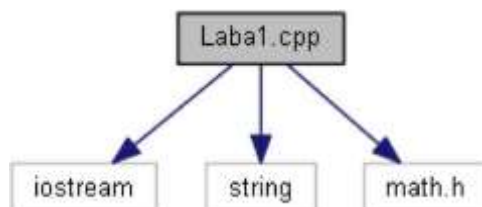


Laba1 .cpp File Reference

Файл реализующий первую лабораторную работу. [More...](#)

```
#include <iostream>
#include <string>
#include <math.h>
```

Include dependency graph for Laba1.cpp:



Functions

template<class T >

string **Standart** (T number)

Функция переводящая число в двоичное представление [More...](#)

template<class T >

string **Union** (T number)

Функция переводящая число в двоичное представление [More...](#)

string **ConvertToBinary** (int argc, char *argv[])

Функция парсящая входные параметры [More...](#)

int **main** (int argc, char *argv[])

Основная функция программы [More...](#)

Description

Detailed

Написать программу, принимающую на вход вещественное число и переводящую его в двоичное
Файл реализующий первую лабораторную работу.

Задача:
представление.

Function Documentation

◆ ConvertToBinary()

```
string ConvertToBinary ( int    argc,
                        char* argv[]
                        )
```

Функция парсящая входные параметры

Parameters

argc – количество передающихся параметров
argv[] – входные строки

Returns

string - результаты метода Standrt/Union

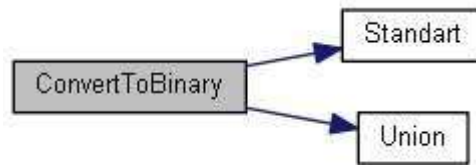
Код функции выглядит следующим образом:

```
string ConvertToBinary(int argc, char* argv[])
{
    string number, typeOfNumber, typeOfOperation;
    for (int i = 0; i < argc; i++)
    {
        string str = argv[i];
        if (str.find("--number") != string::npos)
            number = str.substr(str.find("=") + 1);
        else if (str.find("--type") != string::npos)
            typeOfNumber = str.substr(str.find("=") + 1);
        else if (str.find("--method") != string::npos)
            typeOfOperation = str.substr(str.find("=") + 1);
    }
    try
    {
        if (typeOfOperation == "classic")
        {
            if (typeOfNumber == "float")
                return Standart(stof(number));
            else if (typeOfNumber == "double")
                return Standart(stod(number));
            else
                return "Error type!";
        }
        else if (typeOfOperation == "union")
        {
            if (typeOfNumber == "float")
                return Union(stof(number));
            else if (typeOfNumber == "double")
                return Union(stod(number));
            else
                return "Error type!";
        }
        else
            return "Error method!";
    }
    catch (const std::exception&)
    {
        return "Error number!";
    }
}
```

Authors

Tereshin D.D

Here is the call graph for this function:



Here is the caller graph for this function:



◆ main()

```
int main ( int    argc,
           char * argv[]
        )
```

Основная функция программы

Parameters

argc – количество передающихся параметров
argv[] – входные строки

Returns

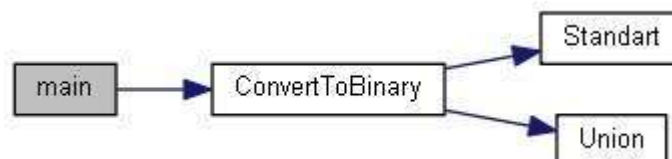
int

```
int main(int argc, char* argv[])
{
    cout << "result: " << ConvertToBinary(argc, argv) << endl;
    return 0;
}
```

Authors

Tereshin D.D

Here is the call graph for this function:



◆ Standart()

```
template<class T >
string Standart ( T number )
```

Функция переводящая число в двоичное представление

Parameters

number – число, которое нужно перевести

Returns

string - результат перевода

Код функции выглядит следующим образом:

```
template <class T>
string Standart(T number)
{
    bool positive = true;
    string result, str;
    int Helper, exp, termOfExp = 0, termOfMantissa = 0, countOfExp = 0;
    T IntPart = 0, FractPart = 0;

    if (number < 0)
    {
        positive = false;
        number = abs(number);
    }

    if (typeid(number).name() == typeid(double).name())
    {
        FractPart = modf(number, &IntPart);
        termOfExp = 1023;
        termOfMantissa = 52;
        countOfExp = 11;
    }
    else if (typeid(number).name() == typeid(float).name())
    {
        FractPart = modf((float)number, &IntPart);
        termOfExp = 127;
        termOfMantissa = 23;
        countOfExp = 8;
    }

    if (number == 0)
    {
        for (int i = 0; i < ((countOfExp + termOfMantissa + 1) / 8); i++)
        {
            for (int j = 0; j < 8; j++)
            {
                result += "0";
            }
            result += " ";
        }
        return result;
    }

    if (IntPart == 0)
        str += "0";

    while (IntPart >= 1)
    {
        Helper = (int)IntPart % 2;
        IntPart /= 2;
        str += to_string(Helper);
    }

    reverse(str.begin(), str.end());

    if (FractPart != 0)
        str += ".";

    while (FractPart != 0)
    {
```

```

    Helper = FractPart * 2;
    FractPart = FractPart * 2 - Helper;
    str += to_string(Helper);
}

int posOfOne = str.find("1");
int posOfDot = str.find(".");

if (posOfDot == string::npos)
{
    exp = str.length() - posOfOne - 1;
    str.erase(0, posOfOne + 1);
}
else
{
    if (posOfDot < posOfOne)
        exp = posOfDot - posOfOne;
    else
        exp = posOfDot - posOfOne - 1;

    str.erase(posOfDot, 1);
    posOfOne = str.find("1");
    str.erase(0, posOfOne + 1);
}

exp = exp + termOfExp;

while (exp >= 1)
{
    Helper = exp % 2;
    exp /= 2;
    result += to_string(Helper);
}

while (result.length() != countOfExp)
    result += "0";

reverse(result.begin(), result.end());

if (positive)
    result.insert(0, "0");
else
    result.insert(0, "1");

for (int i = 0; i < str.length(); i++)
    result += str[i];
for (int i = str.length(); i < termOfMantissa; i++)
    result += "0";

for (int i = result.length() - 8; i > 0; i -= 8)
    result.insert(i, " ");

return result;
}

```

Authors

Tereshin D.D

Here is the caller graph for this function:



◆ Union()

```
template<class T >
string Union ( T number )
```

Функция переводящая число в двоичное представление

Parameters

number – число, которое нужно перевести

Returns

string - результат перевода

Код функции выглядит следующим образом:

```
template <class T>
string Union(T number)
{
    union
    {
        T value;
        char bytes[sizeof(T)];
    }variable;

    variable.value = number;
    string str;

    if ((string(typeid(number).name()) == "float") || (string(typeid(number).name()) == "double"))
    {
        for (int i = sizeof(T) - 1; i >= 0; i--)
        {
            for (int j = 7; j >= 0; j--)
                str += to_string(((variable.bytes[i] >> j) & 1));
            str += " ";
        }
        return str;
    }
    return "Error";
}
```

Authors

Tereshin D.D

Here is the caller graph for this function:



Результаты работы программы.

```
C:\Users\qweds\source\repos\Laba1\Debug>Laba1.exe --number=-0.234 --type=float --method=classic
result: 10111110 01101111 10011101 10110010

C:\Users\qweds\source\repos\Laba1\Debug>Laba1.exe --number=-0.234 --type=double --method=union
result: 10111111 11001101 11110011 10110110 01000101 10100001 11001010 11000001

C:\Users\qweds\source\repos\Laba1\Debug>_
```