**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 2**

Тема: Перегрузка операторов в С++

Студент: Шахниязов Ботир

Группа: 80-201

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2019

1. Постановка задачи

Вариант 5. Создать класс Modulo для работы с целыми числами по модулю N. В классе должно быть два поля: число и N. Реализовать все арифметические операции. Реализовать операции сравнения. Операции сложения, вычитания, умножения, деления, сравнения (на равенство, больше и меньше) должны быть выполнены в виде перегрузки операторов. Необходимо реализовать пользовательский литерал для работы с константами типа Modulo.

1. Описание программы

* Написал класс Modula в соответствии со своим вариантом.
* Программа работает корректно в случаях, когда пользователь вводит отрицательный модуль или пытается разделить на ноль.
* Операции +,-,\*,/ возможны, когда модули соответствующих чисел равны, в противном случае в консоль выводится сообщение о том, что модули не равны.
* Написал меню.

1. Набор testcases

Набор первый. Проверка краевых случаев (деление на ноль, отрицательный модуль от числа и т.д.)

2

100

-5

2

100

0

2

100

8

3

0

10

7

0

Набор второй.

2

123

12

3

321

32

4

5

6

7

8

9

10

0

1. Результаты выполнения тестов.

Первый тест.

Welcome to the modula calculator!

A,B - variable

C - result

A = A.num % A.mod

Menu

0. quit

1. print menu

2. input A

3. input B

4. C = A + B

5. C = A - B

6. C = A \* B

7. C = A / B

8. C = A == B

9. C = A > B

10. C = A < B

Значения:

A = 0 % 1 = 0

B = 0 % 1 = 0

C = 0 % 1 = 0

-->2

A = 100

mod = -5

остаток от деления должен быть больше нуля.

Значения:

A = 0 % 1 = 0

B = 0 % 1 = 0

C = 0 % 1 = 0

-->2

A = 100

mod = 0

остаток от деления должен быть больше нуля.

Значения:

A = 0 % 1 = 0

B = 0 % 1 = 0

C = 0 % 1 = 0

-->2

A = 100

mod = 8

Значения:

A = 100 % 8 = 4

B = 0 % 1 = 0

C = 0 % 1 = 0

-->3

B = 0

mod = 10

Значения:

A = 100 % 8 = 4

B = 0 % 10 = 0

C = 0 % 1 = 0

-->7

Модули не равны, операция / не корректна

Или делитель равен нулю

Значения:

A = 100 % 8 = 4

B = 0 % 10 = 0

C = 0 % 1 = 0

-->0

Второй тест.

Welcome to the modula calculator!

A,B - variable

C - result

A = A.num % A.mod

Menu

0. quit

1. print menu

2. input A

3. input B

4. C = A + B

5. C = A - B

6. C = A \* B

7. C = A / B

8. C = A == B

9. C = A > B

10. C = A < B

Значения:

A = 0 % 1 = 0

B = 0 % 1 = 0

C = 0 % 1 = 0

-->2

A = 123

mod = 12

Значения:

A = 123 % 12 = 3

B = 0 % 1 = 0

C = 0 % 1 = 0

-->3

B = 321

mod = 32

Значения:

A = 123 % 12 = 3

B = 321 % 32 = 1

C = 0 % 1 = 0

-->4

Модули не равны, операция + не корректна

Значения:

A = 123 % 12 = 3

B = 321 % 32 = 1

C = 0 % 1 = 0

-->5

Модули не равны, операция - не корректна

Значения:

A = 123 % 12 = 3

B = 321 % 32 = 1

C = 0 % 1 = 0

-->6

Модули не равны, операция \* не корректна

Значения:

A = 123 % 12 = 3

B = 321 % 32 = 1

C = 0 % 1 = 0

-->7

Модули не равны, операция / не корректна

Или делитель равен нулю

Значения:

A = 123 % 12 = 3

B = 321 % 32 = 1

C = 0 % 1 = 0

-->8

Значения:

A = 123 % 12 = 3

B = 321 % 32 = 1

C = 0 % 2 = 0

-->9

Значения:

A = 123 % 12 = 3

B = 321 % 32 = 1

C = 1 % 2 = 1

-->10

Значения:

A = 123 % 12 = 3

B = 321 % 32 = 1

C = 0 % 2 = 0

-->0

1. Листинг программы

/\*

\* Шахниязов Ботир. Группа: м8о-201Б.

\*

\* Создать класс Modulo для работы с целыми числами по модулю N.

\* В классе должно быть два поля: число и N.

\* Реализовать все арифметические операции. Реализовать операции сравнения.

\* Операции сложения, вычитания, умножения,

\* деления, сравнения (на равенство, больше и меньше) должны быть выполнены в виде перегрузки операторов.

\* Необходимо реализовать пользовательский литерал для работы с константами типа Modulo.

\*/

#include <iostream>

class Modulo

{

private:

int number;

int N;

public:

Modulo(int \_number, int \_N)

{

number = \_number;

N = \_N;

}

Modulo(int \_number)

{

number = \_number;

N = 1;

}

Modulo()

{

number = 0;

N = 1;

}

Modulo operator +(Modulo b)

{

Modulo c(0,N);

if (N == b.N)

c.number = (number + b.number) % N;

else{

c.number = 0;

c.N = 1;

std::cout << "Модули не равны, операция" << " + " << "не корректна\n";

}

return c;

}

Modulo operator -(Modulo b)

{

Modulo c(0,N);

if (N == b.N)

c.number = (number - b.number) % N;

else{

c.number = 0;

c.N = 1;

std::cout << "Модули не равны, операция" << " - " << "не корректна\n";

}

return c;

}

Modulo operator \*(Modulo b)

{

Modulo c(0,N);

if (N == b.N)

c.number = (number \* b.number) % N;

else{

c.number = 0;

c.N = 1;

std::cout << "Модули не равны, операция" << " \* " << "не корректна\n";

}

return c;

}

Modulo operator /(Modulo b)

{

Modulo c(0,N);

if (N == b.N && b.number != 0)

c.number = (number / b.number) % N;

else{

c.number = 0;

c.N = 1;

std::cout << "Модули не равны, операция" << " / " << "не корректна\n";

std::cout << "Или делитель равен нулю\n";

}

return c;

}

int operator ==(Modulo b)

{

Modulo c;

if(number % N == b.number % b.N)

return 1;

else

return 0;

}

int operator >(Modulo b)

{

Modulo c;

if(number % N > b.number % b.N)

return 1;

else

return 0;

}

int operator <(Modulo b)

{

Modulo c;

if(number % N < b.number % b.N)

return 1;

else

return 0;

}

const void print\_modula() const

{

std::cout << number << " % " << N << " = " << number % N << std::endl;

}

void set(int \_number, int \_N)

{

number = \_number;

N = \_N;

}

};

Modulo operator "" \_num(unsigned long long b)

{

return b;

}

int main()

{

std::cout << "\n\nWelcome to the modula calculator!\n";

std::cout << "A,B - variable\n";

std::cout << "C - result\n";

std::cout << "A = A.num % A.mod\n\n";

Modulo a, b, c;

int k=1, num, mod;

bool loop = true;

while(loop)

{

switch (k)

{

case 1:

std::cout << "Menu\n";

std::cout << "0. quit\n";

std::cout << "1. print menu\n";

std::cout << "2. input A\n";

std::cout << "3. input B\n";

std::cout << "4. C = A + B\n";

std::cout << "5. C = A - B\n";

std::cout << "6. C = A \* B\n";

std::cout << "7. C = A / B\n";

std::cout << "8. C = A == B\n";

std::cout << "9. C = A > B\n";

std::cout << "10. C = A < B\n";

break;

case 2:

std::cout << "A = "; std::cin >> num;

std::cout << "mod = "; std::cin >> mod;

if(mod > 0)

a.set(num,mod);

else

std::cout << "остаток от деления должен быть больше нуля.\n";

break;

case 3:

std::cout << "B = "; std::cin >> num;

std::cout << "mod = "; std::cin >> mod;

if(mod > 0)

b.set(num,mod);

else

std::cout << "остаток от деления должен быть больше нуля.\n";

break;

case 4: c = a + b; break;

case 5: c = a - b; break;

case 6: c = a \* b; break;

case 7: c = a / b; break;

case 8: c.set(a == b,2); break;

case 9: c.set(a > b,2); break;

case 10: c.set(a < b,2); break;

default: break;

}

std::cout << "\nЗначения:" << std::endl;

std::cout << "A = "; a.print\_modula();

std::cout << "B = "; b.print\_modula();

std::cout << "C = "; c.print\_modula();

std::cout << " \n-->";

std::cin >> k;

if(!k) loop = false;

}

return 0;

}

1. Вывод:

Научился создавать пользовательские литералы, а также научился перегружать стандартные арифметические операторы.

## Список литературы

1. Перегрузка операторов в C++ [Электронный ресурс]  
   URL: <https://habr.com/ru/post/132014/>
2. Пользовательские литералы в C++11 [Электронный ресурс]  
   URL: <https://habr.com/ru/post/140357/>