

МЕТОДИКА ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

Т.А. Гамова

ПРОГРАММИРОВАНИЕ ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ КАК СРЕДСТВО ФОРМИРОВАНИЯ ИНТЕГРАТИВНОГО МЫШЛЕНИЯ

Интенсивные технологии образовательного процесса, ориентированные на развитие личностных потенциалов студентов особенно актуальны в современном обществе в связи с адаптацией отечественной промышленности в области автоматизации производства с западными технологиями.

На рынке труда особенно востребованы выпускники вузов электротехнического направления, владеющие компетенциями по программированию логических контроллеров. Автоматизация производства на программном уровне является на сегодняшний период в мире общей тенденцией развития производства.

Программирование логических контроллеров предназначены языки программирования, соответствующие международному стандарту DIN EN 6.1131-3: STL (assembler-подобный язык), LAD (представление в виде логических схем), FBD (язык функциональных блок-схем), SCL («Structured Control Language» – «структурированный язык управления», Pascal-подобный язык высокого уровня), S7-GRAPH (для графической разработки программ систем автоматизации SIMATIC в виде последовательности шагов и переходов между ними), S7-HiGraph (для графической разработки программ систем автоматизации SIMATIC в виде графа состояний системы и переходов между ними) и CFC («continuous function chart» – план соединений программных блоков). Пользователю предоставляется полное право выбора из этого набора различных методов представления для описания функций при решении задач программирования контроллеров.

Например, программировать контроллеры на SCL дают дополнительные возможности сначала решать вычислительные задачи, а по результату принимать решение по управлению технологическим процессом.

```
//- DB1 контейнер для a и b -----  
DATA_BLOCK DB1
```

```
STRUCT
```

```
buffer_a : ARRAY[1..10] OF INT := 10(0); // несортированные a
```

```

    buffer_b : ARRAY[1..10] OF INT := 10(0);    // несортированные b
END_STRUCT
BEGIN

END_DATA_BLOCK

//- DB2 контейнер для a, b и длины c, отсортированные по c -----
DATA_BLOCK DB2

STRUCT
    sort_a : ARRAY[1..10] OF INT;    // значение a
    sort_b : ARRAY[1..10] OF INT;    // значение b
    sort_c : ARRAY[1..10] OF REAL;    // длина c
END_STRUCT
BEGIN

END_DATA_BLOCK

//- DB3 контейнер для площадей и среднего значения -----
DATA_BLOCK DB3

STRUCT
    area : ARRAY[1..10] OF REAL;    // площади
    mean : REAL;    // среднее значение площадей
END_STRUCT
BEGIN

END_DATA_BLOCK

//- FC63 вычисление площадей -----
FUNCTION FC63 : VOID

AUTHOR : HF
NAME : Flaeche
FAMILY : Training
// KNOW_HOW_PROTECT

CONST
    BUFLength := 10;    // длина буфера

```

END_CONST

VAR_INPUT

DBS_A : ANY; // значение a

DBS_B : ANY; // значение b

END_VAR

VAR_OUTPUT

DB_RESULT : ANY; // значения площадей

END_VAR

VAR_TEMP

i : INT; // индекс цикла

val_a : ARRAY[1..BUFLNGTH] OF INT; // временный массив для a

val_b : ARRAY[1..BUFLNGTH] OF INT; // временный массив для b

val_area : ARRAY[1..BUFLNGTH] OF REAL; // временный массив для
площадей

ret : INT;

END_VAR

BEGIN

ret := SFC20(SRCBLK:=DBS_A,DSTBLK:=val_a); // взять массив для a

ret := SFC20(SRCBLK:=DBS_B,DSTBLK:=val_b); // взять массив для b

FOR i := 1 TO BUFLNGTH BY 1 DO // вычислить все площади

val_area[i] := val_a[i] * val_b[i] / 2;

END_FOR;

ret := SFC20(SRCBLK:=val_area,DSTBLK:=DB_RESULT); // сохранить
площади

END_FUNCTION

//- FC64 вычисление среднего значения площадей -----

FUNCTION FC64 : VOID

AUTHOR : HF

NAME : Mittel

```

FAMILY : Training
// KNOW_HOW_PROTECT

CONST
  BUFLNGTH := 10;      // длина буфера
END_CONST

VAR_INPUT
  DB_AREA : ANY;      // DB-массив для площадей
END_VAR

VAR_OUTPUT
  DB_RESULT : ANY;    // DB-массив для среднего
  MEAN : REAL;
END_VAR

VAR_TEMP
  i : INT;
  val_mean : REAL;
  val_area : ARRAY[1..BUFLNGTH] OF REAL;
  ret : INT;
END_VAR

BEGIN
  ret := SFC20(SRCBLK:=DB_AREA,DSTBLK:=val_area); //чтение массива
площадей
  val_mean := 0;
  i := 1;

  WHILE i <= BUFLNGTH DO          // прибавить площадь
    val_mean := val_mean + val_area[i] ;
    i:= i + 1;
  END_WHILE;

  val_mean := val_mean / BUFLNGTH; // вычислить среднее
  MEAN := val_mean;              // передать на выход

  ret := SFC20(SRCBLK:=val_mean,DSTBLK:=DB_RESULT); //сохранить
среднее

```

END_FUNCTION

//- FC62 сортировка длин -----

FUNCTION FC62 : VOID

AUTHOR : HF

NAME : Sorter

FAMILY : Training

// KNOW_HOW_PROTECT

CONST

BUFLength := 10; // длина буфера

END_CONST

VAR_IN_OUT

US_VALUES: // IN: несортированное,OUT:сортированное
//значение

STRUCT // структура массивов

val_a : ARRAY[1..BUFLength] OF INT; // значений a

val_b : ARRAY[1..BUFLength] OF INT; // значений b

val_c : ARRAY[1..BUFLength] OF REAL; // длин c

END_STRUCT;

END_VAR

VAR_TEMP

i,tmp_a,tmp_b : INT;

tmp_c : REAL;

swapped : BOOL;

s_values : STRUCT // временная структура для значений

val_a : ARRAY[1..BUFLength] OF INT;

val_b : ARRAY[1..BUFLength] OF INT;

val_c : ARRAY[1..BUFLength] OF REAL;

END_STRUCT;

END_VAR

BEGIN

```
s_values := US_VALUES; //копирование значений в локальную структуру
STRUCT->
```

```
    //сортировка быстрее. L-Stack OB35 должен быть увеличен
REPEAT          // сортировка значений a, b и C, в зависимости от C
    swapped := FALSE;
```

```
FOR i := BUFLLENGTH TO 2 BY -1 DO
    IF s_values.val_c[i-1] > s_values.val_c[i]
    THEN // смена позиций в буфере
        tmp_a      := s_values.val_a[i];
        tmp_b      := s_values.val_b[i];
        tmp_c      := s_values.val_c[i];
        s_values.val_a[i] := s_values.val_a[i-1];
        s_values.val_b[i] := s_values.val_b[i-1];
        s_values.val_c[i] := s_values.val_c[i-1];
        s_values.val_a[i-1] := tmp_a;
        s_values.val_b[i-1] := tmp_b;
        s_values.val_c[i-1] := tmp_c;
        swapped      := TRUE;
    END_IF;
END_FOR;
```

```
UNTIL NOT swapped // повторять, пока все не отсортировано
END_REPEAT;
```

```
US_VALUES := s_values; // выходное отсортированное значение
```

```
END_FUNCTION
```

```
//- FC61 вычисление длин -----
FUNCTION FC61 : VOID
```

```
AUTHOR : HF
NAME   : Abstand
FAMILY : Training
// KNOW_HOW_PROTECT
```

```
CONST
    BUFLLENGTH:=10; // длина буфера
```

END_CONST

VAR_INPUT

DB_A : ANY; // значение a

DB_B : ANY; // значение b

END_VAR

VAR_OUTPUT

DBS_A : ANY; // сортированные значения a

DBS_B : ANY; // сортированные значения b

DBS_C : ANY; // сортированные значения c

END_VAR

VAR_TEMP

values: STRUCT // временные структуры значений

val_a : ARRAY[1..BUFLNGTH] OF INT;

val_b : ARRAY[1..BUFLNGTH] OF INT;

val_c : ARRAY[1..BUFLNGTH] OF REAL;

END_STRUCT;

ret,index : INT;

END_VAR

BEGIN

ret := SFC20(SRCBLK:=DB_A,DSTBLK:=values.val_a); // взять a из DB

ret := SFC20(SRCBLK:=DB_B,DSTBLK:=values.val_b); // взять b из DB

index := 1; // инициализация индекса

WHILE index <= BUFLNGTH DO // вычисление длин

values.val_c[index]:= SQR(SQR(values.val_a[index]) +
SQR(values.val_b[index]));

index:=index +1;

END_WHILE;

FC62(US_VALUES:=values); // вызов FC62 для сортировки

ret := SFC20(SRCBLK:=values.val_a,DSTBLK:=DBS_A); // сохранить a

```
ret := SFC20(SRCBLK:=values.val_b,DSTBLK:=DBS_B); // сохранить b
ret := SFC20(SRCBLK:=values.val_c,DSTBLK:=DBS_C); // сохранить
длины
```

```
END_FUNCTION
```

```
//- FB2 сохранить координаты в буфере -----
FUNCTION_BLOCK FB2
```

```
AUTHOR : HF
NAME : Speichern
FAMILY : Training
// KNOW_HOW_PROTECT
```

```
CONST
  BUFLength := 10; // длина буфера
END_CONST
```

```
VAR_INPUT
  NewValue : INT; //координаты
END_VAR
```

```
VAR_IN_OUT
  BUF_DB : ANY; // буфер в DB
END_VAR // ANY неразрешен для VAR_OUTPUT
```

```
VAR_OUTPUT
  READY : BOOL:=FALSE; // все переменные сохранены
END_VAR
```

```
VAR
  index : INT := 1; // счетчик № сохраненных значений
  field : ARRAY[1..BUFLength] OF INT; // контейнер для сохранения
значений
  ret : INT;
END_VAR
```



```

BEGIN
  IF index > BUFLNGTH THEN      // очистка буфера
    index := 1;
    READY := FALSE;
  END_IF;

  field[index]:= NewValue;      // сохранить значение

  IF index = BUFLNGTH THEN      // буфер полон
    ret := SFC20(SRCBLK:=field,DSTBLK:=BUF_DB); //сохранение значе-
ний в DB
    READY := TRUE;              //все значения сохранены
  END_IF;

  index := index +1;

END_FUNCTION_BLOCK

//- FC60 генератор случайных чисел -----
FUNCTION FC60 : INT;

AUTHOR : HF
NAME   : Zufall
FAMILY : Training
// KNOW_HOW_PROTECT

VAR_INPUT
  RAND:INT;      // инициализация
END_VAR

CONST
  m:=101;        // случ. число от 0 до 100
END_CONST

BEGIN          // вычисление случ. числа

  FC60 := (ABS(RAND*DINT_TO_INT(TIME_TO_DINT(SFC64())))) MOD m;

END_FUNCTION

```

```

//- OB 35 -----
ORGANIZATION_BLOCK OB35

VAR_TEMP
  info : ARRAY[0..19] OF BYTE;
  random: INT;
END_VAR

BEGIN
  random := FC60(RAND:=711);           //создание случ. числа a
  FB2.DB11(NewValue:=random,BUF_DB:=DB1.buffer_a); //его сохранение

  random := FC60(RAND:=13);           // создание случ. числа b
  FB2.DB12(NewValue:=random,BUF_DB:=DB1.buffer_b); // его сохранение

  IF DB12.READY = TRUE THEN           //если буфер полон
    //вычислить длину
    FC61(DB_A:=DB1.buffer_a,DB_B:=DB1.buffer_b, //сортировать значе-
ния
    DBS_A:=DB2.sort_a,DBS_B:=DB2.sort_b,DBS_C:=DB2.sort_c);
  END_IF;

END_ORGANIZATION_BLOCK

```

```

//- OB 1 -----
ORGANIZATION_BLOCK OB1

VAR_TEMP
  info : ARRAY[0..19] OF BYTE;
  dummy : REAL;           // неиспользуемый выход FC64
END_VAR

BEGIN

  IF I0.0 <> M0.1 AND I0.0 = TRUE THEN //если на I0.0 положительный
фронт, то
    //вычислить площади и среднее
    FC63(DBS_A:=DB2.sort_a,DBS_B:=DB2.sort_b,DB_RESULT:=DB3.area);

```

```
FC64(DB_AREA:=DB3.area,DB_RESULT:=DB3.mean,MEAN := dummy);  
END_IF;  
M0.1 := I0.0;  
END_ORGANIZATION_BLOCK
```

Из вышеприведенного примера следует, что содержание образования должно ориентироваться на создание условий по овладению широким диапазоном профессиональных знаний, умений и навыков, одним из которых является программирование логических контроллеров, которое невозможно получить без системных, интегративных знаний. Интегративные знания формируются на основе интеграции дисциплин «Технологии программирования», «Теоретические основы электротехники», «Моделирование электрических и магнитных цепей», «Электрические машины», «Теория автоматического управления», «Теории автоматов» «Физические основы электроники» и др., что позволяет студентам создавать интегративные проекты.

Исследователи по-разному определяют сущность проектного обучения [1, 6, 8] Они рассматривают его как тип обучения, позволяющий студентам через самостоятельную постановку и решение технических проблем переосмыслить имеющийся опыт в контексте реально существующих задач профессиональной деятельности, получить необходимые знания, развить навыки рефлексии в ходе планирования, реализации и оценки проекта. В центре проектного обучения находится студент как субъект деятельности, а педагог создает условия для повышения уровня его профессиональной (и, в частности, рефлексивной) готовности.

В нашем исследовании под «проектом» мы понимаем результат программирования логических контроллеров, с помощью которых осуществляется автоматизация конкретного технологического процесса.

Нами были определены следующие этапы формирования компетенции по программированию логических контроллеров: учебно-познавательная, учебно-профессиональная и творческий этап деятельности.

Под «учебно-познавательную деятельность» мы понимаем деятельность по овладению теоретических знаний по «Теории автоматов», «Физические основы электроники», «Технология программирования», т.е. те основные знания, без которых невозможно осуществлять программирование контроллеров. На данном этапе приобретаются интегративные знания, на основе которых формируется интегративное мышление будущих выпускников. В ходе учебно-профессиональной деятельности, которая осуществляется на производственной практике, перед студентами ставятся задачи по автоматизации определенного технологического процесса через программирования логических контроллеров,

в результате чего они получают опыт выполнения прикладных исследований по научно-техническим разработкам под управлением опытного наставника. На творческом этапе студенты решают производственные задачи по автоматизации технологического процесса с помощью программирования логических контроллеров без помощи наставника. На этом этапе они имеют возможность проявить самостоятельность, активность, способность проектировать свою деятельность, самостоятельно принимать решения и нести ответственность за них, критично оценивать результаты своих действий в соответствии с социально-профессиональными ценностями.

Под социально-профессиональными ценностями мы понимаем единство наиболее признаваемых и принимаемых социально-профессиональным сообществом ориентиров поведения, личностного отношения к целям и результатам своей профессиональной деятельности.

Таким образом, приобретение опыта программирования логических контроллеров невозможно без наличия широкого диапазона профессиональных знаний, а сам процесс программирования является средством формирования интегративного мышления, без которого невозможно решать задачи автоматизации производства.

Библиографический список

1. Дьюи Д., Дьюи Э. Школы будущего // Народное образование. - 200С № 8. - С. 244-269.
2. Селевко Г.К. Современные образовательные технологии: Учеб. пособие. - М.: Нар. образование, 1998. - 255 с.
3. Шишов С.Е., Кальней, В.А. и др. Структура и содержание проектной деятельности: Метод проектов в России и за рубежом // Стандарты и мониторинг в образовании. - 2004. - № 5. - С. 23-31.

Н.В. Зинякова, И.А. Ридингер

АКТУАЛЬНЫЕ ВОПРОСЫ ТРУДОУСТРОЙСТВА МОЛОДЫХ СПЕЦИАЛИСТОВ

Существует более значимая проблема в период кризиса – это получение в период обучения молодыми людьми тех специальностей, которые не являются востребованными на рынке труда или которыми рынок уже перенасыщен.

В кризисных условиях мировой экономики проблемы с трудоустройством затронули не только работающее население, но и особенно молодых специали-