

контроллинга – это координация целей разнообразных средств, уровней и методов их осуществления, чтобы получить положительные результаты.

Следующая функция контроллинга – это контроль внутри предприятия. Функция заключается в контроле экономической работы его отделов и организации в целом. Система контроллинга подразделения не должна принимать решения, указания и санкции. Все перечисленное – это задача ревизии, а контроллинг ориентируется на результаты деятельности на данный момент и не связан с проверкой документов на предприятиях.

Изучение контроллинга в Российской Федерации необходимо не только преподавателям, студентам и аспирантам, но и всем, интересующимся иностранным опытом организации управления, контроля и учета, а также желающим расширить свой кругозор. В рыночной экономике затраты и результаты деятельности организаций зависят от эффективности управления, объема и качества работы каждого подразделения и сотрудника. Чтобы объективно оценить эти параметры, нужно изменить систему составления планов, нормирования, отчетности и учета и направить ее сначала на удовлетворение потребностей управления внутри фирмы. В скором будущем и в нашей стране контроллинг станет неотъемлемой частью в системе управления компанией. Для этого необходимо ввести во многих организациях системы антикризисного управления, которые ориентированы на реформирование предприятия, а также внедрение в них долгосрочного внутреннего планирования. Опыт зарубежных фирм показывает, что долгосрочное планирование тесно связано с системой контроллинга.

УДК 004.051

Студ. Н.С. Егорова
Рук. Т.С. Крайнова
УГЛТУ, Екатеринбург

СТИЛЬ ОФОРМЛЕНИЯ ПРОГРАММЫ

Информация, содержащаяся в программном коде, сосредоточена плотней, чем информация в большинстве книг. Если страницу книги можно прочесть и понять за 1 или 2 минуты, то большинство программистов не могут читать и понимать листинг программы, составленной другим разработчиком, со скоростью, даже приблизительно сравнимой с этой. Поэтому текст программы должен давать гораздо больше подсказок, чем любая книга.

Форматирование – это ключ к структуре программы. Компьютеру важен исключительно смысл и последовательность информации, передаваемой операторами, а читатель-человек склонен делать выводы из визуального представления кода.

Хорошее визуальное форматирование программы лучше отражает ее логическую структуру. Создание красивого кода хорошо, а демонстрация правильной структуры кода – лучше. Если одна методика лучше показывает структуру кода, а другая выглядит красивей, следует использовать ту, которая лучше демонстрирует структуру [1].

Можно получить хороший формат кода, по-разному используя несколько основных инструментов. Исходя из того, что любая программа неоднократно просматривается разработчиками, следует придерживаться хорошего стиля ее написания [2].

Правила именования объектов программы

При выборе имен программных объектов следует соотносить с их содержанием и назначением (например, MaxEl – максимальный элемент или NextEl – следующий элемент). Если позволяет язык программирования, можно использовать символ “_” для визуального разделения имени, состоящего из нескольких слов (например, Max_El или Next_El). Следует избегать близких по написанию имен, например, Index и InDec.

Правила оформления модулей

Каждый модуль должен предваряться заголовком, который содержит название модуля, краткое описание его назначения, краткое описание входных и выходных параметров с указанием единицы измерения, список используемых модулей, ФИО автора программы, номер версии или дату последней корректировки. Например:

```
{*****}
{* Функция: Length_Path(n:word; L:array of real):real *}
{*   Цель: определение суммарной длины отрезков *}
{*   Исходные данные: *}
{*     n – количество отрезков, *}
{*     L – массив длин отрезков (в метрах) *}
{*   Результат: длина (в метрах) *}
{*   Вызываемые модули: нет *}
{*   Назначение: отрезки суммируются методом накопления, *}
{*               n>=0 *}
{*   Дата: 25.10.2016   Версия 1.01. *}
{*   Автор: И.И. Иванов *}
{*   Исправления: нет *}
{*****}
```

Стиль оформления модулей

Стиль оформления текстов определяет использование отступов, пропусков строк и комментариев, облегчающих понимание программы и разделяющих модуль на логические части.

Для таких языков, как Delphi, Pascal, C++ и Java, наличие отступов позволяет прояснить структуру программы: дополнительный отступ обозначает вложение операторов языка, например:

```
Amax:=A[1,1];
For i:=1 to n do
  For j:=1 to m do
    If A[i,j]>Amax then Amax:=A[i,j];
```

Комментарии следует использовать для разъяснения цели выполнения тех или иных действий, а также для группы операторов, связанных общим действием.

Для языков низкого уровня, например Ассемблера, стиль, облегчающий понимание, определить сложнее. Целесообразно комментировать и блоки операторов, и каждый оператор. Например:

```
; цикл суммирования элементов массива
;   установки цикла
    mov    AX, 0           ; обнуляем сумму
    mov    CX, n          ; загружаем счетчик цикла
    mov    BX, 0          ; смещение первого элемента массива
;   тело цикла
Cycle: add    AX, a[BX]    ; добавляем элемент
      add    BX, 2        ; определяем адрес следующего
      loop  cycle        ; цикл на n повторений
;   выход из цикла при обнулении счетчика
```

Пустые строки позволяют продемонстрировать организацию программы. В среднем оптимальное число пустых строк в программе должно составлять от 8 до 16 %. Если оно больше 16 %, то время на отладку модуля значительно увеличивается. Например:

```
{проверка количества отрезков и выход, если отрезки не заданы}
  If n<0 then
    Begin
      WriteLn('Количество отрезков отрицательное');
      Exit;
    End;
```

```
{цикл суммирования длин отрезков}
  S=0;
  For i:=0 to n-1 do S:=S+Len[i];
```

Таким образом, хорошая схема форматирования единообразно показывает логическую структуру кода, упрощает чтение кода, что позволяет лучше переносить модификацию кода, а также поиск и исправление ошибок программирования. С точки зрения технологичности хорошим считают стиль оформления программы, облегчающий ее восприятие как самим

автором, так и другими программистами, которые будут проверять ее или модифицировать. «Помните, программы читаются людьми», призывал Д. Ван Тассел в своей монографии, посвященной проблемам программирования [3].

Библиографический список

1. Гагарина Л.Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Виснадул. – М.: ТД «Форум»: ИНФРА-М, 2009. – 400 с.
2. Иванова Г.С. Технология программирования: учебник / Г.С. Иванова. – М.: КНОРУС, 2011. – 336 с.
3. Тассел Д. Ван. Стиль, разработка, эффективность, отладка и испытание программ: пер. с англ. – М.: Мир, 1985. – 187 с.

УДК 005.95 + 005.96

Маг. Е.В. Ишков
Рук. Н.В. Сырейщикова
ЮУрГУ, Челябинск

ОСВОЕНИЕ МЕТОДОВ НАСТАВНИЧЕСТВА ДЛЯ СОВЕРШЕНСТВОВАНИЯ ПРОЦЕССА ОБУЧЕНИЯ ПЕРСОНАЛА В УСЛОВИЯХ ОАО «ЧЭМК»

*Инвестиции в людей никогда не бывают ошибочными.
В худшем случае они не принесут немедленной отдачи.
В лучшем случае они станут решающим конкурентным фактором.*

Б. Карлоф, С. Седерберг

На сегодняшний день общепризнанно, что главная ценность организации – это ее персонал и при прочих схожих условиях именно качество персонала определяет конкурентные преимущества предприятия. В связи с этим важно своевременно повышать качество человеческого капитала персонала в первую очередь за счет обучения.

На кафедре технологии автоматизированного машиностроения ЮУрГУ совместно с ОАО «ЧЭМК» проведена НИР для совершенствования процесса обучения персонала, который по оценке предприятия функционирует не в полной мере: наблюдаются серьезные недостатки в обучении персонала и, как следствие, появляется необеспеченность в полной мере компетентным персоналом, а также текучесть кадров.