

**Лабораторная работа № 1.
Знакомство с языком C++**

Материалы к защите

ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ

1. Область применения языка C++. Принципы проектирования языка и влияние на них областей его применения. Стандартизация C++. Степени свободы поведения программы.
2. Средства разработки: средства компиляции (компилятор, компоновщик), среда разработки (редактор, отладчик), интегрированная среда разработки. Краткое описание и характеристика предпочитаемых (лично) средств разработки для C++.
3. Встроенные (фундаментальные) типы данных C++. Неизменяемость (константность). Приведение типов. Особенности чисел с плавающей запятой.
4. Область видимости и перекрытие имен. Глобальные переменные. Массивы (одномерные и многомерные): объявление, использование. Обращения по индексу вне границ измерения массива.
5. Основные операторы C++: циклы, условия, переходы (**break**, **continue**, **return**). Оператор **goto** и проблемы его применения. Условное выражение (оператор **?:**).
6. Выражения в языке C++: арифметические, логические и побитовые операции, оператор присваивания, условное выражение (оператор **?:**).
7. Функции: объявление, определение, вызов. Перегрузка функций. Рекурсия. Стек. Статические переменные в функциях.
8. Раздельная компиляция: единицы трансляции и заголовочные файлы. Препроцессор. Механизм включения файлов. «Стражи включения» (**include guards**).

ТЕОРЕТИЧЕСКИЕ ЗАДАЧИ

1. Какие достоинства, а также недостатки или неудобства, на ваш взгляд, имеет изученная часть библиотеки ввода-вывода в стиле C (`cstdio`)? Укажите способы борьбы с названными недостатками при написании программ.
2. Выясните, как представляются числа с плавающей точкой, согласно IEEE 754 (только общий вид). Объясните, что такое потеря точности, машинное эпсилон (`machine epsilon`), не-число (NaN), и как эти понятия проявляются в программах.
3. Язык C++ является статически- и строго-типизированным. Что это означает? Какие преимущества обеспечивает данный подход? Какие неудобства создает? Какие известные вам языковые средства C++ связаны со строгой типизацией?
4. Выясните поведение и ограничения побитовых логических операторов и операторов битового сдвига при работе со знаковыми и отрицательными операндами. Какими средствами вы воспользуетесь для поиска ответа и почему?
5. Имеются два цикла, один вложен в другой. По условию, находящемуся в теле внутреннего цикла, требуется немедленно выйти из обоих циклов сразу. Предложите способы это сделать, опишите сравнительные преимущества и недостатки решений. Весь код в вашем распоряжении (при условии, что логика работы сохраняется).
6. Чем условное выражение (тернарный оператор) отличается от оператора `if`? Можно ли написать функцию, эквивалентную тернарному оператору (конкретному и произвольному)? Если да, то как, если нет, то почему?
7. Предлагается избежать переполнения стека рекурсивной функцией `f()` следующим образом: пусть `f()` вызывает `g()`, а `g()` вызывает `f()` — таким образом, `f()` не вызывает себя напрямую. Обоснованно поддержите или оспорьте эту идею.
8. Безопасно ли передавать в функцию с переменным числом аргументов, например, `printf()`: а) лишние параметры, б) недостаточно параметров, — и к чему это может привести? Ответ обоснуйте.
9. Аргументированно оцените рекомендации по написанию кода:
 - а) Фигурные скобки следует опускать всюду, где это возможно.
 - б) Код следует стремиться разбивать на функции так, чтобы размер кода каждой не превышал размера экрана (приблизительно).
 - в) Возвращаемое **return** значение нужно заключать в круглые скобки.
 - г) Следует применять цикл **for** только в виде **for (int i = a; i < b; i++)**, где *i* — переменная-счетчик, *a* и *b* — границы цикла.
 - д) Все переменные и константы следует объявлять в начале функции.

ПРАКТИЧЕСКИЕ ЗАДАЧИ

1. Экспериментально определить машинное эпсилон для типов **float** и **double**; напечатать двоичное представление этих значений.
2. Повернуть на 90° по часовой стрелке квадратную матрицу целых чисел. Объем используемой для поворота памяти не должен зависеть от размера матрицы.
3. Вычислить значение квадратичной формы $f(\vec{x}) = \vec{x}^T \cdot A \cdot \vec{x}$ для заданной матрицы A в точке \vec{x} . Поощряется экономия используемой оперативной памяти.
4. Решить СЛАУ с трехдиагональной матрицей методом прогонки.
5. Проинтегрировать методом Эйлера уравнение $y'(t) = \frac{1}{T}[k - y(t)]$, $y(0) = 0$, $k > 0$, $T > 0$ на отрезке $[0; t_1]$ с шагом Δt , где все параметры задаются пользователем.
6. Найти в массиве четной длины все пары ближайших друг к другу элементов (как при кластеризации). Например, в массиве $\{1, 3, 4, -2\}$ это пары $(3, 4)$ и $(-2, 1)$.
7. По заданной одномерной выборке вычислить коэффициенты линейной регрессии.
8. Вычислить оценку медианы по неупорядоченной выборке действительных чисел.
9. Вычислить оценку ковариации двух выборок вещественных чисел.
10. При помощи функции `clock()` заголовочного файла `<time.h>` определить, что быстрее: умножение или деление переменных, и на сколько (относительно).