**Министерство образования Республики Беларусь**

**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Специальность Программная инженерия**

**Отчёт по лабораторной работе №5**

**По дисциплине**

**«Основы программной инженерии»**

**1 курс**

**10 группа**

**2 подгруппа**

Выполнил:

**Ярохович С.А.**

Проверила:

**Архипенко О.А.**

Минск

2023

**Цель работы**: ознакомление с основными парадигмами программирования.

**Краткие теоретические сведения:**

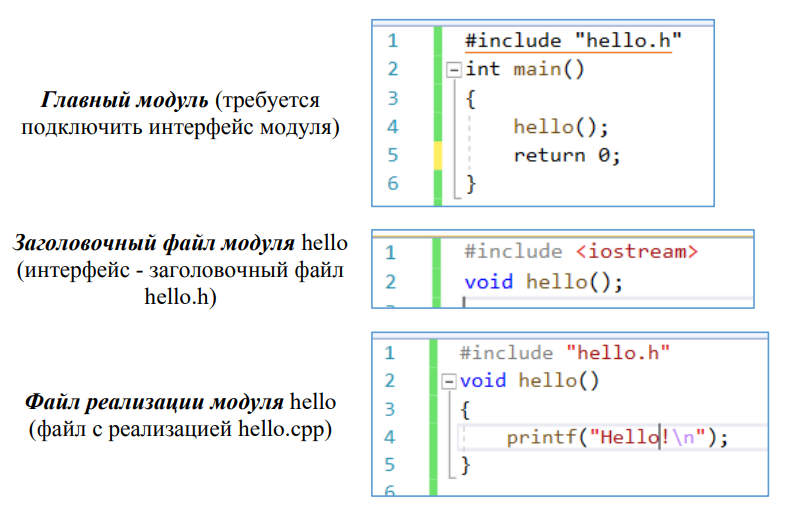
**Модульное программирование** – это организация программы как совокупности небольших независимых блоков, называемых модулями.

**Модуль** – функционально законченный фрагмент программы, оформленный в виде отдельного файла с исходным кодом.

**Функциональная декомпозиция задачи** – разбиение большой задачи на ряд более мелких, функционально самостоятельных подзадач – модулей. Каждый модуль в функциональной декомпозиции представляет собой «черный ящик» с одним входом и одним выходом.

**Модуль** – это фрагмент описания процесса, оформленный как самостоятельный программный продукт, пригодный для многократного использования.\

**Цели модульного программирования:** уменьшить сложность программ; предотвратить дублирование кода, упростить тестирование программы и обнаружение ошибок.



**Модуль, основные характеристики:**

− **один вход и один выход** (на вход программный модуль получает набор исходных данных, выполняет их обработку и возвращает набор выходных данных);

− **функциональная завершенность** (модуль выполняет набор определенных операций для реализации каждой отдельной функции, достаточных для завершения начатой обработки данных);

− **логическая независимость** (результат работы данного фрагмента программы не зависит от работы других модулей);

− **слабые информационные связи** с другими программными модулями (обмен информацией между отдельными модулями должен быть минимален);

− **размер и сложность программного элемента должна быть в разумных рамках.**

**Задание 3.2. Выполнить постановку задачи.**

Напишите диалоговую программу, которая в зависимости от выбранного варианта использования выполняет действия пунктов 1, 2, 3 для любого введенного с клавиатуры символа.

**Варианты использования:**

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

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

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

4 – выход из программы.

**Задание 3.4. Записать алгоритм её решения в виде блок-схемы.**

Блок-схема

**Начало**

Вывод: "1 – определение разницы значений кодов в ASCII буквы в прописном и строчном написании (символ латинского алфавита)

Вывод: "2 определение разницы значений кодов в Windows - 1251 буквы впрописном и строчном написании (символ русского алфавита)

**А**

**А**

Вывод:"3 – вывод в консоль кода символа, соответствующего введенной цифре

Вывод:"Вы ввели: "

Ввод: num

num

x + y +z < xyz

3

2

1

char arr\_1[3]{ };

char arr\_2[3]{ };

char arr1[3]{ };

char arr2[3]{ };

Вывод:"Введите символ"

int i = 0; i < 3; i++

int i = 0; i < 3; i++

Вывод:"Введите " << i + 1 << " пару: \n"

Вывод:"Введите " << i + 1 << " пару: \n"

**D2**

**D1**

**C2**

**C1**

**D**

**C**

**D2**

**D1**

**C2**

**C1**

**D**

**C**

**B**

Ввод: arr1[i] >> arr2[i];

Ввод: arr1[i] >> arr2[i];

char z

Ввод: z

int i = 0; i < 3; i++

int i = 0; i < 3; i++

Вывод:"Код символа: " << (int)z

x + y +z < xyz

x + y +z < xyz

(int)z >= -64 && (int)z <= -1

(int)z >= 65 && (int)z <= 122

**Нет**

**Да**

**Нет**

**Да**

(int)z1 >= -64 && (int)z1 <= -1

Вывод: "Error"

Вывод: "Error"

**Нет**

x + y +z < xyz

(int)z1>= 65 && (int)z1<= 122

**Нет**

x + y +z < xyz

**Да**

**Да**

Вывод: "Разница состовляет: " << z -z1

Вывод: "Разница состовляет: " << z1 - z

**Конец**

**Задание 3.3. Определить входные, выходные данные.**

**В первой части** программы (до вхождения программы в значения **switch**) входными данными **являются цифры**, введённые пользователем, в зависимости от которых пользователь попадает в соответствующий **case**.

**Во второй части** программы пользователь вводит **6 символов** **(1, 2 case)** или **1 символ** **(3 case)**, в результате этих действий пользователь получает **целочисленное значение** **(разница кодов или код символа)**.

**Задание 3.5.**

**Модуль Head**

**НАЧАЛО**

**ЕСЛИ** (символ => 65 и символ <=122) и (символ1 => 65 и символ1 <=122);

**НАЧАЛО БЛОКА**

**ВЫВОД**: “Разница состовляет: ”

**ВЫЧИСЛИТЬ** символ – символ1;

**ВЕРНУТЬ** символ – символ1;

**КОНЕЦ БЛОКА**

**ИНАЧЕ**

**НАЧАЛО БЛОКА**

**ВЫВОД:** «Error!»;

**КОНЕЦ БЛОКА**

**КОНЕЦ**

**Модуль Two**

**НАЧАЛО**

**ЕСЛИ** (символ => 65 и символ <=122);

**НАЧАЛО БЛОКА**

**ЕСЛИ** (символ1 => 65 и символ1 <=122);

**НАЧАЛО БЛОКА**

**ВЫВОД**: “Разница состовляет: ”

**ВЫЧИСЛИТЬ** символ – символ1;

**ВЕРНУТЬ** символ – символ1;

**КОНЕЦ БЛОКА**

**КОНЕЦ БЛОКА**

**ИНАЧЕ**

**НАЧАЛО БЛОКА**

**ВЫВОД:** «Error!»;

**КОНЕЦ БЛОКА**

**КОНЕЦ**

**Вывод:** таким образом, в ходе лабораторной работы был повторен материал об основных парадигмах программирования. Для создания программы был использован структурный стиль программирования. Для программы были определены входные/выходные данные.