**Глава 7. Оператор ветвления алгоритмов switch**

**7.1 Синтаксис оператора switch**

Оператор switchпредназначен для разветвления процесса вычислений на несколько направлений – расширенная версия if - else.

В программе используется ключевое слово “switch”, после которого следует переключающее выражение: оно должно быть целочисленным, а затем перечисляются действия, которые выполняются при определенных значениях этого выражения.

Синтаксис switch начинается с ключевого слова “switch”, за которым следует выражение, которое необходимо проверить. Затем идет блок кода, заключенный в фигурные скобки.

Внутри блока кода используются несколько операторов case, каждый из которых соответствует определенному значению переменной. Оператор switch предоставляет несколько полезных возможностей. Можно использовать числа, символы или перечисления в качестве значений case. Также можно использовать ключевое слово “default”, которое будет выполнено, если ни одно из значений case не соответствует проверяемой переменной (рисунок 7.1).

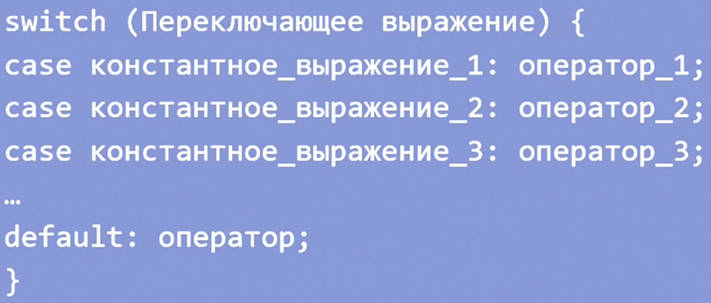


Рисунок 7.1 – Синтаксис оператора switch

Внутри блока case можно объявлять переменную, но нельзя её инициализировать [14].

Инициализация переменной требует присвоения значения в область памяти, а выполнение case, содержащего инициализацию, может никогда не потребоваться (рисунок 7.2).

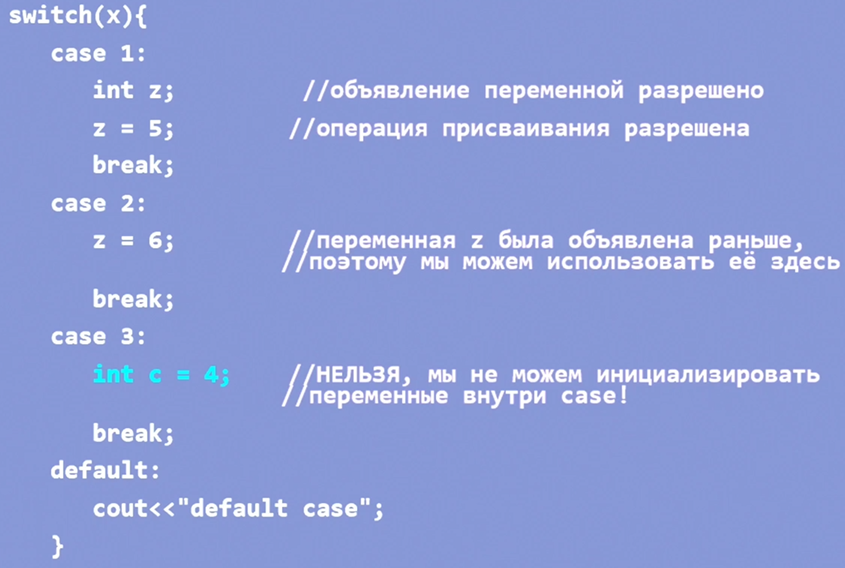


Рисунок 7.2 – Работа с переменными в операторе switch

**7.2 Правила работы с оператором switch**

1. Управление передается той ветке, для которой константное выражение совпало с переключающим. Если нет совпадения, то выполняется оператор default, который может быть только один.
2. Значение выражения, которое проверяется, должно быть целочисленного типа или типа перечисления.
3. Если default отсутствует и не было совпадений, то компилятор полностью игнорирует switch и выполняет алгоритм дальше.
4. Если внутри case не предусмотрен выход или переход, то после него последовательно выполняются все операторы, начиная с того case, куда было передано управление.
5. Каждый блок кода case должен завершаться оператором break, чтобы предотвратить выполнение последующих блоков (рисунок 7.3) [27].

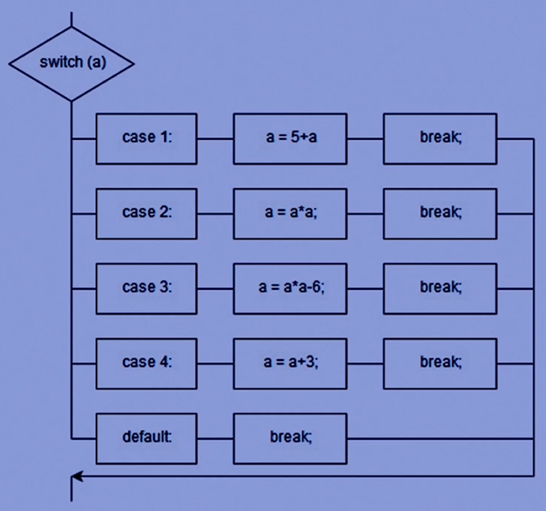


Рисунок 7.3 – Блок-схема алгоритма выполнения оператора ветвления switch

**7.3 Оператор break**

Оператор break позволяет выйти из блока кода оператора switch, когда условие в одном из блоков case выполнится. Это предотвращает выполнение последующих блоков кода [29].

Этот оператор завершает выполнение ветвления switch. Управление в этом случае передается первому оператору, следующему за конструкцией switch. Без оператора break все блоки кода после соответствующего case будут выполнены.

**7.4 Практические задания**

Для закрепления материала нужно выполнить три задачи, представленные ниже.

Задание 1:

Что будет выведено в результате выполнения программы, представленной на рисунке 7.4?

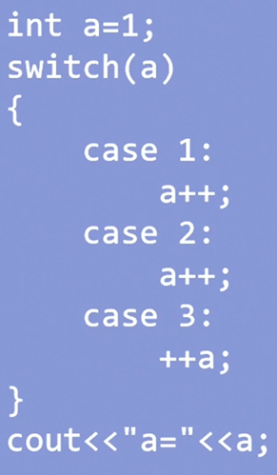


Рисунок 7.4 – Фрагмент кода программы к заданию 1

Задание 2:

Что будет выведено в результате выполнения программы, представленной на рисунке 7.5?

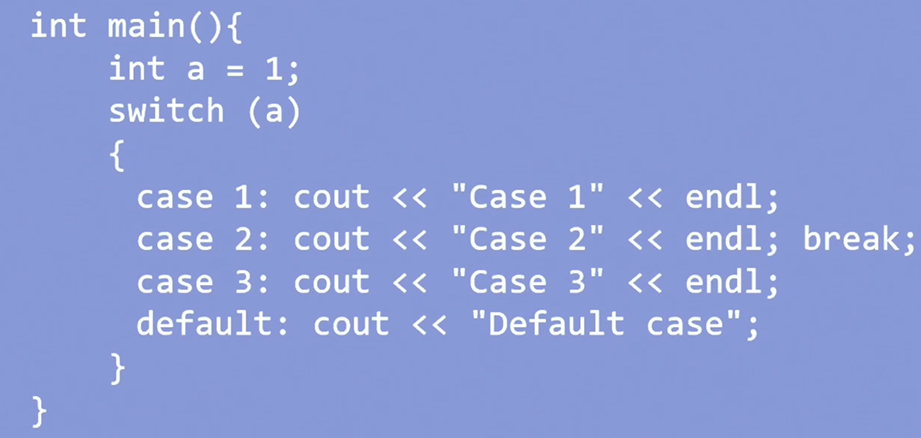


Рисунок 7.5 – Фрагмент кода программы к заданию 2

Задание 3:

Что будет выведено в результате выполнения программы на рисунке 7.6 при ch = C, при ch = A и при ch = a?

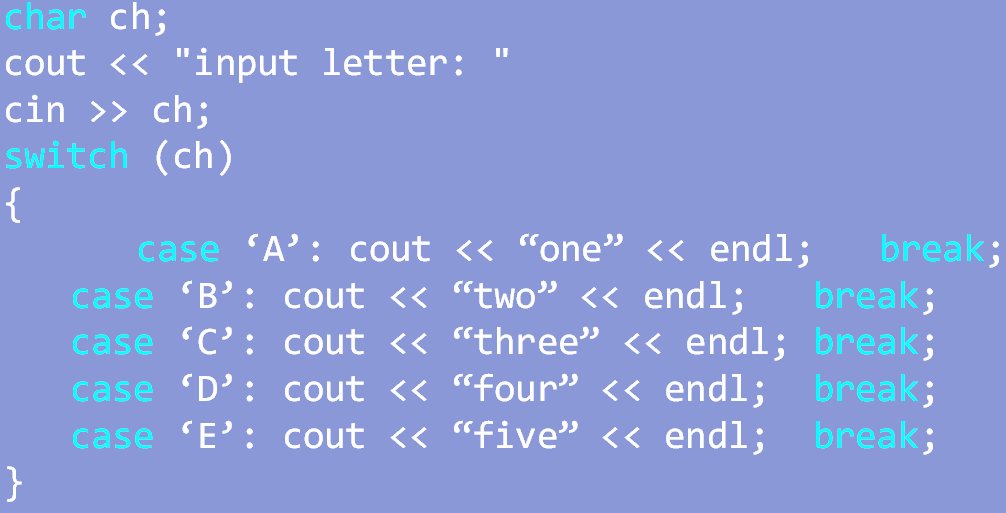


Рисунок 7.6 – Фрагмент кода программы к заданию 3