Частное учреждение образования

«Колледж бизнеса и права»

УТВЕРЖДАЮ

Ведущий

методист колледжа

\_\_\_\_\_\_\_\_\_\_\_ Е.В. Паскал

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2021

|  |  |
| --- | --- |
| Специальность: 2-40 01 01 «Программное обеспечение информационных технологий» | Дисциплина: «Технология разработки программного обеспечения» |

# Лабораторная работа №20

**Инструкционно-технологическая карта**

Тема: Формирование модулей и связей между ними

Цель работы: Получить навыки разработки и отладки программных модулей, научиться анализировать полученные результаты с целью модернизации кода

Время выполнения: 2 часа

**Контрольныевопросы**

1. Что такое модульное программирование?
2. Перечислите и опишите преимущества использования подхода модульного программирования.
3. Приведите пример модуля.

**Теоретические сведения для выполнения работы**

Модульное программирование — это процесс разделения компьютерной программы на отдельные подпрограммы. Модуль — это отдельный программный компонент. Он часто может использоваться в различных приложениях и функциях с другими компонентами системы.

Некоторые программы могут иметь тысячи или миллионы строк, и управлять такими программами становится довольно сложно, поскольку в программе может быть слишком много синтаксических ошибок или логических ошибок, поэтому для управления программами такого типа подошла концепция модульного программирования .

Каждый субмодуль содержит что-то необходимое для выполнения только одного аспекта желаемой функциональности.

Модульное программирование делает упор на разбивке больших программ на мелкие задачи, чтобы повысить удобство сопровождения, удобочитаемость кода и сделать программу удобной для внесения любых изменений в будущем или для исправления ошибок.

Вопросы, о которых следует позаботиться до разработки модульной программы:

1. Ограничения каждого модуля должны быть решены.
2. Каким образом программа должна быть разбита на разные модули?
3. Связь между различными модулями кода для правильного выполнения всей программы.

Преимущества использования подхода модульного программирования —

1. Простота использования: этот подход обеспечивает простоту, поскольку вместо того, чтобы сосредоточиться на одном и том же коде тысяч и миллионов строк, мы можем получить к нему доступ в виде модулей. Это позволяет легко отлаживать код и меньше подвержен ошибкам.
2. Возможность повторного использования : позволяет пользователю повторно использовать функциональность с другим интерфейсом без повторного ввода всей программы.
3. Простота обслуживания. Это помогает уменьшить коллизии во время работы над модулями, помогая команде работать с надлежащей совместной работой при работе над большим приложением.

Пример модульного программирования на C

C называется структурированным языком программирования, потому что для решения большой проблемы язык программирования C делит проблему на более мелкие модули, называемые функциями или процедурами, каждый из которых выполняет определенную ответственность. Программа, которая решает всю проблему, представляет собой набор таких функций.

Модуль в основном представляет собой набор взаимосвязанных файлов, которые разделяют детали своей реализации, но скрывают его от внешнего мира. Как мы можем реализовать модульное программирование в c? Каждая функция, определенная в C по умолчанию, доступна глобально. Это можно сделать, включив заголовочный файл, в котором определяется реализация функции.  
Предположим, мы хотим объявить тип данных стека и одновременно скрыть реализацию, включая ее структуру данных, от пользователей. Мы можем сделать это, сначала определив общедоступный файл stack.h, который содержит общий тип данных стека данных и функции, которые поддерживаются типом данных стека.  
В заголовочный файл мы должны включать только определения констант, структур, переменных и функций с именем модуля, что позволяет легко определить источник определения в более крупной программе со многими модулями.  
Ключевые слова **[extern](http://espressocode.top/understanding-extern-keyword-in-c/)** и **[static](http://espressocode.top/static-variables-in-c/)** помогают в реализации модульности в C.

stack.h:

extern stack\_var1;

extern int stack\_do\_something(void);

Теперь мы можем создать файл с именем stack.c, который содержит реализацию типа данных стека:

stack.c

#include

int stack\_var1;

static int stack\_var2;

int stack\_do\_something(void)

{

stack\_var1 = 2;

stack\_var2 = 5;

}

Основной файл, который может включать в себя стек модулей

#include

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

while(1){

stack\_do\_something();

}

}

**Порядок выполнения работы**

1. Изучите теоретические сведения.
2. Разработайте 3 программных модуля по теме индивидуального задания. Модули должны соответствовать принципам, указанным в теоретической части.

**Домашнее задание**

Оформите отчет и ответьте на контрольные вопросы.

Преподаватель \_\_\_\_\_\_\_\_ А.С. Комаровский

Рассмотрено на заседании цикловой

комиссии ПОИТ №10

Протокол № от «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021

Председатель ЦК \_\_\_\_\_\_\_\_ В.Ю. Михалевич