

# Процедурное программирование на языке C — Семестр II

## Модули

- 1) Многофайловый проект. Модуль. Разделы описания и реализации, область видимости. Этапы получения исполняемого файла.

## Многофайловый проект

- 1) Автоматизация сборки проекта на примере утилиты `make`. Зависимости и правила.

## Динамическая память в C

- 1) Стек как структура данных.
- 2) Куча как структура данных.
- 3) Разделение памяти программы.
- 4) Работа с динамической памятью в C. Функции `malloc`, `calloc`, `realloc`, `free`.
- 5) Одномерные динамические массивы элементов простых типов или статических структур данных.
- 6) Одномерные динамические массивы указателей на одномерные динамические массивы элементов простых типов или статических структур данных.
- 7) Записи с указателями на динамическую память.
- 8) Рекурсивная деструкция.

## Классы памяти в C

- 1) Область видимости и время жизни.
- 2) Ключевое слово `auto`.
- 3) Ключевое слово `extern`.
- 4) Ключевое слово `static`.
- 5) Ключевое слово `register`.

## Матрицы

- 1) Размещение матриц по строкам и столбцам.
- 2) Хранение матрицы в виде указателя на массив указателей на массивы-строки.
- 3) Хранение матрицы в виде указателя на массив указателей на массивы-столбцы.
- 4) Хранение матрицы в виде указателя на массив указателей на подмассивы-строки внутри единого блока данных.

- 5) Хранение матрицы в виде указателя на массив указателей на подмассивы-столбцы внутри единого блока данных.
- 6) Хранение матрицы в виде указателя на объединённый массив элементов и указателей на массивы-строки.
- 7) Хранение матрицы в виде указателя на объединённый массив элементов и указателей на массивы-столбцы.

## **Элементы динамического программирования**

- 1) Рекурсивная подпрограмма.
- 2) Хвостовая рекурсия.
- 3) Мемоизация.
- 4) Рекурсия, хвостовая рекурсия и мемоизация на примере реализации функции факториала.
- 5) Рекурсия, хвостовая рекурсия и мемоизация на примере реализации функции Фибоначчи.

## **Подпрограммы**

- 1) Соглашение о вызовах, единство бинарного интерфейса.
- 2) Соглашение `cdecl`.
- 3) Соглашение `pascal`.
- 4) Соглашение `stdcall`.
- 5) Соглашение `fastcall`.
- 6) Реализация подпрограммы с переменным числом аргументов с использованием стека.
- 7) Реализация подпрограммы с переменным числом аргументов с использованием `va_args`.

## **Абстракция данных**

- 1) Указатель на `void`.
- 2) Указатель на функцию. Реализация сортировки с передаваемым указателем на функцию-компаратор.
- 3) Абстрактный тип данных. Определение применимости реализации к определённым типам и структурам данных, расширение, сужение области применимости.
- 4) Смещение, макрос `offsetof`. Абстрактные типы данных, построенные на смещениях. Списки в стиле Беркли.

## **Библиотеки**

- 1) Статические библиотеки.
- 2) Динамические библиотеки. Динамическая компоновка.
- 3) Динамические библиотеки. Динамическая загрузка.
- 4) Особенности взаимодействия библиотек и внешних ресурсов.

## **Структуры данных, построенные на узлах**

- 1) Узел как структура данных.
- 2) Односвязный линейный список.
- 3) Односвязный кольцевой список.
- 4) Двусвязный линейный список.
- 5) Двусвязный кольцевой список.
- 6) Реализация стека и очереди на базе списков.

## **Остальное**

- 1) Встраиваемые функции. Ключевое слово `inline`.
- 2) Побитовые операции.
- 3) Операции сдвига.
- 4) Неопределённое поведение.
- 5) Поведение, определяемое реализацией.
- 6) Директивы препроцессора.

## **Элементы событийного программирования**

- 1) Мастер-цикл.
- 2) Графический интерфейс на основе событийной парадигмы. Основные компоненты графического интерфейса.
- 3) Псевдопараллелизм. Таймеры.