::string raschet(int god) {
    int a = god % 19;
    int b = god % 4;
    int c = god % 7;
    int d = (19 * a + 15) % 30;
    int e = (2 * b + 4 * c + 6 * d + 6) % 7;
    int f = d + e;
    std::ostringstream rez;
    if (f <= 26) {
        rez << (f + 4) << ".04";
    } else {
        rez << (f - 26) << ".05";
    }
    rez << "." << god;
    return rez.str();
}

int main(int, char**) {
    std::cout << raschet(2021) << std::endl;
    std::cout << raschet(2022) << std::endl;
    return 0;
}
```

Приложение Б

Полный исходный текст программы на языке Pascal

```
program Potentsial;

function toStr(x : integer) : string;
begin
    str(x, toStr)
end;

function raschet(god : integer) : string;
var
```

```
a, b, c, d, e, f : integer;
rez : string;
begin
  a := god mod 19;
  b := god mod 4;
  c := god mod 7;
  d := (19 * a + 15) mod 30;
  e := (2 * b + 4 * c + 6 * d + 6) mod 7;
  f := d + e;
  if f <= 26 then
    rez := toStr(f + 4) + '.04'
  else
    rez := toStr(f 26) + '.05';
  raschet := rez + '.' + toStr(god)
end;

begin
  writeln(raschet(2021));
  writeln(raschet(2022))
end.
```

Приложение В

Полный исходный текст программы на языке Python

```
def raschet(god):
    a = god % 19
    b = god % 4
    c = god % 7
    d = (19 * a + 15) % 30
    e = (2 * b + 4 * c + 6 * d + 6) % 7
    f = d + e
    if f <= 26:
        rez = str(f + 4) + ".04"
    else:
        rez = str(f 26) + ".05"
    return rez + "." + str(god)

print(raschet(2021))
print(raschet(2022))
```

Приложение Г

Полный исходный текст программы на школьном алгоритмическом языке

```
алг Потенциал
нач
    вывод Расчет(2021), нс;
    вывод Расчет(2022), нс;
кон

алг лит Расчет(цел Год)
нач
    цел a, b, c, d, e, f;
    a := mod(Год, 19);
    b := mod(Год, 4);
    c := mod(Год, 7);
    d := mod(19 * a + 15, 30);
    e := mod(2 * b + 4 * c + 6 * d + 6, 7);
    f := d + e;
    если f <= 26
        то
            знач := цел_в_лит(f + 4) + «.04»;
        иначе
            знач := цел_в_лит(f 26) + «.05»;
    все
    знач := знач + «.» + цел_в_лит(Год);
кон
```

Приложение Д

Результаты работы программ

2.05.2021
24.04.2022