

Лекция 7

- Средства отладки программ
- **Контроль текста программы**

Контроль текста программы

Условие задачи:

$$y = x, \quad x < 4$$

$$y = x^2, \quad x > 4 \quad x = 0(2)8$$

$$y = x^3, \quad x = 4$$

Контроль текста программы

```
#include <stdio.h>
```

```
void main()
```

```
{ float y, x=0;
```

```
while (x<=8)
```

```
{ if (x < 4) y = x;    if (x > 4) y = x*x; if (x == 4) y= x * x * x;
```

```
printf("x=%f\t", x);
```

```
printf("y=%5.2f\n",y);
```

```
x=x+2; }
```

```
}
```

Контроль текста программы

Прокрутка:

- $x = 0;$
- $0 \leq 8; 0 < 4; y = x = 0;$ ВЫВОД $x = 0$ и $y = 0;$
- $x = x + 2 = 0 + 2 = 2$
- $2 \leq 8; 2 < 4; y = x = 2;$ ВЫВОД $x = 2$ и $y = 2;$
-
- $8 = 8; 8 > 4; y = x * x = 8 * 8 = 64;$ ВЫВОД $x = 6$ и $y = 64;$
- $x = x + 2 = 8 + 2 = 10$
- $10 > 8$ Выполнение программы закончено.

Средства отладки программ

- **Ошибки** в программах разделяются на синтаксические и логические.
- *Синтаксические* появляются при нарушении правил написания программы (обнаруживаются на этапе трансляции и могут быть легко исправлены).
- *Логические ошибки* – ошибки алгоритма и неправильного понимания операторов.
- **Процесс отладки** состоит из многократных попыток выполнения программы на компьютере и анализа получившихся результатов.

Для запуска служит команда **Отладка / Начать отладку** (или **F5**)

Тестирование – проверка правильности функционирования на различных наборах исходных данных из диапазона допустимых значений.

Средства отладки можно разделить на две группы: **средства интерактивной отладки** (позволяют выполнять программу по шагам); **средства планируемой отладки** (можно спланировать сценарий отладки на множестве шагов).

Выполнение приложения с использованием средств интерактивной отладки

- Пошаговое выполнение приложения может выполняться двумя способами:
- – **без трассировки вызываемых методов** (клавиша **F10** или команда меню **Отладка / Шаг с обходом**) или иконка на панели инструментов **Отладка** (вызывается **Вид / Панели инструментов**);
- – **с трассировкой вызываемых методов** (клавиша **F11**, или иконка на панели инструментов **Отладка** или команда меню **Отладка / Шаг с заходом**).

- Если выбрать команду **Шаг с заходом**, то отладчик войдет в функцию и начнет выполнять шаг за шагом все ее операторы.
- При выборе команды **Шаг с обходом** отладчик выполнит функцию как единое целое и перейдет к строке, следующей за вызовом функции.

Выполнение приложения с использованием средств планируемой отладки

При планируемой отладке используются **точки останова**.

Точка останова назначается *щелчком левой кнопки мыши* в специальном сером поле слева от строки текста программы.

Точку останова можно назначить также *клавишей **F9*** или командой меню **Отладка / Точка останова**.

Выполнение приложения с использованием средств планируемой отладки

- Для контроля значений рекомендуется использовать *средства непрерывного контроля состояния объектов*. Имена интересующих объектов после запуска программы в режиме отладки заносят в специальное окно просмотра **Видимые**.
- Имя заносится в столбец **Имя**.

Выполнение приложения с использованием средств планируемой отладки

- Если рядом с именем переменной стоит знак плюс, то для этой переменной может быть отображена дополнительная информация (массивы, указатели или объекты класса). Если нажать <F10> два раза и щелкнуть на «+» возле имени переменной, то отладчик отобразит значение, хранимое в памяти по адресу, содержащемуся в указателе.

Выполнение приложения с использованием средств планируемой отладки

- Вкладка **Локальные** показывает значение переменных, локальных по отношению к текущей функции.
- Вкладка **Потоки** позволяет просматривать и управлять потомками в многопоточных приложениях.
- Вкладка **Модули** перечисляет детали модулей кода, выполняемых в данный момент.

Создание исполняемого файла

- Для переключения в окончательную конфигурацию необходимо выбрать команду **Построение / Диспетчер конфигураций**.
- Выбирается опция ***Win32 Release***.
- Повторяется построение выполнением **Построение / Перестроить проект**.
- Каждая конфигурация проекта определяет также папки, куда будут помещены файлы с промежуточными и окончательными результатами компиляции и компоновки. По умолчанию это папки ***Debug*** и ***Release***, которые располагаются в папке проекта.

Назначение папок и файлов приложения

Папка решения содержит файл текущего примера решения ***.sln**, файл с информацией о проектах решения и опциях решения ***.suo**, данные **Intellisense** для решения и вложенную папку с названием решения.

В файле решения зафиксирован перечень проектов, входящих в решение.

Папка **Debug** используется для хранения временных файлов. В этой папке размещаются файлы с программным кодом на промежуточном языке. В частности, файл ***.exe** содержит программный код, который реализует функциональность приложения

Назначение папок и файлов приложения

Во вложенной папке находятся файлы и папки проекта:

* **.vcproj** в формате XML содержит перечень файлов, включенных в проект;

* **.cpp** содержит исходный код программы на языке C++;

* **.obj** – объектные файлы, содержащие машинный код исходных файлов проекта;

* **.pch** - предварительно скомпилированный файл заголовков;

* **.pdb** - файл с отладочной информацией, используемой при выполнении программы в режиме отладки;

* **.idb** - файл с информацией, необходимой для перестройки всего решения и др.