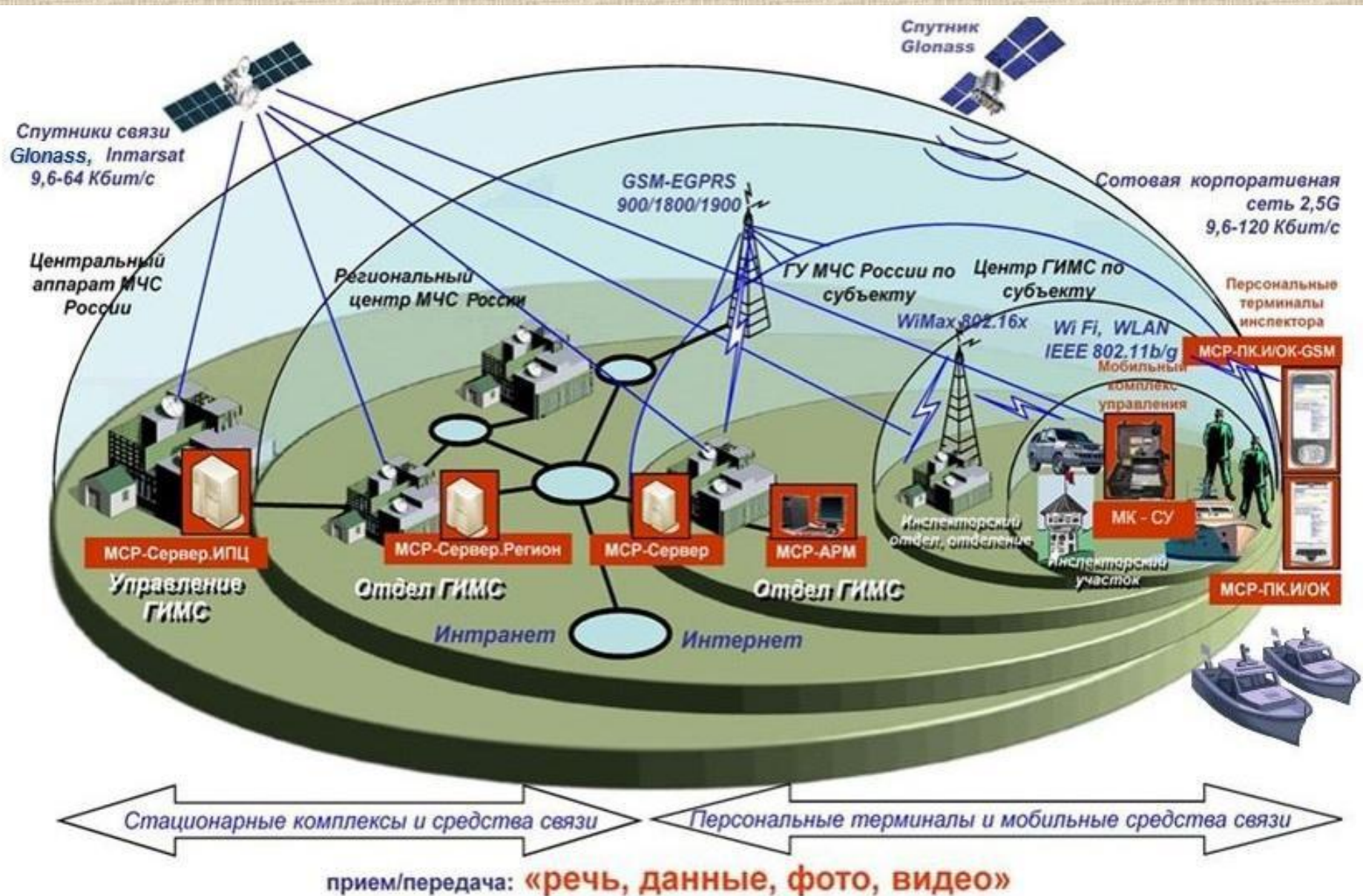


МДК 01.02. Прикладное программирование

Объектно-ориентированное программирование



Основная литература

1. Роберт Лафоре. Объектно-ориентированное программирование в С++. – 2011 г.
2. Стенли Липпман и др. Язык программирования С++. Вводный курс. – 2007 г.
3. Бьерн Страуструп. Программирование. Принципы и практика использования С++. – 2011 г.
4. Брюс Эккель. Философия С++. Практическое программирование. – 2004 г.
5. Харви М. Дейтел. Как программировать на С++. – 2008 г.

Дополнительная литература

1. Бьерн Страуструп. Язык программирования C++. – 2011 г.
2. Эндрю Кениг, Барбара Му. Эффективное программирование на C++. – 2002 г.
3. Скотт Мейерс. Эффективное использование C++. – 2006 г.
4. Герб Саттер. Решение сложных задач на C++. – 2008 г.
5. Герб Саттер. Новые сложные задачи на C++. – 2005 г.

Программа, которая выводит на экран дисплея фразу
Hello, World!

```
//HELLO.CPP – имя файла с программой  
#include <iostream>  
void main ()  
{ cout << "\nHello, World!\n";  
}
```

Результат выполнения программы:
Hello, World!

1. `//HELLO.CPP` – имя файла с программой
2. `#include <iostream>`
3. `void main ()`
4. `{ cout << "\nHello, World!\n";`
5. `}`

1. В первой строке текста программы – однострочный комментарий, начинающийся парой символов `“//”` и заканчивающийся неизображаемым символом «конец строки». Между этими разделителями может быть помещен любой текст. В этом примере в комментарии указано имя файла `hello.cpp`, в котором хранится исходный текст программы.

1. `//HELLO.CPP` – имя файла с программой
2. `#include <iostream>`
3. `void main ()`
4. `{ cout << "\nHello, World!\n";`
5. `}`

2. Во второй строке помещена команда (директива) препроцессора, обеспечивающая включение (`include`) в программу средств связи со стандартными потоками ввода и вывода данных. Указанные средства находятся в файле с именем `iostream.h` (мнемоника: “i” (input) – ввод, “o” (output) – вывод, `stream` – поток, “h” (head) – заголовок). Стандартным потоком вывода по умолчанию считается вывод на экран дисплея. Стандартный поток ввода обеспечивает чтение данных от клавиатуры.

Примечание: использование суффикса «.h» при подключении стандартных файлов («`#include <iostream.h>`») – это устаревший

1. `//HELLO.CPP` – имя файла с программой
2. `#include <iostream.h>`
3. `void main ()`
4. `{ cout << "\nHello, World!\n";`
5. `}`

3. Третья строка является заголовком функции с именем `main`. Любая программа на языке C++ должна включать одну и только одну функцию с этим именем. Именно с нее начинается выполнение программы. Перед именем `main` помещено служебное слово `void` – спецификатор типа, указывающий, что функция `main` в данной программе не возвращает никакого значения. Круглые скобки после `main` требуются в соответствии с форматом (синтаксисом) заголовка любой функции. В них помещается список параметров. В данном примере параметры не нужны, и список пуст.

1. `//HELLO.CPP` – имя файла с программой
2. `#include <iostream.h>`
3. `void main ()`
4. `{ cout << "\nHello, World!\n";`
5. `}`

4. Тело любой функции – это заключенная в фигурные скобки последовательность описаний, определений и операторов. Каждое описание, определение или оператор заканчивается символом «точка с запятой». В теле функции `main` явных описаний и определений нет, а есть только один оператор `cout << "\nHello, World!\n";` Имя `cout` в соответствии с информацией, содержащейся в файле `iostream.h`, является именем объекта, который обеспечивает вывод информации на экран дисплея (в стандартный поток вывода).

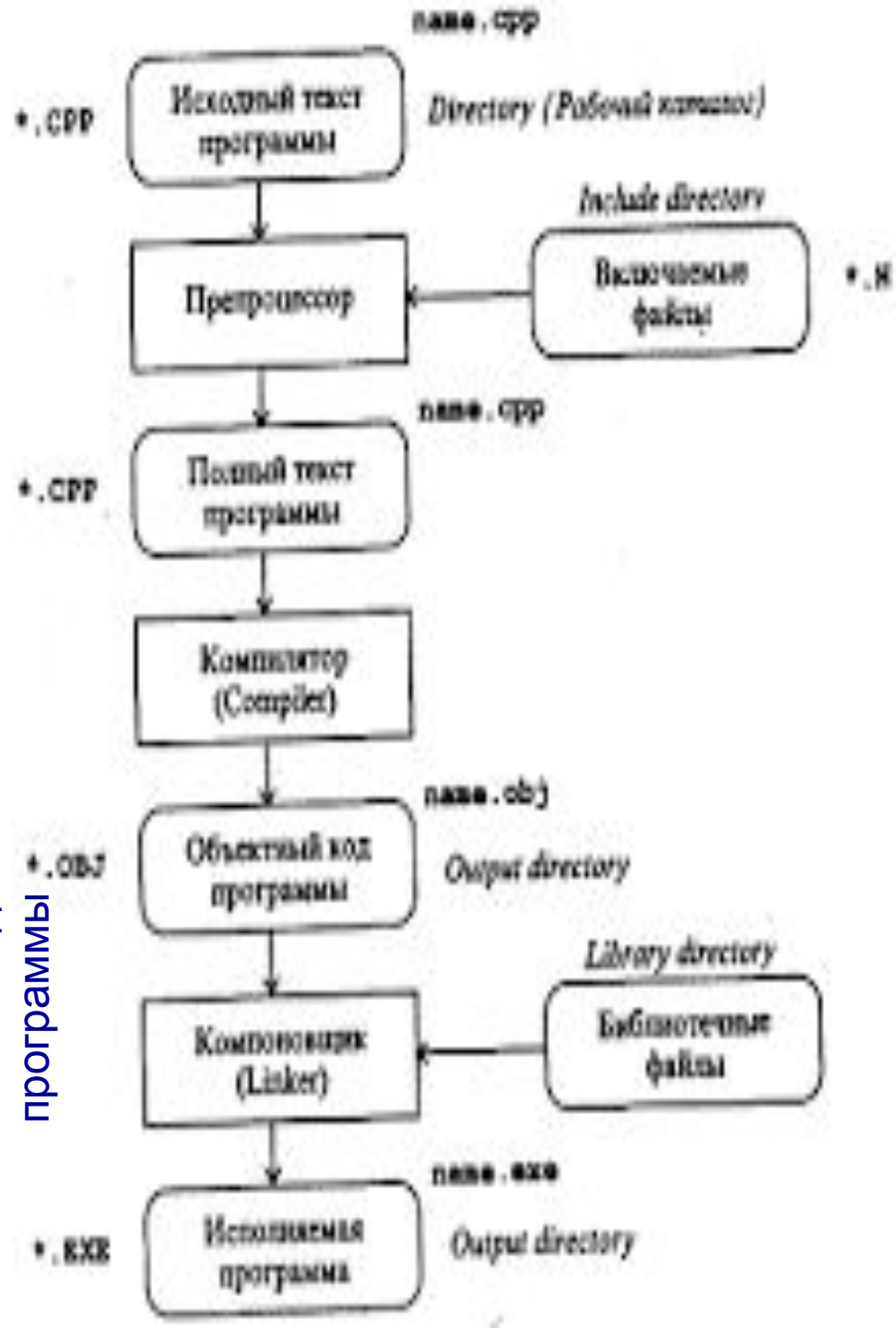
1. `//HELLO.CPP` – имя файла с программой
2. `#include <iostream.h>`
3. `void main ()`
4. `{ cout << "\nHello, World!\n";`
5. `}`

4. Информация для вывода передается объекту `cout` с помощью операции `<<` («поместить в»). То, что нужно вывести, помещается справа от знака операции `<<`. В данном случае это строка (строковая константа) `"\nHello, World!\n"`. Строка в языке C++ определена как заключенная в кавычки почти любая последовательность символов. Среди них могут встречаться обозначения неизображаемых на экране дисплея управляющих символов. Например, `'\n'` (newline) – обозначение управляющего символа перехода к началу следующей строки экрана. Т.о., программа выведет на новой строке фразу `Hello, World!` и переведет курсор в начало следующей строки экрана.

Одна из принципиальных особенностей языка C++, называемая перегрузкой или **расширением действия стандартных операций**, заключается в том, что лексема << означает операцию вставки («поместить в») только в том случае, если слева от нее находится имя объекта **cout**. В противном случае пара символов << означает бинарную операцию сдвига влево.

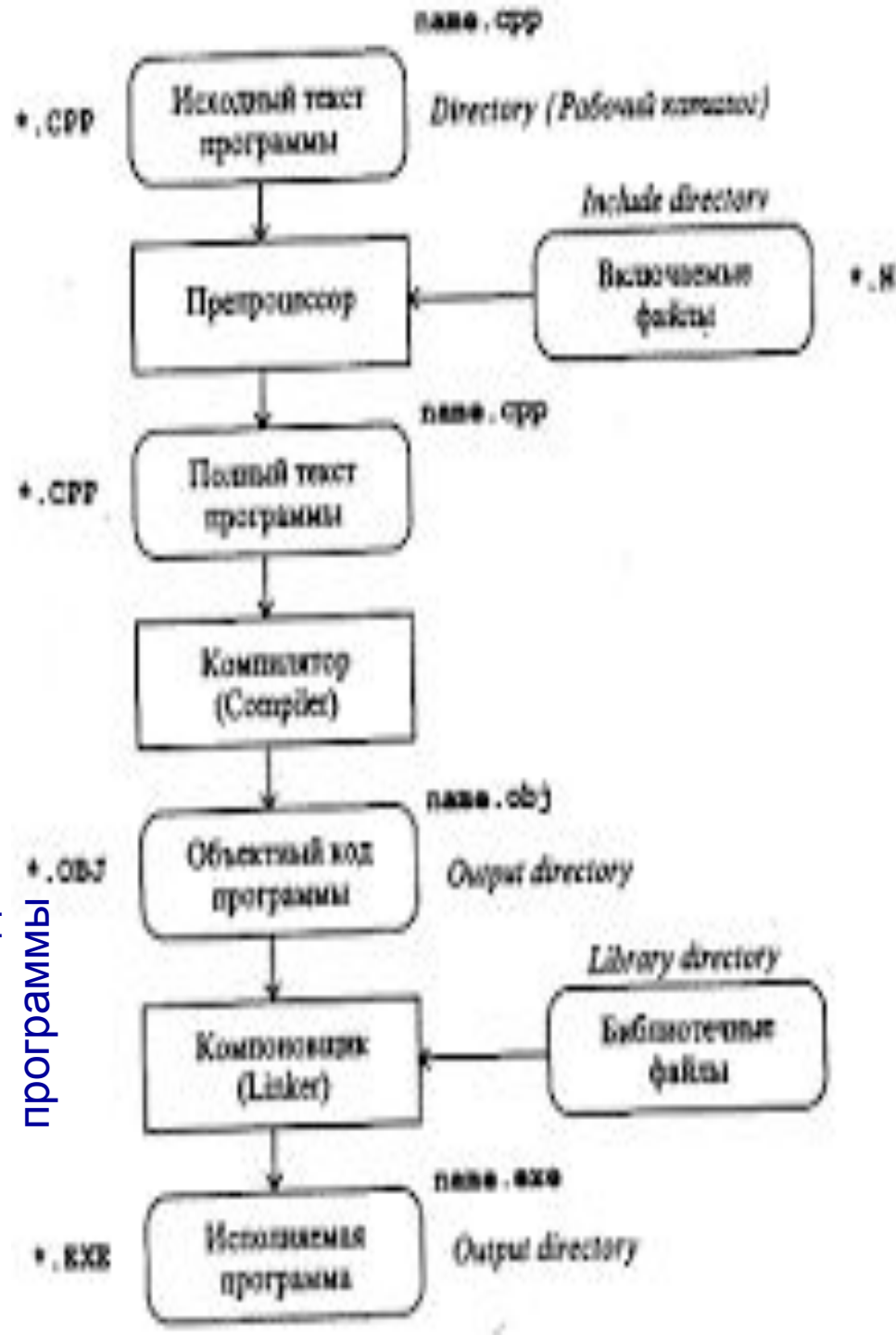
До выполнения программы необходимо подготовить ее текст в файле с расширением **.cpp**; передать этот файл на компиляцию и устранить синтаксические ошибки, выявленные компилятором; безошибочно откомпилировать (получится объектный файл с расширением **.obj**); дополнить объектный файл нужными библиотечными функциями (компоновка) и получить исполняемый

Схема подготовки исполняемой программы



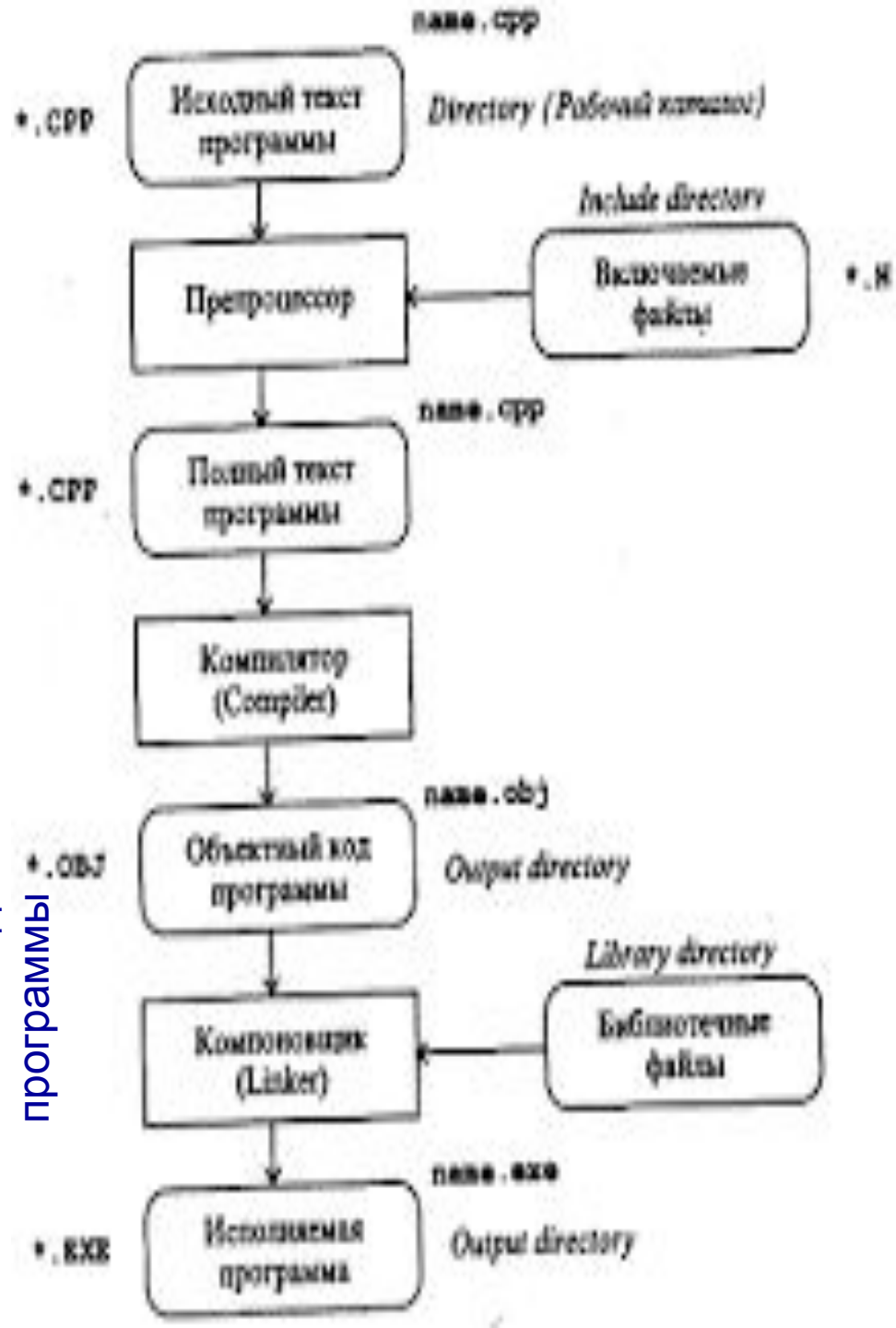
Перед шагом компиляции показан шаг препроцессорной обработки текста программы. В примере с [Hello, World!](#) Препроцессор обрабатывает директиву `#include <iostream.h>` и подключает к исходному тексту программы средства для обмена с дисплеем (для поддержки операции `<<`).

Схема подготовки исполняемой программы



Если исходный текст программы подготовлен в файле `hello.cpp`, то препроцессор сформирует полный текст программы, компилятор создаст объектный файл `hello.obj`, выбрав (по умолчанию) для него указанное имя, а компоновщик (редактор связей, Linker) дополнит программу библиотечными функциями, например, для работы с объектом `cout`, и построит модуль `hello.exe`. Особенности выполнения перечисленных действий зависят от конкретного компилятора языка C++ и от ОС.

Схема подготовки исполняемой программы



Еще второй вариант:

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    cout << "Hello, World!";

    cout << "\n";
    system ("Pause");
    cin.get();
    return EXIT_SUCCESS;
}
```


И еще третий вариант:

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello, World!" << endl;
7     system ("Pause");
8     return 0;
9 }
```

1: поддержка потоков (streams) ввода/вывода (Input/Output)

2: использовать имена из стандартной библиотеки C++

3: пустые строки и комментарии // игнорируются; они служат для повышения читабельности и документирования

Третий вариант:

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello, World!" << endl;
7     system ("Pause");
8     return 0;
9 }
```

4: main() — любая программа на C++ должна включать функцию с этим именем; ключевое слово int означает, что программа возвратит операционной системе целое число
5, 9: тело любой функции заключается в фигурные скобки

Третий вариант:

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello, World!" << endl;
7     system("Pause");
8     return 0;
9 }
```

6: cout - это имя объекта, который обеспечивает вывод информации на экран дисплея

6: то, что нужно вывести, разделяется символами <<

6: печать endl приводит к переводу курсора на новую строку

7: конструкция system("pause") ожидает от пользователя нажатия любой кнопки и не дает окну программы закрыться

8: точка выхода из программы; операционной системе передается значение «0»

Последний вариант:

```
#include <cstdlib>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    cout << "Hello, World!" << endl << "Hello, C++!" << endl;
```

```
    system("Pause");
```

```
    cin.get();
```

```
    return EXIT_SUCCESS;
```

```
}
```