ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ C++

2 лекция

Дисциплина «Программирование микроконтроллеров на языке C++»

Технологии программирования

Технологии программирования - это совокупность методов, инструментов и организационных подходов, обеспечивающих эффективную разработку, отладку и сопровождение программного обеспечения. Они направлены на повышение качества программ, сокращение сроков их создания и упрощение дальнейшей поддержки.

Ранние подходы (линейное программирование)

Программа на алгоритмическом языке представляет собой линейную последовательность операторов присваивания, условных операторов и циклов. Такие программы часто содержали запутанные конструкции, в которых было трудно проследить логику работы. Подобный способ разработки позволял решать только сравнительно простые задачи (объемом до ≈ 100 строк кода). Это привело к необходимости более строгой организации программного процесса и развитию **структурного программирования**.

Структурное программирования

Технология структурного программирования основывается на формировании программы из фиксированного множества подпрограмм с использованием базовых конструкций:

- линейной,
- ветвления,
- цикла,

Преимущества:

- ясность структуры программы,
- простота тестирования и отладки,
- легкость модификации.

Основные цели: повышение читаемости, надежности и эффективности программ.

Принципы структурного программирования

- **Декомпозиция** – разбиение сложной задачи на более простые подзадачи.
- **Нисходящее программирование (top-down design)** или программирования **сверху вниз.** Задача делится на подзадачи первого уровня, далее каждая подзадача уточняется до уровня, где алгоритм можно реализовать в виде кода. Итоговая программа состоит из подпрограмм, каждая из которых решает свою часть задачи.
- **Восходящее программирование (bottom-up design)** или программирования **снизу вверх.** Программа строится из уже готовых процедур и функций, под которые подстраивается решение задачи.

Модульное программирования

Следующим шагом (70-е годы XX века) стал **модульный подход**. Программы начали строиться из отдельных самостоятельных частей – **модулей**, которые могли включаться в разные проекты.

Характерные особенности:

- экспорта и импорта имен,
- инкапсуляция данных и процедур,
- повторное использование кода..

Модульное программирование стало логическим развитием структурного подхода и позволило создавать более крупные и удобные в сопровождении программные комплексы.

Объектно-ориентированное программирование

В результате многолетних исследований был разработан и опробован так называемый **объектно-ориентированный подход** *(ООП)* (в 1967 году создан первый язык, основанный на данном подходе – Simula-67, в 1983 году - язык C++). Главная причина возникновения ООП связана с поиском простых путей для создания сложных программ. ООП наследует лучшие черты структурного программирования и комбинирует их с некоторыми новыми подходами.

Ключевые принципы ООП:

- **инкапсуляция** – объединение данных и методов их обработки внутри одного объекта,
- **наследование** – возможность построения иерархии классов и повторного использования кода,
- **полиморфизм** – способность объектов вести себя по-разному при использовании общих интерфейсов.

Объектно-ориентированное программирование

- ООП унаследовало лучшие идеи структурного программирования и дополнило их новыми возможностями, позволяющими создавать сложные программные системы более удобным и наглядным образом.
- Главная причина появления ООП заключалась в поиске простых и эффективных способов разработки больших и сложных программ. Благодаря этому подходу стало возможным строить программное обеспечение как систему взаимодействующих объектов, что существенно упростило разработку, тестирование и поддержку программных продуктов.
- В итоге развитие технологий программирования можно представить как последовательность:
  линейное → структурное → модульное → объектно-ориентированное программирование.

Этапы разработки программы на языке C++

1 Редактирование исходного кода.

Первое, что нужно сделать для создания программы, - записать её текст в файл. Для этого используется текстовый редактор, позволяющий писать, сохранять и исправлять текстовые данные. Текстовые процессоры (например, *Microsoft Word*) для этих целей не подходят, так как добавляют лишнюю информацию о форматировании.

Файлы с текстами программ, которые создаются с помощью редактора текста, называются ***исходными кодами***, и они имеют названия с расширением *.CPP, например programma.cpp. В именах файлов не рекомендуется использовать русские буквы.

Этапы разработки программы на языке C++

2 Компиляция.

Для превращения исходного кода в выполняемую компьютером программу используется компилятор. После компиляции создаётся объектный файл с расширением **.OBJ**. Он содержит промежуточный код, но ещё не является исполняемой программой.

Этапы разработки программы на языке C++

3 Редактирование связей.
Для того чтобы превратить объектный файл в программу, нужно будете запустить на выполнение еще одну специальную программу - ***редактор связей*** (linker).

Данная программа просматривает в указанных объектных файлах списки нужных функций, находит их и присоединяет к общей программе. Во время своей работы редактор связей ищет функции в различных библиотеках. ***Библиотека*** является набором объектных файлов, которые поставляются вместе с компилятором, либо приобретаются отдельно, либо создаются самими программистами. Все компиляторы C++ поставляются со ***стандартной библиотекой*** полезных функций и классов, которые можно использовать при разработке программ.

В результате работы редактора связей получается выполняемая программа, которая записывается в файл с расширением *.EXE.

Этапы разработки программы на языке C++

4. Отладка программы.

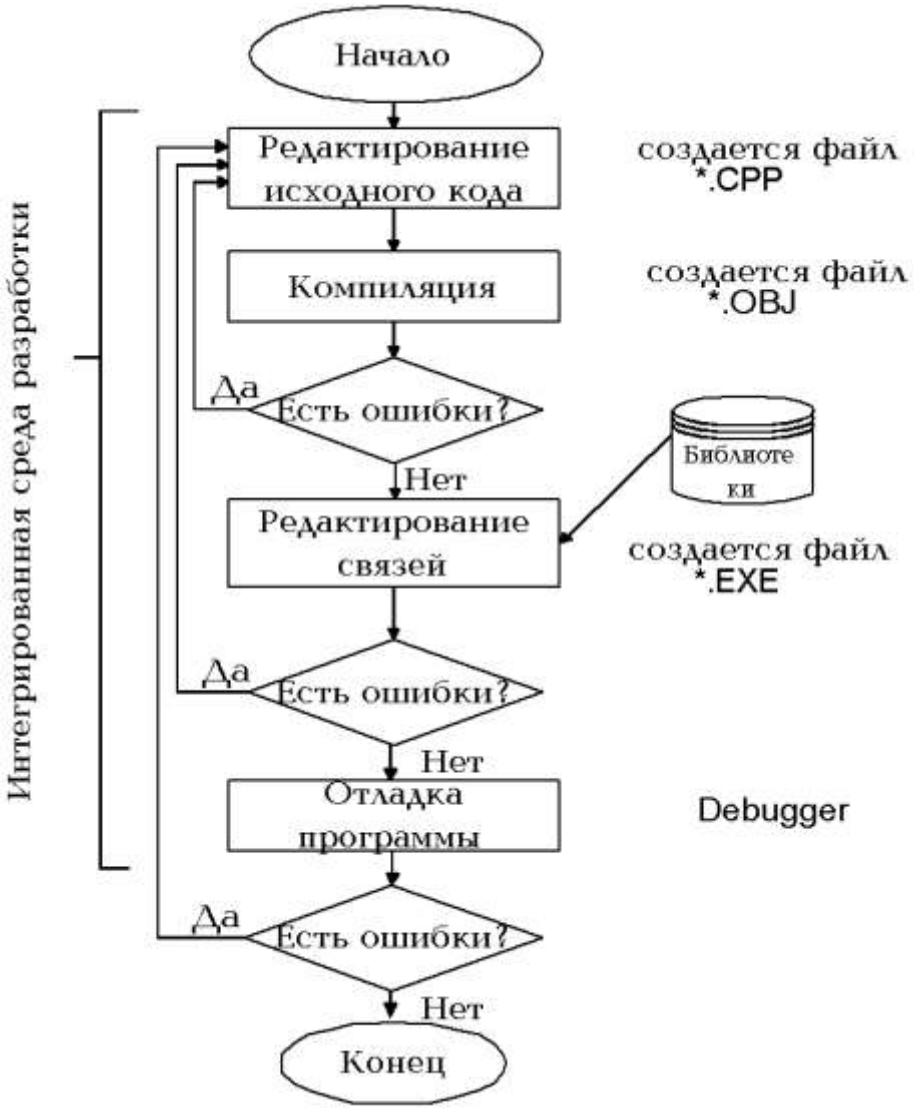
Для проверки работы программы используются специальные программы ***отладчики (debugger).*** Он позволяет отслеживать выполнение программы и контролировать значения переменных. .

При обнаружении ошибок программист возвращается к редактированию исходного кода, выполняет перекомпиляцию и повторного редактирования связей, а затем нового запуска программы.

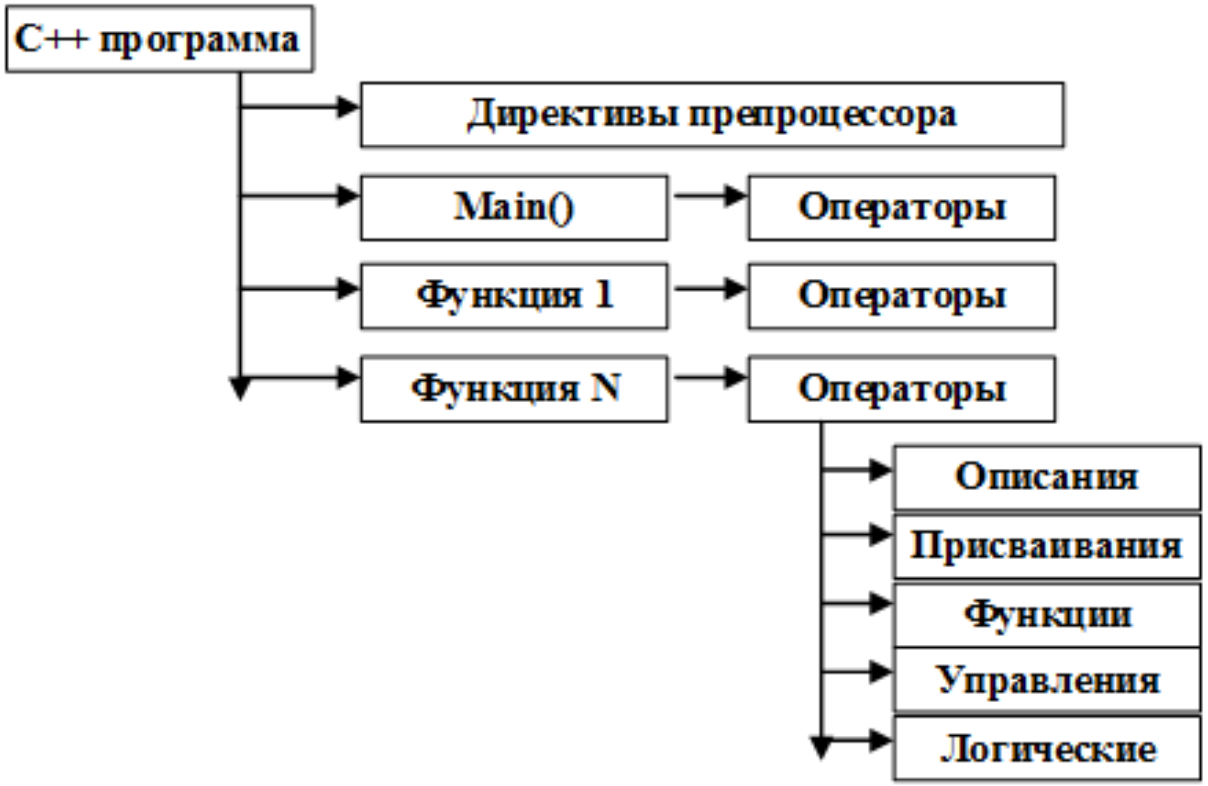
5. Завершение работы.

Полный цикл разработки (редактирование → компиляция → линковка → отладка) повторяется до тех пор, пока программа не будет полностью отлажена.

Этапы разработки программы на языке C++



Структура программы на С++



Структура программы

Простейшая программа на языке С++.

При выполнении эта программа предлагает пользователю ввести его возраст в годах, а затем рассчитывает соответствующее количество дней и выводит полученный результат на экран.

```
#include<iostream>
using namespace std;
int main()
{
int years, days;
cout << "How old are you in years? ";
cin >> years; //  вводим количество лет
days = years * 365; //  вычисляем количество дней
cout<<"Your age in days is about: " << days<<endl; //  выводим на экран результат
system ("pause");
return 0;
}
```

Структура программы

Данная программа включает такие общие для всех программ на языке C++ части, как

- директиву препроцессора;
- директиву задания пространства имен using;
- определение функции main
- ввод данных и вывод результатов из программы
- комментарии

- Каждый раз, когда компилятор запускается на выполнение, начинает работать специальная часть компилятора - *препроцессор*. Препроцессор читает исходный текст, ищет строки, которые начинаются с символа решетки (#), и обрабатывает эти строки, прежде чем начнет работать компилятор.
- Символ # указывает, что прежде чем транслировать программу, необходимо включить в нее функции из файла <iostream> - это препроцессорная обработка, которая осуществляет обработку текста программы перед началом компиляции.
- Угловые скобки вокруг имени файла дают команду препроцессору искать этот файл в специальном подкаталоге INCLUDE, где обычно такие файлы находятся. Этот подкаталог содержит все ***заголовочные*** файлы, которые используются компилятором. В заголовочных файлах содержатся объявления стандартных функций, классов и констант, которые можно использовать в программе. В данном файле объявлены потоки ввода-вывода cin и cout, с помощью которых выполняется ввод исходных данных с клавиатуры и вывод сообщений на дисплей.

- ***Препроцессор*** запускается на выполнение до начала работы компилятора каждый раз, когда компилятор вызывается на выполнение. Препроцессор преобразует все строки, которые начинаются с символа решетка (#) в специальные команды, преобразуя файл с исходным кодом к виду, который требуется для его последующей обработки компилятором.
- Полные имена объектов, которые хранятся в библиотеках, включают название ***пространства имен***, где они определены. Пространство имен это просто название группы имен. Например, объект cout имеет полное имя std::cout. Здесь, std - название пространства имен, в котором описаны все элементы стандартной библиотеки языка C++, а cout имя объекта. Символы “::“ называются операцией уточнения области видимости. Если использовать только имя объекта, то компилятор, будет выдавать сообщение, что данный объект не объявлен. Поэтому, мы должны либо использовать полные имена библиотечных объектов в программе (что может быть утомительным), либо заранее сказать компилятору, чтобы он искал все стандартные объекты в пространстве имен std с помощью директивы: using namespace std;

- Строка main() содержит название функции, с которой всегда начинается программа. Когда программа начинает работать, эта функция вызывается автоматически операционной системой. Пустые скобки обозначают, что эта функция без параметров. В самом общем виде, *функция является блоком инструкций программы, который решает какую-то конкретную задачу*.
- Для функции main(), так же как и для других функций, должно быть указано, результат какого типа она возвращает. Типом возвращаемой величины для функции main() в SIMPLE.CPP является int, означающее, что результатом работы функции является целое число.

- Инструкции, которые входят в функцию, заключаются в фигурные скобки. Все, что находится между открывающей и закрывающей скобками называется телом функции.
- Имена cin и cout используется в языке C++ для описания ввода данных с клавиатуры и вывода текста и значений на экран. Объекты cin и cout называются ***потоками ввода и вывода***. Если требуется вывести на экран текст (строку символов), то его нужно обязательно заключать в двойные апострофы.
- Обычно функция main() возвращает значение int. Это значение передается операционной системе, когда программа завершает работу. Некоторые программисты сообщают об ошибке, возвращая значение 1. В данном случае, функция main() будет всегда возвращать 0.
- После символов // стоят комментарии (для одной строки).

Вывод

- Структура программы на языке C++ строго регламентирована: она начинается с директив препроцессора, включает функцию main() как точку входа и использует дополнительные элементы (библиотеки, заголовочные файлы, функции, комментарии). Строгая структура программы на языке C++ способствует её ясности, удобству сопровождения и расширения. Знание этих основ позволяет создавать программы различной сложности – от простых обучающих примеров до крупных программных систем.
- Таким образом, процесс разработки программы на языке C++ можно рассматривать как сочетание теоретических принципов программирования и практических инструментов, обеспечивающих эффективное решение современных задач автоматизации и управления.