**

Колледж космического машиностроения и технологии

**КУРСОВОЙ ПРОЕКТ**

**По дисциплине «Прикладное программирование»**

**Тема: «Разработка программы-шифровальщика»**

Выполнил студент

Лоборев Максим Владимирович

Группа П2-17

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_ (Дата сдачи работы)

Королев, 2020

**Оглавление**

[Введение](#_1fob9te) **3**

[Глава 1. Теоретическая часть](#_3znysh7) **4**

[1.1 Описание предметной области](#_2et92p0) 4

[1.2 Описание существующих разработок](#_2s8eyo1) 5

[Глава 2. Проектная часть](#_3rdcrjn) **6**

[2.1 Диаграмма прецедентов](#_26in1rg) 6

[2.2 Выбор инструментов](#_35nkun2) 7

[2.3 Проектирование сценария](#_44sinio) 8

[2.4 Диаграмма классов](#_z337ya) 9

[2.5 Описание главного модуля](#_3j2qqm3) 10

[2.6 Описания спецификаций к модулям](#_1y810tw) 12

[2.7 Описание модулей](#_4i7ojhp) 21

[2.8 Описание тестовых наборов модулей](#_2xcytpi) 22

[2.9 Описание применения средств отладки](#_1ci93xb) 25

[Глава 3. Эксплуатационная часть](#_qsh70q) **28**

[Руководство оператора](#_3as4poj) 28

[3.1 Назначение программы](#_1pxezwc) 28

[3.2 Условия выполнения программы](#_49x2ik5) 29

[3.3 Выполнение программы](#_147n2zr) 30

[Заключение](#_3o7alnk) **31**

[Список литературы и интернет-источников](#_23ckvvd) **31**

[Приложения](#_ihv636) **31**

# Введение

Целью данного курсового проекта является написание программы-шифровальщика, которая пользователю общаться тайно. Это нужно для облегчения процесса шифрования текста и для передачи данных, не предназначаются для чужих глаз. Данный проект будет отличаться быстродействием и низкими требованиями к системным требованиям.

В первой части будут рассмотрены предметная область данной темы и несколько готовых решений.

Во второй части будут рассмотрены разработанные модули и инструменты, а также листинги ключевых частей программы.

В третьей части будет рассмотрено руководство оператора.

# Глава 1. Теоретическая часть

## 1.1 Описание предметной области

Шифрование данных - распространённый способ тайной передачи информации. Оно необходимо для:

* Разведчиков, так как данные не могут быть рассекречены
* Хранения информации в ненадёжных источниках и для её передачи по незащищённым каналам связи

Степень изученности методов шифрования достаточно высокая. Каждый год создается множество программ, литературы, посвящённым криптографическим системам. Криптографическая система - семейство преобразований шифра и совокупность ключей. Существуют Симметричные и Асимметричные криптосистемы.

Симметричные криптосистемы (с секретным ключом - secret key systems) - данные криптосистемы построены на основе сохранения в тайне ключа шифрования. Процессы зашифрования и расшифрования используют один и тот же ключ. Секретность ключа является постулатом.

Асимметричные криптосистемы (системы открытого шифрования - с открытым ключом public key systems) - смысл данных криптосистем состоит в том, что для зашифрования и расшифрования используются разные преобразования. Одно из них - зашифрование - является абсолютно открытым для всех. Другое же - расшифрование - остается секретным.

На данный момент чаще используются достаточно стойкие системы, системы с достаточно сложным алгоритмом шифрования. Из-за необходимости различных объектов зашифровывать секретные данные и криптографические системы не стоят на месте и постоянно совершенствуются.

## 1.2 Описание существующих разработок

AS-Text Crypt – бесплатная и простая утилита для шифрования текста. Пригодится для безопасного обмена текстовой информацией в интернете.

Шифровальщик позволяет обезопасить текст используя два алгоритма шифрования на выбор – AES 128 и 256 bit или Blowfish 128, 256 и 384 bit. Пользоваться достаточно просто – в поле «Text» нужно ввести текст, можно скопировать его из буфера обмена, а после указать пароль и выбрать алгоритм шифрования. После этого нажать кнопку Encrypt.

Далее нужно сообщить вашему собеседнику пароль расшифровки, и можно отправлять сообщения по электронной почте или передавать в чатах, не опасаясь утечки личной переписки.

[AS-Text Crypt](https://blogosoft.com/23963)

Входными данными моей программы являются:

Незашифрованная строка, состоящая из латинских символов и цифр

# Глава 2. Проектная часть

## 2.1 Диаграмма прецедентов

Для определения вариантов использования к проекту была построена диаграмма прецедентов

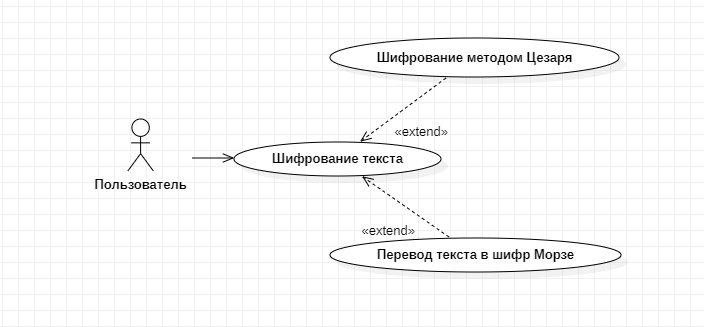


Рисунок 1. Диаграмма прецедентов для проекта

## 2.2 Выбор инструментов

При написании курсовой работы были использованы: интегрированная среда разработки "Microsoft Visual Studio" и язык C++.

|  |  |  |
| --- | --- | --- |
|  | CodeBlocks | Microsoft Visual Studio |
| Простота | 6 | 7 |
| Быстрота | 8 | 9 |
| Функционал | 7 | 10 |
| Удобство использования | 6 | 8 |
| Документация на русском языке | 3 | 9 |
| Итого | 30 | 43 |

Microsoft Visual Studio — инструментальная среда разработки, включающая в себя интегрированную среду разработки, редактор исходного кода, встроенный отладчик. Многие другие инструменты возможно получить благодаря подключению плагинов – сторонних расширений.

## 2.3 Проектирование сценария

В данном разделе приведён пример сценария использования программы оператором и клиентом.

После запуска программы клиент может выбрать валюту и её необходимое количество. Затем он должен ввести свои паспортные данные (ФИО, номер паспорта). После этих действий программа выдаст квитанцию об оплате.

Также, для оператора предусмотрен режим редактирования данных, в который можно получить доступ, используя пароль. В нём оператору доступны различные функции, подробнее о которых можно узнать в Главе 3 "Руководство оператора".

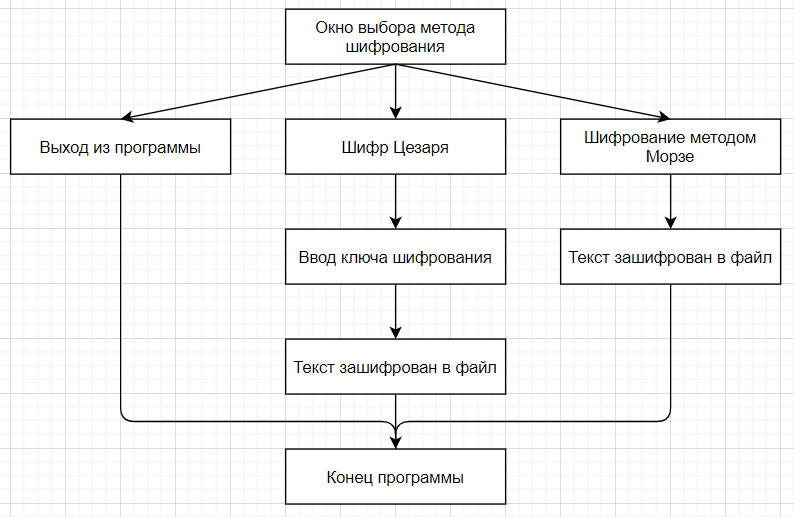


Рисунок 2. Сценарий использования

## 2.4 Диаграмма классов

В диаграмме классов представлен класс Cipher, содержащий два метода шифрования текста.

В классе Cipher используется спецификатор доступа public, что говорит нам о том, что доступ открыт всем, кто видит определение данного класса.

В данном классе нет полей, есть только методы класса, представленные ниже.

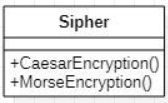


Рисунок 3. Диаграмма классов

## 2.5 Описание главного модуля

Главный модуль представляет собой файл main.cpp.

К главному модулю подключаются остальные модули, содержащие в себе классы и функции.

Листинг 1.

#include <iostream>

#include <fstream>

#include <string>

#include <Windows.h>

#include "Sipher.h"

int main()

{

//Изменение кодировки проекта на Windows 1251

system("chcp 1251");

system("cls");

std::cout << "Выберите метод шифрования:" << std::endl;

std::cout << "1. Шифр Цезаря." << std::endl;

std::cout << "2. Морзе." << std::endl;

std::cout << "3. Выход из программы." << std::endl;

std::cout << "Ответ: ";

//Авторская переменная

Cipher mloborev;

//Выбор метода шифрования

int choice;

std::cin >> choice;

switch (choice)

{

case 1:

mloborev.CaesarEncryption(); break;

case 2:

mloborev.MorseEncryption(); break;

case 3:

exit(0);

}

std::cout << std::endl;

system("pause");

exit(0);

}

## 2.6 Описания спецификаций к модулям

Спецификация к модулю CaesarCipher (Листинг 2).

Листинг 2.

void CaesarEncryption()

{

system("cls");

std::string message;

char ch;

int key;

//Запись строки из файла

std::ifstream filein("Unencrypted\_message.txt");

std::getline(filein, message);

filein.close();

//Проверка на наличие строки в файле

if (message.size() == 0)

{

std::cout << "Не обнаружена строка в файле Unencrypted\_message.txt\n\n";

system("pause");

exit(0);

}

wrongKey:

std::cout << "Введите ключ шифра от 1 до 5: ";

std::cin >> key;

if (key >= 1 && key <= 5)

{

for (int i = 0; message[i] != '\0'; ++i)

{

ch = message[i];

//Посимвольное шифрование строки для строчных букв

if (ch >= 'a' && ch <= 'z')

{

ch = ch + key;

if (ch > 'z')

{

ch = ch - 'z' + 'a' - 1;

}

message[i] = ch;

}

//И для заглавных букв

else if (ch >= 'A' && ch <= 'Z')

{

ch = ch + key;

if (ch > 'Z')

{

ch = ch - 'Z' + 'A' - 1;

}

message[i] = ch;

}

}

}

//Проверка на ввод верного символа

else

{

std::cout << "Неверный ключ. Значение не должно выходить за заданные рамки.\n";

system("pause");

system("cls");

//Очистка буфера

std::cin.clear();

std::cin.ignore(32767, '\n');

goto wrongKey;

}

system("cls");

std::cout << "Сообщение зашифровано в файл Encrypted\_message.txt\n";

//Запись зашифрованной строки в файл

std::ofstream fileout("Encrypted\_message.txt");

fileout << message;

}

## 

## 2.7 Описание модулей

В данной главе описаны используемые модули.

1. Главный модуль

Главный модуль представляет собой файл main.cpp. К главному модулю подключается модуль Sipher, содержащий в себе класс и функции.

1. Модуль, содержащий класс Cipher (Cipher.h)

Класс Cipher предназначен для работы шифрования. В него входят следующие методы:

* метод, предназначенный для шифрования строки методом Цезаря
* метод, предназначенный для перевода строки в азбуку Морзе

## 2.8 Описание тестовых наборов модулей

В этом разделе будут продемонстрированы результаты тестирования "черного ящика".

Тест 1. Шифрование методом Цезаря

Действия: Внести незашифрованную строку в файл Unencrypted\_message.txt, выбрать шифрование методом Цезаря, выбрать ключ шифрования.

Ожидаемый результат: Получение зашифрованной строки в файле Encrypted\_message.txt.

Результат теста:

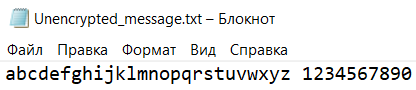


Рисунок 4. Внесение незашифрованного сообщения в файл

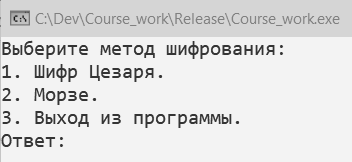


Рисунок 5. Выбор метода шифрования

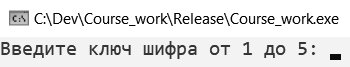


Рисунок 6. Выбор ключа шифрования

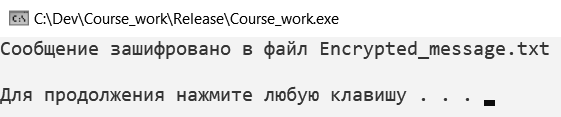


Рисунок 7. Запись зашифрованного сообщения в файл

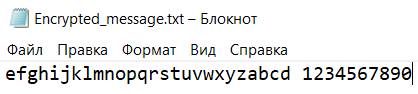


Рисунок 8. Получение зашифрованного сообщения

Тест 2. Шифрование методом Морзе

Действия: Внести незашифрованную строку в файл Unencrypted\_message.txt, выбрать шифрование методом Морзе.

Ожидаемый результат: Получение зашифрованной строки в файле Encrypted\_message.txt.

Результат теста:

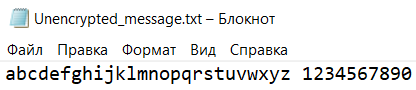


Рисунок 9. Внесение незашифрованного сообщения в файл

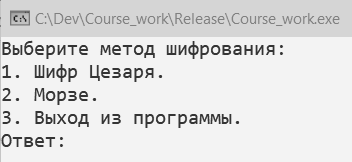


Рисунок 10. Выбор метода шифрования

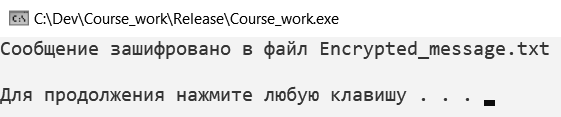


Рисунок 11. Запись зашифрованного сообщения в файл



Рисунок 12. Получение зашифрованного сообщения

## 2.9 Описание применения средств отладки

В ходе написания курсового проекта было получено следующее сообщение:

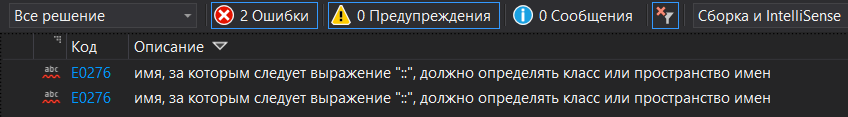


Рисунок 13. Обнаружена ошибка

Суть ошибки заключается в неправильном использовании оператора “точка”. Вместо точки был написан другой оператор - ::, из-за чего элементы класса не могли работать, поскольку не были привязаны к переменной класса.

Для решения данной ошибки нужно перейти в файл main.cpp и обратить внимание на 27 и 29 строки (рис 16).

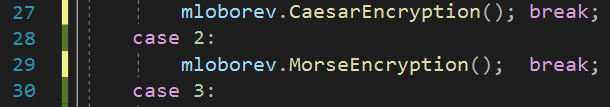


Рисунок 14. Ошибка в файле main.cpp

Необходимо изменить строку mloborev::CaesarEncryption(); break; на строку mloborev.CaesarEncryption(); break; и mloborev::MorseEncryption(); break; на строку mloborev.MorseEncryption(); break;, тем самым указав компилятору, что переменная класса связана с методом.

Результатом отладки стало следующее сообщение:



Рисунок 15. Результат отладки программы

Анализ оптимальности использования памяти и быстродействия

В данном разделе представлен анализ оптимальности использования памяти и быстродействия программы.

Принятые оптимальные решения:

* В целях оптимизации кода, было принято решение часто используемые блоки кода сделать функциями. Примеры таких функций представлены в Листинге 4.

Листинг 4.

void print() //вывод таблицы с данными на экран

{

for (int i = 0; i < dollar.size(); i++)

{

cout << i + 1 << ". ";

dollar[i].m\_print\_name();

cout << " ";

dollar[i].m\_print\_value();

cout << endl;

}

}

void vector\_to\_file() //запись таблицы с данными в файл "filename"

{

ofstream fin(filename);

for (int i = 0; i < dollar.size(); i++)

fin << dollar[i].m\_name << " " << dollar[i].m\_value << endl;

fin.close();

}

# Глава 3. Эксплуатационная часть

## Руководство оператора

## 3.1 Назначение программы

Функциональное назначение программы

Основной функцией проекта "Cipher" является автоматизация процесса шифрования текста для последующего его использования в других сферах жизни.

Эксплуатационное назначение программы

Программное обеспечение "Cipher" может быть использовано пользователями, желающими общаться тайно.

Состав функций

Функции пользователя:

1. Шифрование строки методом Цезаря с возможностью выбора ключа шифра и вывод её в файл
2. Перевод строки в азбуку Морзе и вывод её в файл

## 3.2 Условия выполнения программы

Минимальный состав аппаратных средств

Минимальный состав используемых технических (аппаратных) средств:

• Windows XP, 7, 8.1, 10;

• Процессор с тактовой частотой не ниже 2 ГГц.

• 2 МБ ОЗУ;

• Место на жестком диске: до 40 КБ;

• Видеоадаптер с минимальным разрешением 144p.

Требования к персоналу (Пользователю)

Элементарный навык работы с компьютером

## 3.3 Выполнение программы

Программа состоит из трёх файлов: "Cipher.exe", "Encrypted\_message.txt", "Unencrypted\_message.txt". В текстовых файлах находятся: незашифрованное сообщение и зашифрованное сообщение. Все эти файлы должны находиться в одном каталоге для корректной работы.

При запуске программы должно открыться следующее окно:

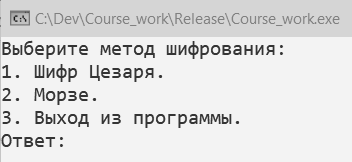


Рисунок 16. Главное окно программы

Выполнение программы

Для шифрования строки нужно нажать клавишу 1 или 2. Появится строка, сообщающая, что строка успешно зашифрована и скопирована в файл.

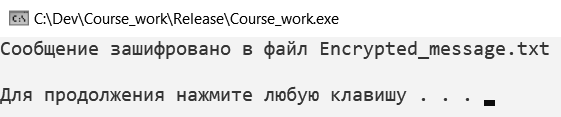


Рисунок 17. Окно успешного выполнения программы

Если выбрано шифрование методом Цезаря, будет предложено ввести ключ.

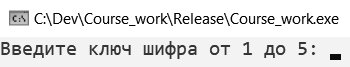


Рисунок 18. Окно режима редактирования

# Заключение

В результате выполнения курсового проекта была написана программа "Cipher" для автоматизации процесса шифрования текста.

В ходе работы были проанализированы предметная область, существующие разработки, посвященные данному направлению, получены практические навыки с работой в Microsoft Visual Studio.

Также планируется продолжить работу над данным проектом с целью расширения возможностей и удобства для пользователей. Планы по доработке представлены ниже.

1. Реализация новых методов шифрования;
2. Создание и последующая доработка интерфейса;

# 

# Список литературы и интернет-источников

Теоретические материалы:

1. <https://tproger.ru/translations/10-codes-and-ciphers/>
2. <https://zen.yandex.ru/media/tproger/kak-obscatsia-taino-5-shifrov-i-kodov-kotorye-mojno-ispolzovat-v-obychnoi-jizni-5cdd5e131d586800b3879cdf>
3. <https://habr.com/ru/post/444176/>
4. <https://cryptoperson.ru/cryptography/shifrovanie-teksta>

# 

# Приложения

**Приложение 1: Код главного модуля main.cpp**

#include <iostream>

#include <fstream>

#include <string>

#include <Windows.h>

#include "Cipher.h"

int main()

{

//Изменение кодировки проекта на Windows 1251

system("chcp 1251");

system("cls");

std::cout << "Выберите метод шифрования:" << std::endl;

std::cout << "1. Шифр Цезаря." << std::endl;

std::cout << "2. Морзе." << std::endl;

std::cout << "3. Выход из программы." << std::endl;

std::cout << "Ответ: ";

//Авторская переменная

Cipher mloborev;

//Выбор метода шифрования

int choice;

std::cin >> choice;

switch (choice)

{

case 1:

mloborev.CaesarEncryption(); break;

case 2:

mloborev.MorseEncryption(); break;

case 3:

exit(0);

}

std::cout << std::endl;

system("pause");

exit(0);

}

**Приложение 2: Код модуля Cipher.h**

#pragma once

class Cipher

{

public:

void CaesarEncryption()

{

system("cls");

std::string message;

char ch;

int key;

//Запись строки из файла

std::ifstream filein("Unencrypted\_message.txt");

std::getline(filein, message);

filein.close();

//Проверка на наличие строки в файле

if (message.size() == 0)

{

std::cout << "Не обнаружена строка в файле Unencrypted\_message.txt\n\n";

system("pause");

exit(0);

}

wrongKey:

std::cout << "Введите ключ шифра от 1 до 5: ";

std::cin >> key;

if (key >= 1 && key <= 5)

{

for (int i = 0; message[i] != '\0'; ++i)

{

ch = message[i];

//Посимвольное шифрование строки для строчных букв

if (ch >= 'a' && ch <= 'z')

{

ch = ch + key;

if (ch > 'z')

{

ch = ch - 'z' + 'a' - 1;

}

message[i] = ch;

}

//И для заглавных букв

else if (ch >= 'A' && ch <= 'Z')

{

ch = ch + key;

if (ch > 'Z')

{

ch = ch - 'Z' + 'A' - 1;

}

message[i] = ch;

}

}

}

//Проверка на ввод верного символа

else

{

std::cout << "Неверный ключ. Значение не должно выходить за заданные рамки.\n";

system("pause");

system("cls");

//Очистка буфера

std::cin.clear();

std::cin.ignore(32767, '\n');

goto wrongKey;

}

system("cls");

std::cout << "Сообщение зашифровано в файл Encrypted\_message.txt\n";

//Запись зашифрованной строки в файл

std::ofstream fileout("Encrypted\_message.txt");

fileout << message;

}

void MorseEncryption()

{

system("cls");

std::string message;

char ch;

//Запись строки из файла

std::ifstream filein("Unencrypted\_message.txt");

std::getline(filein, message);

filein.close();

//Проверка на наличие строки в файле

if (message.size() == 0)

{

std::cout << "Не обнаружена строка в файле Unencrypted\_message.txt\n\n";

system("pause");

exit(0);

}

std::ofstream fileout("Encrypted\_message.txt");

std::cout << "Сообщение зашифровано в файл Encrypted\_message.txt\n";

for (int i = 0; message[i] != '\0'; ++i)

{

ch = message[i];

//Посимвольная шифровка сообщения для строчных букв

if (ch >= 'a' && ch <= 'z')

{

switch (ch)

{

case 'a':

fileout << ".- "; break;

case 'b':

fileout << "-... "; break;

case 'c':

fileout << "-.-. "; break;

case 'd':

fileout << "-.. "; break;

case 'e':

fileout << ". "; break;

case 'f':

fileout << "..-. "; break;

case 'g':

fileout << "--. "; break;

case 'h':

fileout << ".... "; break;

case 'i':

fileout << ".. "; break;

case 'j':

fileout << ".--- "; break;

case 'k':

fileout << "-.- "; break;

case 'l':

fileout << ".-.. "; break;

case 'm':

fileout << "-- "; break;

case 'n':

fileout << "-. "; break;

case 'o':

fileout << "--- "; break;

case 'p':

fileout << ".--. "; break;

case 'q':

fileout << "--.- "; break;

case 'r':

fileout << ".-. "; break;

case 's':

fileout << "... "; break;

case 't':

fileout << "- "; break;

case 'u':

fileout << "..- "; break;

case 'v':

fileout << "...- "; break;

case 'w':

fileout << ".-- "; break;

case 'x':

fileout << "-..- "; break;

case 'y':

fileout << "-.-- "; break;

case 'z':

fileout << "--.. "; break;

}

}

//Для заглавных букв

else if (ch >= 'A' && ch <= 'Z')

{

switch (ch)

{

case 'A':

fileout << ".- "; break;

case 'B':

fileout << "-... "; break;

case 'C':

fileout << "-.-. "; break;

case 'D':

fileout << "-.. "; break;

case 'E':

fileout << ". "; break;

case 'F':

fileout << "..-. "; break;

case 'G':

fileout << "--. "; break;

case 'H':

fileout << ".... "; break;

case 'I':

fileout << ".. "; break;

case 'J':

fileout << ".--- "; break;

case 'K':

fileout << "-.- "; break;

case 'L':

fileout << ".-.. "; break;

case 'M':

fileout << "-- "; break;

case 'N':

fileout << "-. "; break;

case 'O':

fileout << "--- "; break;

case 'P':

fileout << ".--. "; break;

case 'Q':

fileout << "--.- "; break;

case 'R':

fileout << ".-. "; break;

case 'S':

fileout << "... "; break;

case 'T':

fileout << "- "; break;

case 'U':

fileout << "..- "; break;

case 'V':

fileout << "...- "; break;

case 'W':

fileout << ".-- "; break;

case 'X':

fileout << "-..- "; break;

case 'Y':

fileout << "-.-- "; break;

case 'Z':

fileout << "--.. "; break;

}

}

//Для цифр

else if (ch >= '0' && ch <= '9')

{

switch (ch)

{

case '1':

fileout << ".---- "; break;

case '2':

fileout << "..--- "; break;

case '3':

fileout << "...-- "; break;

case '4':

fileout << "....- "; break;

case '5':

fileout << "..... "; break;

case '6':

fileout << "-.... "; break;

case '7':

fileout << "--... "; break;

case '8':

fileout << "---.. "; break;

case '9':

fileout << "----. "; break;

case '0':

fileout << "----- "; break;

}

}

//Для пробела

else if (ch == ' ')

fileout << " ";

}

}

};