Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 6

По дисциплине “Основы программной инженерии”

На тему “Модульное программирование**”**

Выполнил:

Студент 1 курса 6 группы

Кравченко Сергей Сергеевич

Преподаватель: Наркевич А.С

2023, Минск

1. Используйте при выполнении лабораторной работы материал лекции 7.

2. Отчет по лабораторной работе оформить в виде документа в MS Word.

3. Задание. Для задачи из п.5 лабораторной работы 5:

1) Дополнительно предусмотреть возможность ввода с клавиатуры нескольких символов последовательно.

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

unsigned char code;

char c;

int n = 0;

int m = 0;

int u = 0;

setlocale(LC\_CTYPE, "Russian");

while (n != 4) {

cout << "Выберите вариант использования программы" << endl;

cout << "1 – определение разницы значений кодов в ASCII буквы в прописном и строчном написании, если введен символ латинского алфавита, иначе вывод сообщения об ошибке " << endl;

cout << "2 – определение разницы значений кодов в Windows-1251 буквы в прописном и строчном написании, если введен символ русского алфавита, иначе вывод сообщения об ошибке " << endl;

cout << "3 – вывод в консоль кода символа, соответствующего введенной цифре, иначе вывод сообщения об ошибке " << endl;

cout << "4 – выход из программы." << endl;

cin >> n;

cout << "Введите количество символов: ";

cin >> u;

switch (n)

{

case 1:

{

for (int i = 1; i < (u+1);)

{

cout << "Введите символ ";

cin >> code;

if (code >= 0x61 && code <= 0x7A)

{

printf("Это латинская буква %c, код Windows-1251 = %X\n", code, code);

code = code - 0x20;

printf("Прописная буква %c, ", code);

cout << endl;

}

else if (code >= 0x41 && code <= 0x5A)

{

printf("Это латинская буква %c, код Windows-1251 = %X\n", code, code);

code = code + 0x20;

printf("Строчная буква %c, ", code);

cout << endl;

}

else

{

cout << "Ошибка" << endl;

}

i += 1;

}

}break;

case 2:

{

for (int i = 1; i < (u + 1);) {

cout << "Введите символ ";

cin >> code;

if (code >= 0xE0 && code <= 0xFF)

{

printf("Это русская буква %c, код Windows-1251 = %X\n", code, code);

code = code - 0x20;

printf("Прописная буква %c, ", code);

cout << endl;

}

else if (code >= 0xC0 && code <= 0xDF)

{

printf("Это русская буква %c, код Windows-1251 = %X\n", code, code);

code = code + 0x20;

printf("Строчная буква %c, ", code);

cout << endl;

}

else

{

cout << "Ошибка" << endl;

}

i += 1;

}

}break;

case 3:

{

for (int i = 1; i < (u + 1);) {

cout << "Введите цифру ";

cin >> code;

if (code >= 0x30 && code <= 0x39)

{

printf("Это цифра %c, код Windows-1251 = %X\n", code, code);

}

else {

cout << "Ошибка" << endl;

}

i += 1;

}

}break;

}

}

}

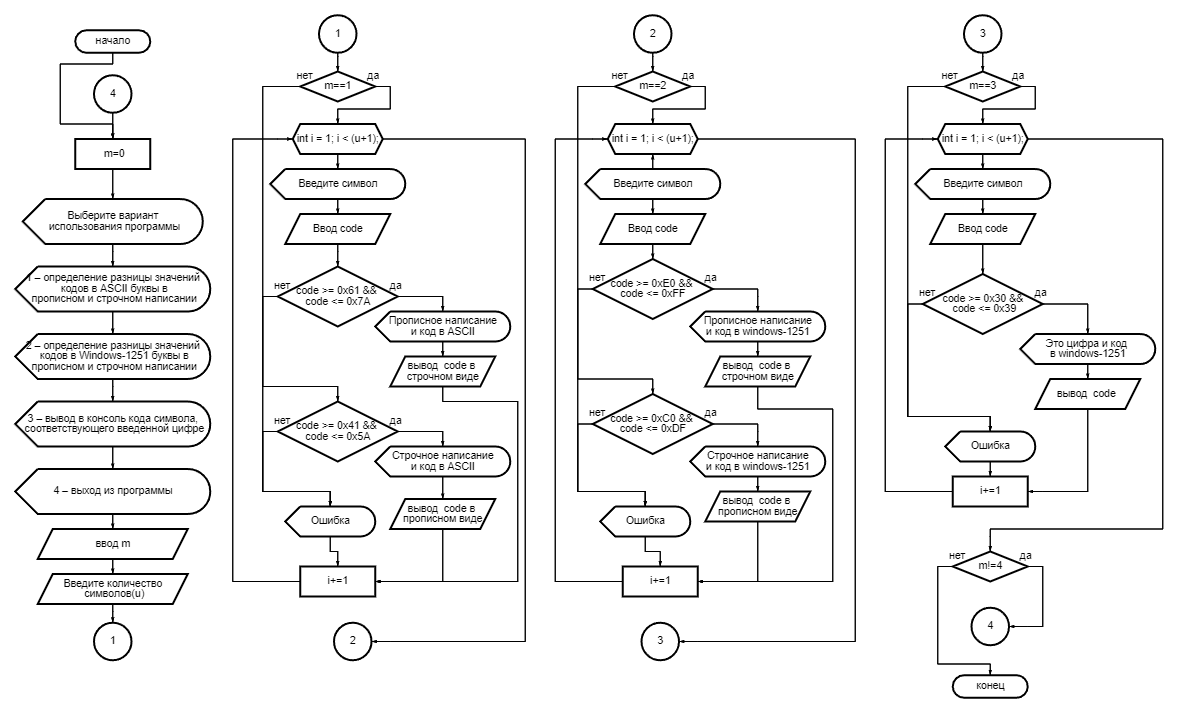
2) Выполнить постановку задачи - создать диалоговую программу, которая в зависимости от выбранного варианта использования выполняет действия пунктов 1, 2, 3 для любого количества введенных с клавиатуры символов.

3) Определить входные, выходные данные.

Входные: (номер варианта использования программы, количество символов, символ)

Выходные:(символы букв в строчном и прописном вариантах, код символов в кодировках, варианты использования программы, ошибка)

4) Записать алгоритм её решения в виде блок-схемы.



5) Разбить программу на модули. Описать состав, назначение, входные/выходные данные и алгоритм (любым способом) каждого модуля.

Код программы разбитой на модули

#include <iostream>

#include <Windows.h>

#include "case1.h"

#include "case2.h"

#include "case3.h"

using namespace std;

int main()

{

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

unsigned char code;

char c;

int n = 0;

int m = 0;

int u = 0;

setlocale(LC\_CTYPE, "Russian");

while (n != 4) {

cout << "Выберите вариант использования программы" << endl;

cout << "1 – определение разницы значений кодов в ASCII буквы в прописном и строчном написании, если введен символ латинского алфавита, иначе вывод сообщения об ошибке " << endl;

cout << "2 – определение разницы значений кодов в Windows-1251 буквы в прописном и строчном написании, если введен символ русского алфавита, иначе вывод сообщения об ошибке " << endl;

cout << "3 – вывод в консоль кода символа, соответствующего введенной цифре, иначе вывод сообщения об ошибке " << endl;

cout << "4 – выход из программы." << endl;

cin >> n;

cout << "Введите количество символов: ";

cin >> u;

switch (n)

{

case 1:

Case1(u);

break;

case 2:

Case2(u);

break;

case 3:

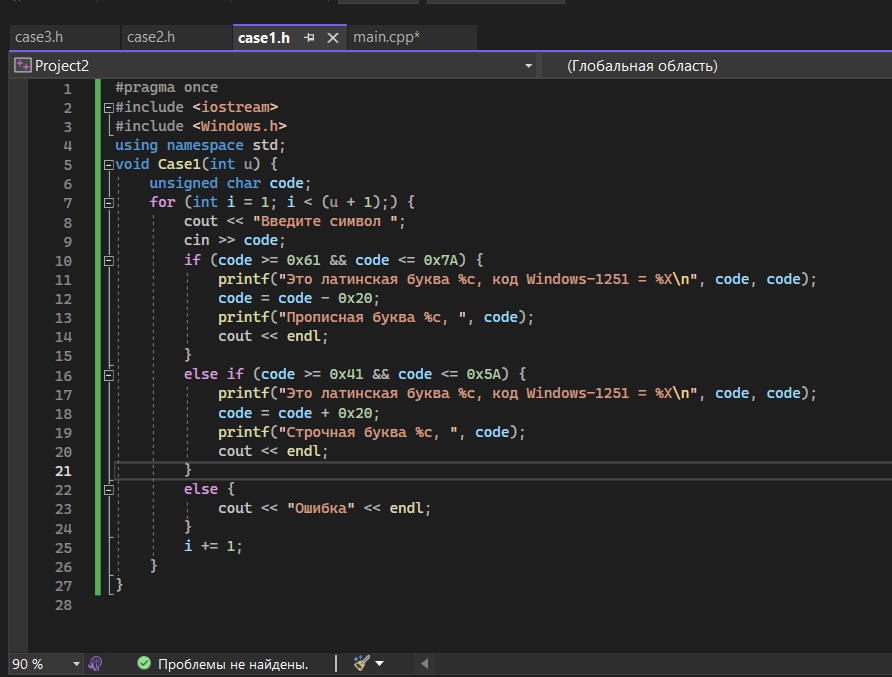
Case3(u);

break;

}

}

}



**Первый модуль**

**Назначение модуля** – модуль служит для определения кода введенного символа латинского алфавита строчных и прописных букв в кодировке Windows-1251

**Входные данные** – символ

**Выходные данные** – код строчных и прописных символов в кодировке Windows-1251

**Словесно-формульное описание алгоритма**

1. Функция `Case1` принимает один аргумент `u`, который определяет количество символов для обработки.

2. Объявляется переменная `code` для хранения вводимого символа.

3. Затем запускается цикл `for`, который выполняется `u` раз.

4. В каждой итерации цикла пользователю предлагается ввести символ.

5. Получить код символа в кодировке Windows-1251.

6. Проверить, если код символа в диапазоне от 0x41 до 0x5A (включительно), то переходим к пункту 9, иначе переходим к пункту 7.

Этот диапазон соответствует прописным латинским буквам в кодировке Windows-1251,

7. Проверить, если код символа в диапазоне от 0x61 до 0x7A (включительно), то переходим к пункту 8.

Этот диапазон соответствует строчным латинским буквам в кодировке Windows-1251.

8. Добавить значение 0x20, чтобы получить код соответствующей строчной буквы.

9.Отнять значение 0x20, чтобы получить код соответствующей строчной буквы.

10. Если введенный символ не попадает ни в один из указанных диапазонов, то выводится сообщение об ошибке.

11. После обработки каждого символа счетчик `i` увеличивается на 1, и цикл продолжается до тех пор, пока не будут обработаны все `u` символов.

**ПСЕВДОКОД**

НАЧАЛО

НАЧАЛО ЦИКЛА

ДЛЯ i <= (u+1) с шагом 1

ПОВТОРЯТЬ

ВВОД code

ЕСЛИ <символ находится в диапазоне от 0x41 до 0x5A (включительно)> ТО <код\_строчной\_буквы = код\_прописной\_буквы + 0x20>

ВЫВОД код\_строчной\_буквы

ИНАЧЕ

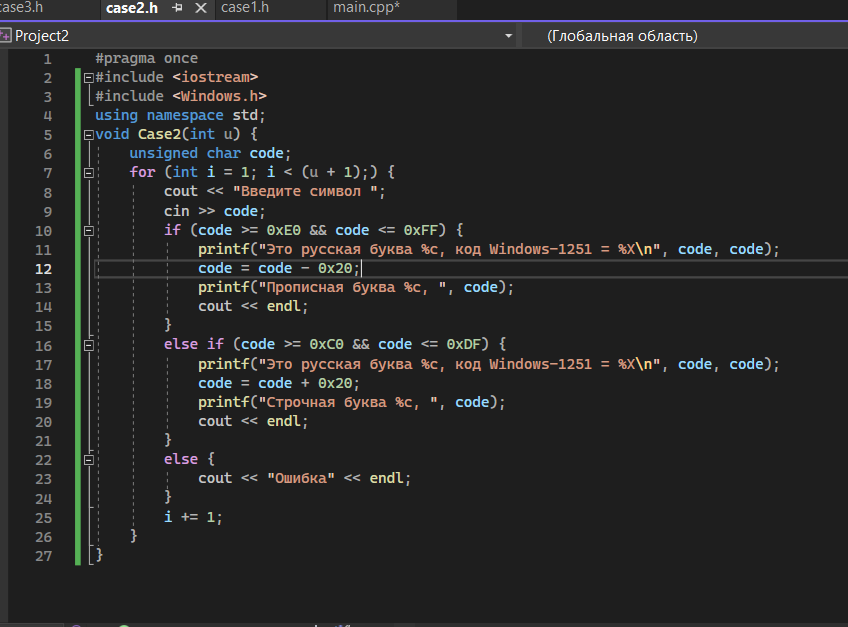
ЕСЛИ <символ находится в диапазоне от 0x61 до 0x7A (включительно)> ТО <код\_прописной\_буквы = код\_строчной\_буквы - 0x20>

ВЫВОД код\_прописной\_буквы

ИНАЧЕ ВЫВОД <Ошибка>

КОНЕЦ

**Второй модуль**



**Назначение модуля** – модуль служит для определения кода введенного символа кириллицы строчных и прописных букв в кодировке Windows-1251

**Входные данные** – символ

**Выходные данные** – код строчных и прописных символов в кодировке Windows-1251

**Словесно-формульное описание алгоритма**

1. Функция `Case1` принимает один аргумент `u`, который определяет количество символов для обработки.

2. Объявляется переменная `code` для хранения вводимого символа.

3. Затем запускается цикл `for`, который выполняется `u` раз.

4. В каждой итерации цикла пользователю предлагается ввести символ.

5. Получить код символа в кодировке Windows-1251.

6.Проверить, если код символа в диапазоне от 0x410 до 0x42F (включительно), то переходим к пункту 9, иначе переходим к пункту 7

Этот диапазон соответствует прописным буквам кириллицы в кодировке Windows-1251

7.Проверить, если код символа в диапазоне от 0x430 до 0x44F (включительно), то переходим к пункту 8. Этот диапазон соответствует строчным буквам кириллицы в кодировке Windows-1251

8.Добавить значение 0x20, чтобы получить код соответствующей строчной буквы.

9.Отнять значение 0x20, чтобы получить код соответствующей строчной буквы.

10. Если введенный символ не попадает ни в один из указанных диапазонов, то выводится сообщение об ошибке.

11. После обработки каждого символа счетчик `i` увеличивается на 1, и цикл продолжается до тех пор, пока не будут обработаны все `u` символов.

**ПСЕВДОКОД**

НАЧАЛО

НАЧАЛО ЦИКЛА

ДЛЯ i <= (u+1) с шагом 1

ПОВТОРЯТЬ

ВВОД code

ЕСЛИ <символ находится в диапазоне от 0x410 до 0x42F (включительно)> ТО <код\_строчной\_буквы = код\_прописной\_буквы + 0x20>

ВЫВОД код\_строчной\_буквы

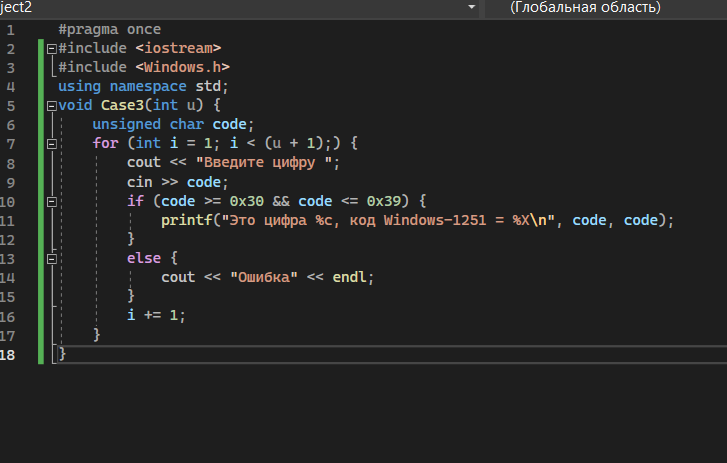
ИНАЧЕ

ЕСЛИ <символ находится в диапазоне от 0x430 до 0x44F (включительно)> ТО <код\_прописной\_буквы = код\_строчной\_буквы - 0x20>

ВЫВОД код\_прописной\_буквы

ИНАЧЕ ВЫВОД <Ошибка>

КОНЕЦ

**Третий модуль** 

**Назначение модуля** – модуль служит для определения кода введенной цифры в кодировке Windows-1251

**Входные данные** – символ

**Выходные данные** – код цифры в кодировке Windows-1251

**Словесно-формульное описание алгоритм**

1. Функция `Case1` принимает один аргумент `u`, который определяет количество символов для обработки.

2. Затем запускается цикл `for`, который выполняется `u` раз.

3. Вводим число

4.Если код символа с находится в диапазоне от 0x30 до 0x39 то п.5, иначе п.6

5.Вывести на дисплей " Символ и код в Windows-1251:"

6. Если введенный символ не попадает ни в один из указанных диапазонов, то выводится сообщение об ошибке.

7. После обработки каждого символа счетчик `i` увеличивается на 1, и цикл продолжается до тех пор, пока не будут обработаны все `u` символов.

**Псевдокод**

НАЧАЛО

НАЧАЛО ЦИКЛА

ДЛЯ i<= (u+1) с шагом 1

ПОВТОРЯТЬ

ВВОД code

ЕСЛИ <код символа с находится в диапазоне от 0x30 до 0x39 > ТО

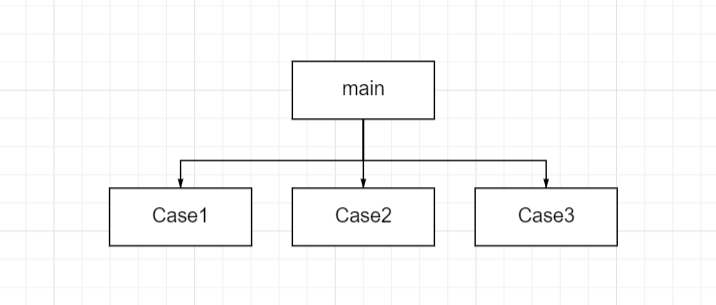
ВЫВОД <"Символ и код в Windows-1251:">

ИНАЧЕ

ВЫВОД <Ошибка>

КОНЕЦ

6) Выполнить нисходящее проектирование программы. Составить модульную схему программы и описать ее, используя псевдокод.



Дополнительное задание:

1. Разработать программу на С++ в стиле модульного программирования. Проверить ее работоспособность.

2. Для задания по своему варианту 4-ой лабораторной работы

Составить алгоритм планирования выходного дня студентом: если будет хорошая погода, студент пойдет гулять, а если плохая − будет писать реферат, пообедает и будет писать реферат. (Входные данные: информация о погоде; выходные данные: результат прошедшего выходного дня)

Составить алгоритм планирования выходного дня студентом

− определить входные, выходные данные

Входные – информация о погоде

Выходные – результат прошедшего дня

− разбить программу на модули и описать их состав, назначение, входные/выходные данные и привести алгоритм (любым способом) каждого модуля;

**Основной файл**

#include <iostream>

using namespace std;

#include "case1.h"

#include "case2.h"

int main() {

char weather;

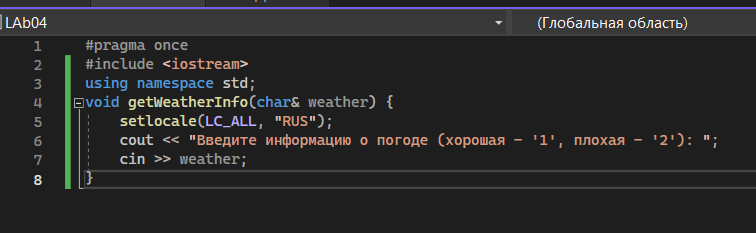
getWeatherInfo(weather);

handleWeather(weather);

return 0;

}

**Первый блок**

****

**Назначение модуля** – модуль служит для определения информации о погоде

**Входные данные** – информация о погоде

**Выходные данные** – нет

**Словесно-формульное описание алгоритм**

1. Выводится сообщение "Введите информацию о погоде (хорошая - '1', плохая - '2'): ", и программа ожидает ввода от пользователя.
2. Введенное пользователем значение записывается в переменную weather

**ПСЕВДОКОД**

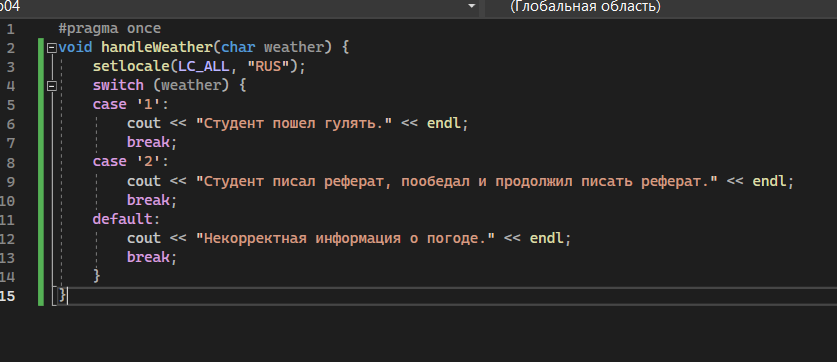
НАЧАЛО

ВЫВОД <Введите информацию о погоде (хорошая - '1', плохая - '2'):>

ВВОД weather

КОНЕЦ

**Второй блок**

****

**Состав модуля** –

**Назначение модуля** – модуль служит для определения кода введенной цифры в кодировке Windows-1251

**Входные данные** – номер switch

**Выходные данные** – сообщение “Студент пошел гулять”

сообщение “Студент писал реферат, пообедал и продолжил писать реферат”

сообщение ”Некорректная информация о погоде”

**Словесно-формульное описание алгоритм**

1) Функция использует оператор switch для обработки введенной информации о погоде.

2) Если weather равно '1', выводится сообщение "Студент пошел гулять.". 3)Если weather равно '2', выводится сообщение "Студент писал реферат, пообедал и продолжил писать реферат."

4)Если weather не равно ни '1', ни '2', выводится сообщение "Некорректная информация о погоде.".

**ПСЕВДОКОД**

НАЧАЛО

weather = получить\_информацию\_о\_погоде()

ЕСЛИ weather равно '1' ТО

ВЫВОД "Студент пошел гулять."

ИНАЧЕ ЕСЛИ weather равно '2' ТО

ВЫВОД "Студент писал реферат, пообедал и продолжил писать реферат."

ИНАЧЕ

ВЫВОД "Некорректная информация о погоде."

КОНЕЦ

− выполнить нисходящее проектирование программы. Составить модульную схему программы и описать ее, используя псевдокод.

