

Дмитрий Анатольевич Клосовский  
Компания «Ваш ИТ Офис»

## НОВЫЙ ПОДХОД К ОРГАНИЗАЦИИ СРЕД ПРОГРАММИРОВАНИЯ<sup>®</sup>

### 1. Введение

Обычные языки программирования используют методику написания текста программы с последующей обработкой его компилятором и переводом в машинный код. В статье предложен способ написания программных алгоритмов с использованием визуальных компонентов. Этот метод имеет значительно больше преимуществ по сравнению с общепринятым.

### 2. Основные принципы метода

В данном подходе используется палитра компонентов. Палитра компонентов содержит типовые блоки, такие как цикл, оператор условного перехода, оператор выбора и т.д. Палитра компонентов имеет контекстный характер и визуализируется в нужном месте программы при нажатии на этом месте правой кнопкой мыши. Далее следует выбрать на палитре компонентов нужный блок, после чего он появится в теле программы. Компонент по своей сути представляет таблицу, шаблон в котором нужно просто ввести нужные параметры. Посмотрим, как реализуется цикл со счетчиком (Рис. 1).

Для упрощения реализации программы следует создать возможность перемещения визуальных компонентов по телу программы с определенным шагом (Рис. 2).

Т.е. компонент вверх-вниз, как и влево-вправо может перемещаться только с каким-то шагом. Т.е. по сетке. Это сделано для оптимизации процесса написания алгоритма программы. Сетка перемещения должна быть визуализирована, но не отвлекать от основных компонентов.

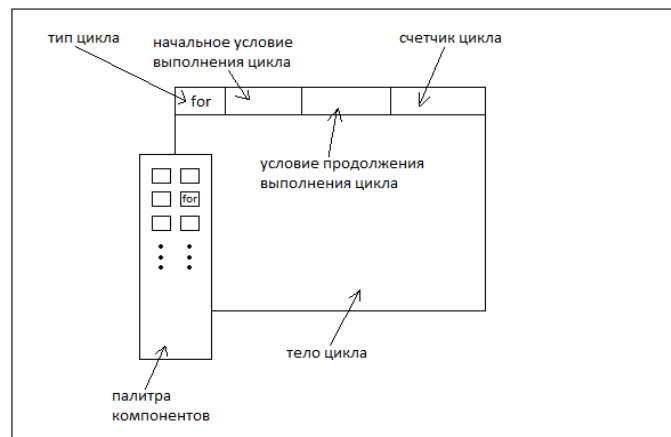


рис.1 Объявление цикла с помощью визуального подхода

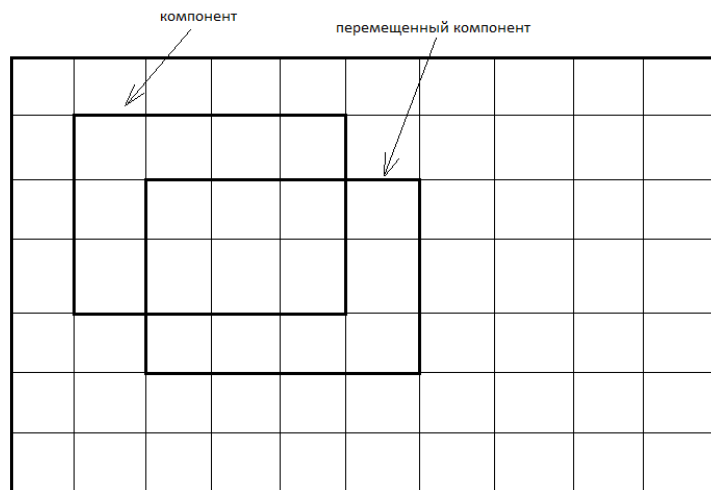


рис.2 Сетка для перемещения визуальных компонентов

### 2.1. Объявление переменных

Приведем пример объявления переменных программы (Рис. 3):

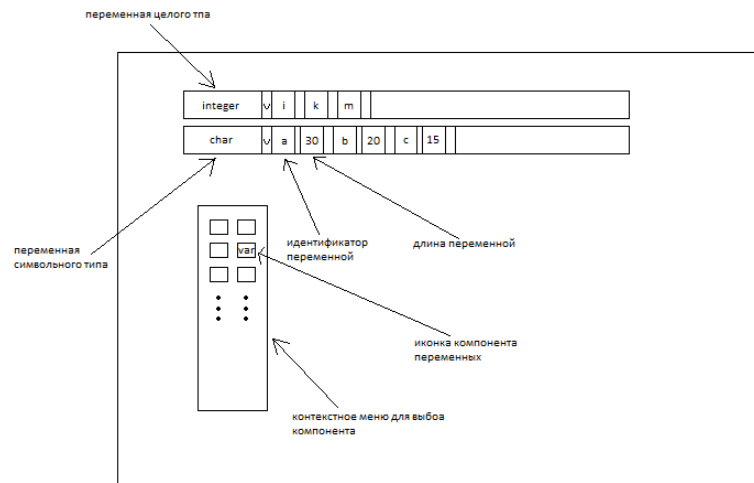


рис.3 Объявление переменных с помощью визуальных компонентов

Мы перетаскиваем с палитры компонентов компонент объявления переменных, после чего он визуализируется в теле программы. В компоненте задается тип содержащихся в нем переменных, а также их имена и другие характеристики.

## 2.2. Тело программы

### 2.2.1. Циклы

Любой язык программирования включает три вида циклов: с предусловием, с постусловием и со счетчиком. Также в любой язык программирования входит оператор перехода, условного и безусловного, а также оператор выбора. Также в состав любого языка программирования входят операции и функции. Т.е. состав компонентов любого языка программирования практически идентичен и невелик.

Любой оператор, такие как оператор цикла, условного или безусловного перехода, а также выбора должен выглядеть в виде визуального элемента (Рис. 4):

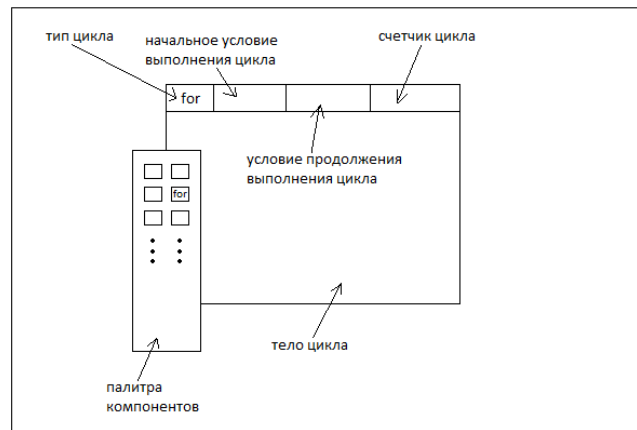


рис.4 Пример объявления цикла с помощью визуальных компонентов

Оператор цикла появляется в теле программы выбором его из контекстного меню палитры компонентов. В его шапке вводятся начальные условия выполнения цикла, счетчик, а также условия продолжения выполнения. В теле цикла размещаются исполняемые структуры. Циклы с постусловием и с предусловием формируются по такому же принципу.

### 2.2.2. Оператор выбора и условного перехода

Оператор выбора (Рис. 7):

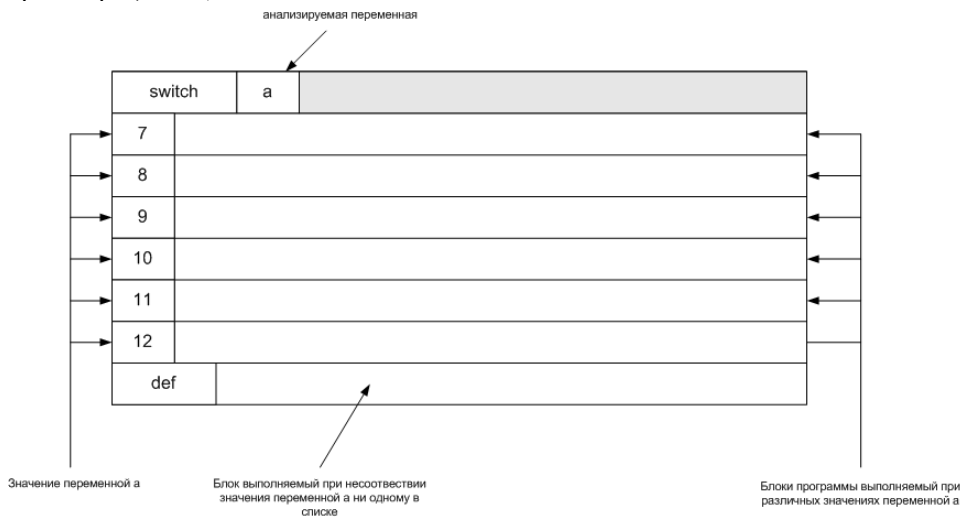


рис.7 Объявление оператора выбора

### 2.2.3. Объявление функций

Рассмотрим использование функций и операций присвоения. Компонент функций расположен на палитре компонентов. Он вносится в тело программы простым выбором из контекстного меню.

После этого визуальный компонент функции появляется в теле программы. Должны быть разные виды компонентов. Компоненты, содержащие математические, строковые, функции и т.д. Название функции выбирается из combo box-a, расположенного на компоненте функции, либо при помощи поиска за счет посимвольного ввода названия функции в поле combo box-a. Если у функции есть параметры, то поля для их редактирования появляются в компоненте функции.

### 2.2.4. Оператор перехода

Рассмотрим объявление оператора перехода. Оператор содержится в компоненте функция и выбирается из списка. Оператор, к которому осуществляется переход, задается при помощи установления порядка выполнения операторов программы. Методика задания последовательности выполнения операторов описывается далее.

### 2.2.5. Оператор присвоения

Далее будет рассмотрена операция присвоения. На палитре компонентов существует компонент переменной. Рассмотрим процесс присвоения выходного параметра функции переменной.

Для присвоения переменной  $a$  значения функции  $fun$  следует вначале перетащить компонент функции и переменной в область тела программы, после чего задать им все необходимые параметры. Далее следует нажать левой клавишей мыши на компонент функции и, не отпуская клавиши мышки перетащить курсор на компонент переменной, после чего следует отпустить клавишу мышки. В результате операция присвоения будет создана.

### 2.2.6. Выражения

Далее речь пойдет об объявлении выражений. Для этой цели введен компонент выражение (Рис. 13).

Для этого необходимо перетащить в тело программы компонент выражение. После этого следует в нем задать все необходимые функции и прочие атрибуты, а также все операции между ними. Далее результат вычисления выражения можно присвоить какой-либо переменной.

### 2.3. Установление порядка выполнения операторов

Порядок выполнения операторов в блоке либо самих блоков задается простым перетаскиванием курсора мыши от одного оператора к другому. Т.е. следует навести курсор на оператор, нажать левую кнопку мыши, перетащить курсор на другой оператор или блок программы и отпустить клавишу мыши.

### 3. Преимущества данного метода по сравнению с общепринятыми способами

Этот метод экономит пространство для алгоритма программы. Не обязательно отводить под один оператор целую строку экрана. Можно создать последовательность операторов и тут же правее и левее создать другую цепочку операторов, после чего установить последовательность выполнения операторов на цепочку рядом.

Подобная компрессия и в обычных средах разработчиками, когда сразу несколько операторов пишутся в одну строчку. Но на практике она практически не применяется из-за сильного снижения читаемости программного алгоритма. При использовании визуальных компонентов такие проблемы не возникают за счет наглядности и четкой структурированности при использовании данного подхода.

Этот метод лучше воспринимается разработчиком, так как основан на визуальном подходе. Это происходит за счет того, что используется более богатая палитра цветов, форм и символики, нежели при реализации текстовым методом, к тому же структурированность программы сильно повышается за счет применения функции задания последовательности выполнения операторов.

Этот метод позволяет частично производить проверку корректности кода в процессе его создания. Например, невозможно в компоненте переменная или функция задать несуществующую переменную или функцию т.к. их названия выбираются из выпадающего списка. Также невозможно неправильно создать цикл, пропустить открывающую или закрывающую скобку. Т.к. нужно всего лишь поместить тело компонента цикла в тело программы и задать нужные параметры.

Этот метод упрощает создание алгоритмов за счет использования визуальных компонентов. Например, чтобы создать переменную необходимо в нужном месте нажать правую кнопку мыши и выбрать из контекстной палитры компонентов компонент «переменная», после чего в выпадающем списке этого компонента выбрать название переменной. Либо возможен посимвольный ввод в поле названия переменной в результате в поле ввода названия переменной будут отображаться идентификаторы, соответствующие вводимому шаблону. Такая же ситуация наблюдается и с функциями. Названия у функций намного длиннее и сложнее. Таким образом, данный подход имеет ещё большую актуальность при работе с функциями.

Если, к примеру, требуется создать цикл, необходимо в нужном месте тела программы из контекстной палитры компонентов выбрать компонент цикл, после чего ввести в поля компонента необходимые параметры. Также отсутствует необходимость долгого и усердного форматирования и структурирования тела цикла как при текстовом подходе программирования. Для изменения местоположения и размеров цикла необходимо всего-навсего перетащить мышкой компонент в другое место и также мышкой изменить положения правого нижнего угла компонента цикла.

---