

УЧЕБНАЯ ПРОГРАММА
по дисциплине
"ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ С++"

А.А.Быков

boombook@yandex.ru

Данный курс предназначен для студентов первого и второго годов обучения. На первом году изучаются главы 1–5, на втором году главы 6–10. В основу курса положен принцип практического освоения каждого изучаемого аспекта языка С++. Результат освоения каждой темы – работающая программа, написанная студентом по заданию. Форма отчетности – зачет или экзамен по каждой главе. Зачет получает студент, выполнивший все практические задания. Экзамен состоит из двух частей, теоретической и практической. Теоретическая часть содержит ориентировочно 10 вопросов по всем изучаемым темам, на каждый из которых дается письменный ответ. Состав вопросов теоретической части по пособию [1]. Практическая часть состоит в создании проекта в среде MS VS С++ по заданию экзаменатора.

Основная литература (доступна в виде документов pdf):

[1] Дейтел Х., Дейтел П. Программирование на С++.

✓ *Дейтел С++, §1.11,*

[2] Либерти Дж. Освой самостоятельно С++ за 21 день.

✓ *Либерти С++21, глава 1.*

[3] Круглински Д., Уингоу С., Шефферд Дж.

Программирование Microsoft Visual Studio.

✓ *Круглински, Уингоу, Шефферд, глава 1.*

Дополнительная литература:

[4] Страуструп Б., Программирование на С++. Специальное издание.

✓ *Страуструп С++, §1.11,*

[5] Янг М. Дж. Visual С++, полное руководство в 2-х т.

✓ *Янг, Том I, глава 1,*

[6] Либерти Дж. Энциклопедия С++.

✓ *Либерти С++ энциклопедия, глава 1.*

Глава 1 Основные понятия С++

Лекция 1. Основные понятия С++ и Microsoft Visual Studio

1.1 Понятие Microsoft Visual studio.

✓ *Круглински, Уингоу, Шефферд, глава 1.*

✓ *Янг, Том I, глава 1, 2,*

✓ *Дейтел С++, §1.15,*

1.1.1 **Инсталляция MS Visual Studio 2005 и MS VC++2005.**

1.1.2 **Среда программирования, компиляции и отладки. Понятие проекта.**

1.1.3 **Проекты Console Application.**

1.1.4 **Проекты win32.**

1.1.5 **Проекты MFC. Особенности проектов Dialog based, SDI, MDI.**

1.2 Понятие С++

✓ *Страуструп С++, §3.2.*

✓ *Дейтел С++, §1.14 – 1.16,*

✓ *Либерти С++21, глава 2.*

1.2.1 **Описание базового проекта, предназначенного для изучения С++.**

1.2.2 **Понятие строки.**

1.2.3 **Понятие файла. Имена и расширения.**

- 1.2.4 Понятие вывода информации в файл.
- 1.2.5 Вывод в файл текстовой строки.
- 1.2.6 Предопределенные объекты `myofs` и `myifs`.
- 1.2.7 Понятие манипулятора `endl`.

Практика 1. r1n01- Проект Visual C++. Вывод текста в файл.

Задание. Измените код так, чтобы вместо файла "protocol.txt" результатом работы кода был файл "Иванов И И группа 123 n01 протокол", где Иванов И И - Ваша фамилия и инициалы, 123 - номер учебной группы, 01 - номер задания. В дальнейшем данный файл будем называть "протокол". Измените код так, чтобы в начале отчета располагалась информация о Вас (Фамилия, имя отчество, номер учебной группы, номер задания, название задания, дата выполнения). В дальнейшем каждый отчет оформляйте аналогичным образом, указывая текущий номер задания и тему работы.

Лекция 2. Встроенные типы С++.

1.3 Встроенные типы С++.

- ✓ *Страуструн С++, §4.1-4.6.*
- ✓ *Либерти С++21, глава 3.*
- ✓ *Дейтел С++, §1.20,*

- 1.3.1 Понятие объекта. Объект как представитель класса (типа).
- 1.3.2 Понятие имени. Блок.
- 1.3.3 Основные атрибуты объекта: поля данных и методы (функции).
- 1.3.4 Основные ключевые точки кода, связанные с каждым объектом.
 - А) Объявление,
 - В) Определение,
 - С) Конструирование,
 - Д) Использование,
 - Е) Разрушение.
- 1.3.5 Понятие класса. Понятие типа (предопределенного системного класса).
- 1.3.6 Понятие поля данных.
- 1.3.7 Понятие функции. Основные действия, которые выполняет функция:
 - Ф) Изменение значения полей данных объекта,
 - Г) Изменение порядка исполнения операторов (функций),
 - Н) Изменение состояния внешних устройств.
- 1.3.8 Встроенные типы С++ и область их применения.
- 1.3.9 Понятие длины (size). Функция `sizeof(имя)`.

1.4 Арифметические типы.

- ✓ *Страуструн С++, §4.4-4.5.*
- ✓ *Дейтел С++, §1.18,*
- ✓ *Либерти С++21, глава 3.*

- 1.4.1 Арифметические типы.
- 1.4.2 Типы `int`, `unsigned int`, `long`, `unsigned long`.
- 1.4.3 Тип `bool`.
- 1.4.4 Типы `float`, `double`.
- 1.4.5 Типы `char`, `char[]`, `char*`.
- 1.4.6 Конструирование и инициализация объекта встроенного типа.
- 1.4.7 Переменные и константные объекты.
- 1.4.8 Понятие преобразования типов.

1.5 Арифметические операторы С++.

- ✓ *Страуструн С++, §6.2.*
- ✓ *Дейтел С++, §1.18, §2.11, §2.12,*
- ✓ *Либерти С++21, глава 4.*

- 1.5.1 Арифметические операции.
- 1.5.2 Операция копирования.
- 1.5.3 Операции сложения, умножения, вычитания, деления.
- 1.5.4 Операция вычисления остатка от деления нацело.
- 1.5.5 Операции инкремента ++ и декремента --.
- 1.5.6 Префиксные и постфиксные операторы.
- 1.5.7 Применение скобок.
- 1.5.8 Стандартные арифметические операторы. Функция `sqrt(double)`.

1.6 Преобразование арифметических типов.

- ✓ Страуструн C++, §B6.
- ✓ Дейтел C++, §1.18,
- ✓ Либерти C++21, глава 4.

- 1.6.1 Преобразование типов при выполнении арифметических операций.
- 1.6.2 Преобразование `int` в `double`.
- 1.6.3 Преобразование `double` в `int`. Функции `double floor(double)`, `double ceil(double)`.
- 1.6.4 Оператор `typedef`.

1.7 Вывод в файл значений арифметических типов.

- ✓ Страуструн C++, §21.2.

- 1.7.1 Вывод `int`, `double`.
- 1.7.2 Вывод с комментариями.

Практика 2. p1n02-Арифметические операторы.

Задание. Создайте несколько переменных типа `int`, инициализируйте, выполните операции сложения, вычитания, умножения, деления, инкремента. Выведите результаты в файл. Создайте несколько переменных типа `double`, инициализируйте, выполните операции сложения, вычитания, умножения, деления, инкремента, декремента. Выведите в файл протокола. Вычислите $\frac{1 + 2 + 3 + 4 + 5}{4}$, $\frac{1.0 + 2.0 + 3.0 + 4.0 + 5.0}{4.0}$. Объясните разницу.

Лекция 3. Операторы цикла.

1.8 Операторы цикла.

- ✓ Страуструн C++, §6.3.
- ✓ Дейтел C++, §2.7–§2.15, §2.17–§2.18,
- ✓ Либерти C++21, глава 4, 7.

- 1.8.1 Цикл `for`. Итератор (переменная цикла).
- 1.8.2 Инициализатор, условие, приращение.
- 1.8.3 О значении итератора после выхода из цикла `for`.
- 1.8.4 Применение счетчика числа итераций.
- 1.8.5 О надежности C++ кода: определение итератора внутри оператора `for`.
- 1.8.6 Применение цикла для вывода в файл текстовой строки.
- 1.8.7 Применение цикла для вывода в файл набора чисел.
- 1.8.8 Циклы с несколькими инициализаторами.
- 1.8.9 Циклы с несколькими операторами приращения.
- 1.8.10 Цикл `while`. Цикл `while(true)`.
- 1.8.11 Цикл `do`.
- 1.8.12 Операторы перехода `break`, `continue`. Выход из цикла с помощью `goto`.
- 1.8.13 Вложенные циклы.

Практика 3. p1n03 Операторы цикла

Задание. Используя оператор цикла `for`, найдите сумму всех натуральных чисел в пределах от 7 до 77. Выведите результаты в протокол с комментарием. С

помощью оператора while найдите наименьшее значение N , при котором

- 1) $\sum_{k=2}^N \frac{1}{k \ln k} > 10$,
- 2) $\sum_{k=1}^N \frac{1}{k(k+1)} > 0,999$,
- 3) $\sum_{k=1}^N \frac{1}{k!} > 1.718$,
- 4) $\sum_{k=1}^N \frac{4}{k(k+1)(k+2)} > 0,9999$.

Лекция 4. Логические операторы

1.9 Логические операторы С++.

- ✓ Страуструн С++, §6.2, §6.3.
- ✓ Дейтел С++, §2.4– §2.6, §2.16, §2.19– §2.20,
- ✓ Либерти С++21, глава 4.

- 1.9.1 Бинарная булевская функция == для встроенных типов.
- 1.9.2 Логический бинарный оператор ||.
- 1.9.3 Логический бинарный оператор &&.
- 1.9.4 Операторы if и if - else.
- 1.9.5 Оператор switch.
- 1.9.6 Совместное применение циклов и логических операторов.

1.10 Ввод текста и данных из файла.

- ✓ Страуструн С++, §3.6, 21.3.

- 1.10.1 Предопределенный объект freopen и его использование.
- 1.10.2 Ввод нескольких слов. Понятие буфера.
- 1.10.3 Понятие разделителя.
- 1.10.4 Ввод целых и вещественных значений.

1.11 Простой интерпретатор.

- 1.11.1 Программирование интерпретатора значений и операций, вводимых из файла.

Практика 4. r1p04 Ввод данных из файла и анализ величин

Задание. Сформируйте файл, содержащий несколько целых чисел. Введите значения переменных из этого файла, используя оператор >>. Найдите наименьшее, наибольшее. Напишите код для вывода некоторого текста, который определяется введенными значениями.

Практика 5. r1p05 Логические операторы

Задание. Используя оператор цикла for и логические операторы, найдите значение суммы всех натуральных чисел в пределах от 5 до 5555, которые делятся на 2, 3, 5, 11, но не делятся на 7. Напечатайте в файле протокола таблицу всех таких чисел.

Глава 2 Функции

Лекция 5. Функции

1.12 Функции.

- ✓ Страуструн С++, §7.1, §7.2, §7.3.
- ✓ Дейтел С++, §3.1– §3.16,
- ✓ Либерти С++21, глава 5.

- 1.12.1 Понятие функции.
- 1.12.2 Формальные параметры, возвращаемое значение.
- 1.12.3 Функция с пустым списком параметров.

- 1.12.4 **Функция, не возвращающая значения.**
- 1.12.5 **Объявление (прототип) и описание функции.**
- 1.12.6 **Оператор вызова функции, фактические параметры.**
- 1.12.7 **Передача параметра в функцию по значению.**

Практика 6. r2n06 Функции, передача параметра по значению

Задание. Напишите функцию, которая вычисляет сумму всех натуральных чисел в пределах от M до N, которые делятся на m, n, k, но не делятся на p, q, r. Параметры передавайте по значению. Напишите функцию `int mymax(int m, int n)`, вычисляющую большую из двух переменных типа `int`. Напишите аналогичную функцию, которая будет работать для 4 переменных и для 8 переменных. При этом используйте `mymax(...)`.

Лекция 6. Различные способы передачи параметров

1.13 Передача параметра по указателю.

- 1.13.1 **Передача входного параметра по указателю.**
- 1.13.2 **Передача выходного параметра по указателю.**
- 1.13.3 **Передача нескольких параметров по указателю.**
- 1.13.4 **Значение NULL.**
- 1.13.5 **Передача значения NULL для управления работой функции.**
- 1.13.6 **Значение параметра по умолчанию. Применение сравнения указателя с NULL.**

✓ *Страуструн C++, §7.1, §7.2, §7.3.*

Практика 7. Функции, передача параметра по указателю

Задание. Напишите функцию `f(x1, x2, x3, x4)`, которая осуществляет циклическую перестановку величин `int x1, x2, x3, x4` (т.е. значение `x1` заменяем на значение `x2`, значение `x2` заменяем на значение `x3`, ... значение `x4` заменяем на значение `x1`). Параметры передавайте через указатели. Проверьте, будет ли данная функция работать для фактических параметров `const`, например можно ли написать нечто типа `f(1, 2, 3, 4)`. Проверьте, будет ли данная функция работать для фактических параметров другого типа (например, `double`). Напишите функцию `f(x1, x2, x3, x4, n)`, которая осуществляет n-кратный вызов функции `f(x1, x2, x3, x4)`. Параметры `x1, x2, x3, x4` передайте через указатели, параметр `n` по значению. Напишите функцию `sort(x1, x2, x3, x4)`, которая осуществляет сортировку величин `int` типа `x1, x2, x3, x4` по возрастанию значений. Способ передачи параметров выберите самостоятельно.

1.14 Передача параметра в функцию по ссылке.

✓ *Страуструн C++, §7.1, §7.2, §7.3.*

✓ *Дейтел C++, §3.17.*

- 1.14.1 **Понятие ссылки. Описание, определение и инициализация.**
- 1.14.2 **Понятие типа ссылки.**
- 1.14.3 **Копирование ссылки.**
- 1.14.4 **Копирование ссылаемого объекта.**

Практика 8. Функции, передача параметра по ссылке

Задание. Напишите функцию `f(x1, x2, x3, x4)`, которая осуществляет циклическую перестановку величин `int x1, x2, x3, x4` (т.е. значение `x1` заменяем на значение `x2`, значение `x2` заменяем на значение `x3`, ... значение `x4` заменяем на значение `x1`). Параметры передавайте по ссылке. Проверьте, будет ли данная функция работать для фактических параметров `const`, например можно ли написать нечто типа `f(1, 2, 3, 4)`. Проверьте, будет ли данная функция работать для фактических параметров другого типа (например, `double`). Напишите функцию `f(x1, x2, x3, x4, n)`, которая осуществляет n-кратный вызов функции `f(x1, x2, x3, x4)`. Параметры `x1, x2, x3, x4` передайте по ссылке, параметр `n` по значению. Напишите функцию `sort(x1, x2, x3, x4)`, которая осу-

ществляет сортировку величин `int` типа `x1`, `x2`, `x3`, `x4` по возрастанию значений. Параметры передавайте по ссылке.

1.15 Защита операндов

✓ *Страуструп С++, §7.1, §7.2, §7.3.*

1.15.1 Константные функции и константные параметры.

1.16 Параметры по умолчанию.

✓ *Страуструп С++, §7.5.*

✓ *Дейтел С++, §3.18.*

✓ *Либерти С++21, глава 5.*

1.17 Перегруженные функции.

✓ *Страуструп С++, §7.4.*

✓ *Дейтел С++, §3.20.*

✓ *Либерти С++21, глава 10.*

1.18 Рекурсивные функции.

1.18.1 Рекурсивные функции

✓ *Дейтел С++, §3.20.*

Практика 9. r2n09 Перегруженные и рекурсивные функции

Задание. Напишите функцию `f(x1, x2, x3, x4)`, которая осуществляет циклическую перестановку величин `double x1, x2, x3, x4`. Пусть в одном файле определены функции `f(int& x1, ...)` и `f(double& x1, ...)`. Включите в код функции оператор выдачи сообщения в файл протокола. Определите, какая версия функции `f(...)` активизируется при выполнении операторов `f(int, int, int, int)`, `f(double, double, double, double)`, `f(int, double, int, double)`. Напишите функцию `int g(int x1, int x2, int x3)`, которая возвращает `x1+x2+x3`. Напишите функцию `int g(double x1, double x2, double x3)`, которая возвращает `x1*x2*x3`. Определите, какая версия функции `g(...)` активизируется при выполнении операторов `g(int, int, int)`, `g(double, double, double)`, `g(int, double, int)`, `g(int, double, char)`. Определите, какая версия функции `g(...)` активизируется при выполнении операторов `g(3, 4, 5)`, `g(3.0, 4.0, 5.0)`, `g('a', 2, 6.0)`.

Задание. Напишите функцию `s(x1, x2, x3, x4)`, которая вычисляет `x1+x2+x3+x4` и которая может быть также активизирована операторами `int m4=s(1, 2, 3, 4)`, `int m3=s(1, 2, 3)`, `int m2=s(1, 2)`, `int m1=s(1)`, `int m0=f()`. Возможно ли написать две перегруженные версии данной функции, одна из которых принимает аргументы типа `int`, а другая типа `double`?

Задание. Напишите рекурсивную функцию.

Глава 3 Массивы и указатели

Лекция 7. Одномерные массивы и указатели

1.19 Указатели.

✓ *Страуструп С++, §5.1, §5.2.*

✓ *Дейтел С++, §5.1– §5.7.*

✓ *Либерти С++21, глава 8, 12.*

1.19.1 Понятие указателя. Описание, определение и инициализация.

1.19.2 Адрес объекта и указатель на объект.

1.19.3 Понятие типа указателя.

1.19.4 Операция разыменовывания указателя.

1.19.5 Копирование указателя.

1.19.6 Копирование указываемого объекта.

1.19.7 Инкремент и декремент указателя.

1.19.8 Константные указатели и указатели на константные объекты.

1.19.9 Указатель на указатель.

Практика 10. р3п10 Указатели

Задание. Создайте и инициализируйте несколько переменных разных типов. Создайте и инициализируйте указатели на эти переменные. Исследуйте операцию копирования указателя. Исследуйте операцию копирования указываемых объектов через указатели на них. Продемонстрируйте разыменовывание указателя. Продемонстрируйте разницу при выводе в файл объекта и указателя на него. Выведите в файл указатель и разыменованный указатель с комментариями. Пусть `int m=7; int* pm=&m;` Найдите и выведите в файл значение величины `*pm`. Сделайте то же самое для величины `(*pm+1)`. Найдите и выведите в файл значение величины `pm`. Сделайте то же самое для величины `(pm+1)`. Объясните. Какое значение будут иметь величины `(*pm)++`, `*(pm++)`, `*pm++`?

1.20 Одномерные массивы

- ✓ *Страуструн С++, §5.1, §5.2.*
- ✓ *Дейтел С++, §4.1– §4.8.*
- ✓ *Либерти С++21, глава 8, 12.*

1.20.1 Понятие одномерного массива с фиксированной размерностью.

1.20.2 Доступ к элементам одномерного массива с помощью переменной с индексом.

1.20.3 Инициализация одномерного массива.

1.20.4 Неявное указание размера массива при инициализации.

1.20.5 Вывод значения массива в файл.

1.21 Одномерные массивы и указатели

- ✓ *Страуструн С++, §5.1, §5.3, §5.4.*
- ✓ *Дейтел С++, §4.1– §4.8.*
- ✓ *Либерти С++21, глава 8, 12.*

1.21.1 Операции над указателями.

1.21.2 Операции инкремента и декремента.

1.21.3 Доступ к элементам одномерного массива с помощью указателей.

1.21.4 Взаимосвязь указателей и массивов.

1.22 Передача одномерного массива в функцию.**Практика 11. р3п11 Одномерные массивы**

Задание. Создайте и инициализируйте несколько одномерных массивов разных типов с заданной размерностью. Исследуйте применение указателей и переменных с индексом для доступа к элементам массива. Выведите в файл.

Задание. Пусть имеется массив `int X[10]`, инициализированный последовательными значениями 1000, 1100, 1200, 1300 и т.д. Какие значения будут иметь величины `X`, `X[0]`, `X[7]`, `*X`, `*X+1`, `*X+7`, `*(X+7)`, `(*X)+7`. Какое значение будет иметь величины `X+10000`, `*(X+10000)`?

Задание. Пусть `int X[10]; int Y[10];` инициализируем массивы `X` и `Y` (значения придумайте сами). Что произойдет при исполнении операторов а) `X[7]=Y[7]`, б) `X[0]=Y[0]`, в) `X=Y`?

Лекция 8. Операторы new и delete.**1.23 Операторы new и delete.**

- ✓ *Страуструн С++, §6.2.6.*
- ✓ *Дейтел С++, §4.1– §4.8.*
- ✓ *Либерти С++21, глава 8.*

1.23.1 Применение операторов `new` и `delete` для создания переменной.

1.23.2 Применение операторов `new` и `delete` для создания одномерного массива.

1.23.3 Методика проверки корректности операции выделения памяти.

1.23.4 Понятие утечки памяти и способы диагностики.

Практика 12. р3п12 Операторы new и delete-1

- Задание.** Создайте и инициализируйте несколько одномерных массивов разных типов с помощью оператора `new`. Используйте оператор `delete` для разрушения. Исследуйте применение указателей и переменных с индексом для доступа к элементам массивов. Выведите в файл. Исследуйте последствия попытки использования элементов массива до создания и после разрушения.
- Задание.** Пусть имеется массив `int* X = new int[10]`, инициализированный последовательными значениями 1000, 1100, 1200, 1300 и т.д. Какие значения будут иметь величины `X`, `X[0]`, `X[7]`, `*X`, `*X+1`, `*X+7`, `*(X+7)`, `(*X)+7`. Какое значение будет иметь величины `X+10000`, `*(X+10000)`?
- Задание.** Пусть `int X=new int[10]; int Y=new int[10];` инициализируем массивы `X` и `Y` (значения придумайте сами). Что произойдет при исполнении операторов а) `X[7]=Y[7]`, б) `X[0]=Y[0]`, в) `X=Y`? Какое значение после этого будут иметь `X[7]` и `Y[7]`?
- Задание.** Что произойдет в результате исполнения операторов `int* X = new int[10]` и `int* X = new int(10)`?
- Задание.** Изучите программу одномерной игры `Life`, представленной в данном проекте. Найдите место в программе, где определяются правила игры. Измените правила по своему усмотрению так, чтобы игра не вырождалась. Например, установите правило, в соответствии с которым одномерный зверь четной длины бежит влево, в зверь нечетной длины бежит вправо. Продемонстрируйте столкновение.
- Задание.** Измените программу игры `Life` так, чтобы одномерный игровой массив был не булевским, а типа `int`. Создайте свои правила игры.

1.24 Операции с массивами.

- ✓ *Либерти С++21, глава 7.*
- ✓ *Дейтел С++, §4.6.*

1.24.1 Копирование массива.

1.24.2 Сортировка массива.

Практика 13. р3п13 Сортировка массива

- Задание.** Создайте и инициализируйте одномерные массивы различных арифметических типов (`int`, `double`, `char`) с помощью оператора `new` (не забудьте разрушить). Создайте функцию сортировки. Продемонстрируйте правильность работы. Напишите функцию сортировки, использующую функцию сравнения (например, `int compare(int x, int y){if(x<y)return -1; if(x==y)return 0; return 1;}` или `int compare(int x, int y){if(x<0&& y>0)return -1; if(x>0&& y<0)return 1; return 0;}`)
- Задание.** Создайте несколько массивов строк, инициализируйте их названиями ваших любимых музыкальных произведений, именами их авторов, год выпуска, название диска, продолжительность звучания и т.д. Сортируйте эти массивы одновременно по различным критериям, а) имя автора, б) год выпуска и т.д.

Лекция 9. Строки и указатели

- ✓ *Страуструп С++, §4.3, §5.2.2, §3.5.*
- ✓ *Дейтел С++, §5.12.*
- ✓ *Либерти С++21, глава 3, 12.*

1.24.3 Константные строки.

1.24.4 Массивы символов `char[]="...";` Символ конца строки.

1.24.5 Строка заданной длины.

1.24.6 Буфер.

1.24.7 Строка переменной длины.

1.24.8 Функции копирования, конкатенации, усечения.

1.24.9 Запрос длины строки.

1.24.10 Доступ к отдельным символам строки.**Практика 14. р3п14 Строки**

- Задание.** Создайте несколько константных строк. Создайте буфер. Примените операции копирования и конкатенации. Выведите в файл.
- Задание.** Создайте строку с помощью оператора `new` (не забудьте разрушить). Задайте значение с помощью функции генерирования случайных чисел. Проведите сортировку символов в строке.
- Задание.** Напишите функцию `swap(char* X, char* Y)` для обмена значений `X` и `Y`. Не забудьте, что при копировании строки большей длины в буфер меньшей длины потребуется переопределить буфер (разрушить старый и создать новый). Исследуйте возможности функции `swap(int** prx, int** pry)`, которая обменивает значения `*prx` и `*pry`.

Лекция 10. Двумерные массивы.**1.25 Двумерные массивы.**

- ✓ *Страуструп С++, §5.3.*
- ✓ *Дейтел С++, §4.9.*
- ✓ *Либерти С++21, глава 8, 12.*

1.25.1 Массивы указателей.**1.25.2 Понятие двумерного массива с фиксированной размерностью.****1.25.3 Доступ к элементам двумерного массива.****1.25.4 Вывод двумерного массива в файл.****Практика 15. р3п15 Двумерные массивы**

- Задание.** Создайте и инициализируйте двумерные массивы типов `int` и `double`. Создайте функцию сортировки. Создайте функцию вычисления наименьшего и наибольшего значений по двумерному массиву и отдельно по строкам и столбцам. Продемонстрируйте правильность работы.
- Задание.** Напишите функцию для печати в файл содержимого массива `char X[M][N]`. Нарисуйте график функции $y=f(x)$ с помощью двумерного массива `char[128][128]`.
- Задание.** Изучите программу `Life`. Переделайте игровой массив на тип `bool` (данное упражнение носит чисто учебный характер и позволит Вам усовершенствовать Ваше понимание типов С++). Запрограммируйте другие правила игры, используя информацию из Wikipedia. Измените программу так, чтобы можно было задавать матрицу правил размера `5x5` (вместо `3x3` в оригинале, данное упражнение носит чисто учебный характер и позволит Вам усовершенствовать Ваше понимание типов С++). Напишите программу, которая будет отслеживать путешествие стандартного life-организма по большому полю, размеры которого много больше, чем размеры экрана. Задайте случайное распределение начальных оккупированных позиций по игровому полю. Напишите код для анализа динамики игры (ширина и высота life-организма, его вес, т.е. число оккупированных позиций, время жизни, интенсивность жизненного цикла, которую определить догадайтесь как сами). Смоделируйте большое количество случайных начальных состояний и отберите из них наиболее жизнеспособные по критерию, который установите самостоятельно. Результаты фиксируйте в файлах начальных данных. После этого исследуйте визуально наиболее жизнеспособные состояния.
- Задание.** * (более сложное). Смоделируйте трехмерную игру.

1.26 Операторы `new` и `delete` для двумерных массивов.

- ✓ *Страуструп С++, §5.3.*
- ✓ *Дейтел С++, §6.10, 6.12–6.13.*
- ✓ *Либерти С++21, глава 8.*

1.26.1 Применение операторов new и delete для создания двумерного массива.**1.27 Передача двумерного массива в функцию.****Практика 16. Операторы new и delete-2**

Задание. Создайте и инициализируйте несколько двумерных массивов разных типов с помощью оператора new. Используйте delete для разрушения. Исследуйте применение указателей и переменных с индексом для доступа к элементам массивов. Выведите в файл. Исследуйте последствия попытки использования элементов массива до создания и после разрушения.

Лекция 11. Указатели на функции**1.28 Указатели на функции.**

- ✓ *Страуструп С++*, §7.7.
- ✓ *Дейтел С++*, §5.9, §5.11.
- ✓ *Либерти С++21*, глава 8.

1.28.1 Указатели на функции.**1.28.2 Массив указателей на функции.****Практика 17. Указатели на функцию**

Задание. Создайте несколько функций и используйте указатели для их вызова. Продемонстрируйте правильность работы.

Глава 4 Ввод и вывод

Лекция 12. Форматированный вывод в файл**1.29 Манипуляторы при выводе.**

- ✓ *Страуструп С++*, §21.1-21.4.
- ✓ *Дейтел С++*, §11.5, §11.6.
- ✓ *Либерти С++21*, глава 16.

1.29.1 Установка ширины поля вывода.**1.29.2 Установка точности.****1.29.3 Обзор манипуляторов.****Практика 18. r4n18 Форматированный вывод в файл**

Задание. Создайте таблицу (двумерный массив) и выведите в файл с нумерацией строк, столбцов. Прочитайте текстовый документ (отрывок из книги, предоставленный преподавателем). Составьте словарь. Найдите частоту повторения каждого слова в документе. Выдайте документ в файл в форматированном виде (выровненном влево, вправо, центрированном, выровненном по ширине).

Лекция 13. Форматированный ввод из файла**1.30 Манипуляторы при вводе.**

- ✓ *Страуструп С++*, §21.1-21.4.
- ✓ *Дейтел С++*, §11.5, §11.6.
- ✓ *Либерти С++21*, глава 16.

1.30.1 Ввод встроенных типов.**1.30.2 Функции распознавания образов.****Практика 19. Форматированный ввод из файла**

Задание. Создайте таблицу (двумерный массив) и выведите в файл с нумерацией строк, столбцов. Прочитайте и проверьте правильность. Прочитайте таблицу смешанных типов (текст-данные).

Лекция 14. Компоновка проекта**1.31 Создание проекта с несколькими файлами.**

- ✓ *Страуструп С++*, §9.1-9.6, §4.9.

- 1.31.1 Основные и заголовочные файлы.
- 1.31.2 Область видимости объекта.
- 1.31.3 Время жизни объекта.
- 1.31.4 Блок.
- 1.31.5 Обеспечение доступа к объекту из нескольких файлов.
- 1.31.6 Пять ключевых точек кода:
 - I) Объявление объекта
 - J) Определение объекта
 - K) Инициализация объекта
 - L) Использование объекта
 - M) Разрушение объекта

Практика 20. Компоновка проекта

Задание. Создайте проект с несколькими h-файлами и несколькими srr-файлами для сортировки одномерного массива так, чтобы функция сортировки была определена в одном файле, а массив подлежащий сортировке в другом. Исследуйте последствия определения двух одноименных объектов в разных файлах. Исследуйте применение блоков.

Глава 5 Объектно ориентированное программирование на С++.

Лекция 15. Понятие класса

1.32 Понятие класса.

- ✓ Страуструп С++, §10.2.
- ✓ Дейтел С++, §6.1-§6.5.
- ✓ Янг, часть I, глава 4,
- ✓ Либерти С++21, глава 6,

- 1.32.1 Понятие класса. Встроенные классы и классы пользователя.
- 1.32.2 Протокол класса.
- 1.32.3 Данные-члены (поля) и функции-члены.
- 1.32.4 Открытые и закрытые члены класса.
- 1.32.5 Дружественные функции.

1.33 Работа с объектами.

- ✓ Страуструп С++, §10.4.
- ✓ Дейтел С++, §6.1-§6.5.
- ✓ Янг, часть I, глава 4,
- ✓ Либерти С++21, глава 6,

- 1.33.1 Объявление и описание объекта данного класса.
- 1.33.2 Доступ к данным и функциям для объектов и указателей на объекты.
- 1.33.3 Соглашение об именах классов, объектов и указателей.

Практика 21. р5п21-Создание простого класса

Задание. Создайте протокол своего класса (например, СМуCar , мой любимый автомобиль) с несколькими полями данных (например, фирма, название, год выпуска, пробег и т.д.). Создайте объект, указатель на объект. Инициализируйте указатель. Выдайте в файл.

1.34 Способы доступа к полям данных.

- ✓ Страуструп С++, §10.4.
- ✓ Дейтел С++, §14.4-14.6, §6.14,
- ✓ Либерти С++21, глава 6.

- 1.34.1 Способы доступа к членам класса. Интерфейсные функции.
- 1.34.2 Интерфейсные функции типа get(...).

- 1.34.3 Интерфейсные функции типа `set(...)`.
- 1.34.4 Методы валидации параметров.
- 1.34.5 Выдача протокола работы программы в файл.

1.35 Обеспечение безопасности

✓ *Страуструн С++, §10.2.*

- 1.35.1 Константные объекты и константные функции-члены.
- 1.35.2 Статические данные-члены и статические функции-члены.

Практика 22. Функции доступа

Задание. Создайте функции доступа. Измените значение полей данных. Выдайте в файл с помощью функций доступа. Создайте `friend` функцию.

Лекция 16. Конструкторы и деструкторы

1.36 Конструкторы и деструкторы.

✓ *Страуструн С++, §10.2, 10.4.*

✓ *Янг, часть I, глава 4,*

✓ *Дейтел С++, §6.10, 6.11, 6.14,*

✓ *Либерти С++21, глава 6.*

- 1.36.1 Конструирование объекта.
- 1.36.2 Принципы работы с указателями на объект класса пользователя.
- 1.36.3 Инициализирующий конструктор (конструктор с параметрами).
- 1.36.4 Конструктор с аргументами по умолчанию. Перегрузка конструктора.

Практика 23. Конструктор и деструктор

Задание. Создайте конструктор и деструктор `SMuCar`. Добавьте инициализирующий конструктор (с параметрами).

1.37 Конструирование объектов, включающих массивы.

✓ *Страуструн С++, §10.4.6.*

- 1.37.1 Конструирование и разрушение объектов, включающих поля данных – массивы с постоянной размерностью.
- 1.37.2 Конструирование и разрушение объектов, включающих поля данных – массивы с переменной размерностью.

Практика 24. Конструктор и деструктор

Задание. Создайте конструктор и деструктор класса, имеющего поле данных – массив с постоянной размерностью.

Лекция 17. Конструктор копирования

1.38 Копирование объектов.

✓ *Страуструн С++, §10.2, 10.4.*

✓ *Дейтел С++, §6.01-6.11, §8.1-8.3,*

✓ *Либерти С++21, глава 6, 10.*

- 1.38.1 Применение указателя `this`.
- 1.38.2 Перегрузка оператора копирования `=`.
- 1.38.3 Операция `swap(..., ...)`.
- 1.38.4 Конструктор копирования.

Практика 25. Конструктор копирования

Задание. Создайте конструктор копирования `SMuCar`. Добавьте в протокол и реализацию класса `SMuTime` перегруженный оператор `"="`. В класс `SMuWork` добавьте метод обмена значений двух полей. В класс `SMuTime` добавьте метод обмена значений двух объектов класса `SMuTime`, используя промежуточный объект того же класса.

Лекция 18. Массив объектов и массив-поле данных

1.39 Массивы объектов класса.

✓ *Страуструн С++, §10.4.*

1.39.1 Массив фиксированного размера объектов класса.

1.39.2 Доступ к элементам массива.

1.39.3 Массив переменного размера объектов класса.

1.40 Массив – поле данных класса пользователя.

✓ *Страуструп С++, §10.4.*

1.40.1 Массив фиксированного размера – поле данных класса.

1.40.2 Массив переменного размера – поле данных класса.

Практика 26. Массив объектов и массив-поле данных

Задание. 1. Создайте массив объектов класса СМуTime. Инициализируйте конструктором по умолчанию. Задайте значения полей (числовых и текстовых) явным образом. Создайте интерфейс для просмотра значений каждого объекта. Не забудьте разрушить массив перед завершением программы.

Задание. 2. Измените протокол и реализацию класса СМуTime так, чтобы поле данных, являющееся одномерным массивом, создавалось динамически, причем размер массива равен заданному постоянному значению. Добавьте проверку превышения размера массива с соответствующей диагностикой.

Лекция 19. Перегрузка операций

1.41 Перегрузка унарных операций.

✓ *Страуструп С++, §11.2.*

✓ *Либерти С++21, глава 10.*

✓ *Дейтел С++, §8.4-§8.8.*

1.41.1 Перегрузка операции сравнения.

1.41.2 Перегрузка операции сравнения, пример: сортировка.

1.41.3 Перегрузка унарных операций.

Практика 27. Перегрузка унарных операций

Задание. Добавьте в протокол и реализацию класса СМуTime перегруженные операторы сравнения “>”, и “==”, и “!=”. Добавьте в протокол и реализацию класса СМуWork метод сортировки, использующий перегруженный оператор сравнения. Проверьте правильность работы сортировки. Добавьте в протокол и реализацию класса СМуTime перегруженные операторы «!» и «++». Проверьте правильность работы. Добавьте в протокол и реализацию класса СМуTime перегруженный оператор сравнения “<”. Проверьте работу оператора для нескольких объектов класса.

1.42 Перегрузка бинарных операций

✓ *Страуструп С++, §11.2.*

✓ *Дейтел С++, §8.4-§8.8.*

1.42.1 Перегрузка бинарных операций для двух операндов одного типа.

1.42.2 Перегрузка бинарных операций для операндов смешанного типа.

Практика 28. Перегрузка бинарных операций

Задание. 1. Изучите перегруженные бинарные операторы с операндами, если оба являются объектами класса пользователя. Изучите перегруженные бинарные операторы с операндами, один из которых является объектом класса пользователя, а другой – объектом встроенного класса. Изучите перегруженные бинарные операторы с операндами, первый из которых является объектом класса пользователя, а второй – объектом другого класса пользователя. Добавьте в протокол и реализацию класса СМуTime перегруженные friend операторы “+”, “-”, “*”. Проверьте правильность работы.

Задание. 2. Добавьте в протокол и реализацию класса СМуTime перегруженные friend операторы “+”, “-”, “*” для операндов смешанного типа, например СМуTime и int. Проверьте правильность работы.

1.43 Преобразование типов.**1.43.1 Преобразование типов для объектов встроенных классов.****1.43.2 Перегрузка операций преобразования типов.****Практика 29. Перегрузка операции преобразования типа**✓ *Страуструн С++, §11.4.*✓ *Дейтел С++, §8.9.*

Задание. Добавьте в протокол и реализацию класса `СMyTime` перегруженные операции преобразования типа `int(СMyTime)` и `СMyTime(double)`. Проверьте правильность работы.

1.44 Перегрузка операции помещения в поток и взятия из потока.✓ *Дейтел С++, §11.3,***1.44.1 Перегрузка операции помещения в поток и взятия из потока.**

Задание: Добавьте в протокол и реализацию класса `СMyTime` перегруженные операторы “<<”, “>>”. Проверьте правильность работы.

Литература: Дейтел С++, §8.5.

Лекция 20. Примеры классов**1.45 Примеры классов, обладающих большой степенью функциональности.**✓ *Дейтел С++, §8.8, 8.10.***1.45.1 Пример: класс `array`.****1.45.2 Пример: класс `complex`.****1.45.3 Пример: класс `string` для обработки строк.****Практика 30. Исследование класса `array`**

Задание. Создайте класс `array` для хранения одномерного массива, оснащенный конструктором, деструктором, арифметическими операторами, операторами сравнения, копирования, сортировки, выдачи в файл. Напишите операторы вычисления минимума, максимума, среднего значения и дисперсии. Проверьте правильность работы.