

4. Переменные, операция присваивания, ввод/вывод значений переменных. (OK)

План ответа:

1. понятие «переменная»;
2. атрибуты переменной;
3. описание переменных на языке Си;
4. операция присваивания, особенности этой операции;
5. функция printf;
6. функция scanf.

1. Переменная в программе представляет собой абстракцию ячейки памяти компьютера или совокупности таких ячеек.

2. Переменную можно охарактеризовать следующим набором атрибутов:

- Имя (идентификатор) – это строка символов, используемая для идентификации некоторой сущности в программе (переменной, функции и т. п.).
- Тип переменной определяет как переменная хранится, какие значения может принимать и какие операции могут быть выполнены над переменной.
- Адрес переменной – это ячейка памяти, с которой связана данная переменная. Адрес переменной иногда называют ее левым значением (l-value).
- Значение переменной – это содержимое ячейки или ячеек памяти, связанных с данной переменной. Значение переменной иногда называется ее правым значением (r-value).
- Область видимости переменной – это часть текста программы, в пределах которой переменная может быть использована.
- Время жизни переменной – это интервал времени выполнения программы, в течение которого переменная существует (т.е. ей выделена память).

3.

Имя может содержать буквы, цифры и символ подчеркивания.

Имя обязательно должно начинаться или с буквы, или с символа подчеркивания.

Имя не должно совпадать с ключевыми словами языка.

Имя чувствительно к регистру символов.

Длина имени практически не ограничена.

Не рекомендуется использовать имена которые начинаются с подчеркивания или совпадают с именами из стандартных заголовочных файлов;

До того, как переменная будет использована в программе, она должна быть определена.

Чтобы определить переменную, необходимо сначала указать ее тип, а затем имя.

Стандарт C99 позволяет смешивать определения переменных и операторов.

Единственное ограничение – переменная должна быть описана перед своим использованием.

4. Переменной может быть «назначено» значение с помощью присваивания. Смешение типов возможно, но не всегда безопасно. Общая форма оператора присваивания: имя_переменной=выражение;

Адресатом (получателем), т.е. левой частью оператора присваивания должен быть объект, способный получить значение, например, переменная. Если в операции встречаются переменные разных типов, происходит преобразование типов. В операторе присваивания действует простое правило: значение выражения в правой части преобразуется к типу объекта в левой части. В одном операторе присваивания можно присвоить одно и то же значение многим переменным.

5.

Для вывода текущего значения переменной используется функция printf. printf("My favorite numbers are %d and %f\n", i, f);

В строке форматирования спецификатор обозначает место, в которое будет выведено соответствующее значение во время отображения строки.

Спецификаторы начинаются с символа "%", а заканчиваются «символом преобразований», который определяет значение какого типа будет отображаться на экране.

В общем виде спецификатор записывается следующим образом:

%[флаги][ширина][.точность][размер]тип

```
// Переменных меньше, чем спецификаторов
```

```
printf("%d %d\n", i);           // 5 2293664
```

```
// Переменных больше, чем спецификаторов (не критично)
```

```
// Спецификаторы не соответствуют типам переменных
```

```
printf("%d %f\n", f, i);    // 0 0.000000
```

```
6. scanf("%d%f", &i, &f);
```

Работа функции `scanf` управляется строкой форматирования.

Для каждого спецификатора `scanf` пытается выделить данные соответствующего типа во входных данных.

Функция останавливается на символе, который не относится к очередному вводимому значению.

Если значение успешно прочитано, `scanf` продолжает обрабатывать строку. В противном случае – прекращает работу.

Первый символ, который не соответствует текущему считываемому значению, «возвращается» обратно для последующего анализа.

Строка форматирования может содержать как обычные символы, так и спецификаторы.