

Лекция «Объектно-ориентированное программирование» (часть 1)

Овчинников П.Е.
МГТУ «СТАНКИН»,
ст.преподаватель кафедры ИС

Языки программирования

ГОСТ 19781-90 Обеспечение систем обработки информации программное. Термины и определения

34. Декларативный язык (англ. Declarative language)

Язык программирования для выражения определений.

Примечание. В качестве такого языка часто выступает язык описания данных

25. Проблемно-ориентированный язык (англ. Problem-oriented language)

Язык программирования, который соответствует понятиям определенного класса прикладных задач.

Примечание. Проблемно-ориентированный язык обычно имеет набор специфических изобразительных средств

35. Объектно-ориентированный язык (англ. Object-oriented language)

Язык программирования, который соответствует концепциям объектно-ориентированного программирования

36. Процедурный язык (англ. Procedural language)

Язык программирования, в котором действия над данными выражаются в терминах последовательностей команд

37. Функциональный язык (англ. Functional language)

Язык программирования, в котором действия над данными выражаются в виде обращений к функциональным процедурам

Методы программирования

ГОСТ 19781-90 Обеспечение систем обработки информации программное. Термины и определения

46. Структурное программирование (англ. Structured programming)

Метод построения программ, использующий только иерархически вложенные конструкции, каждая из которых имеет единственную точку входа и единственную точку выхода.

Примечание. В структурном программировании используются три вида структур, связанных с передачей управления: последовательная, условного перехода и циклическая

47. Объектно-ориентированное программирование (англ. Object-oriented programming)

Метод построения программ как совокупностей объектов и классов объектов, которые могут вызывать друг друга для выбора и выполнения операций.

Примечание. Объекты состоят из данных и операций над данными

48. Логическое программирование (англ. Logical programming)

Метод построения программ как совокупности логических правил с предварительно определенными алгоритмами для обработки входных данных программы в соответствии с ее правилами

Группа FORTRAN

FORTRAN (Фортран) 66 – 90 (1991)

```
_____PRINT*, 'Hello, World!'  
_____END
```

```
Real:: arr(I1:I2,J1:J2,K1:K2), arr1(I1:I2,J1:J2,K1:K2), arr2(I1:I2,J1:J2,K1:K2)  
Real:: frr(100), frr1(10)/1,2,3,3*4,4*5/  
      ! или  
Real:: frr1(10)=(/1,2,3,4,4,4,5,5,5,5/)  
  
...  
arr=1.  ! присваивание массива (встроенная перегрузка для оператора присваивания)  
arr1=Sin(arr)+arr ! элементная функция sin применяется к каждому элементу массива  
  
arr2(I1:I2:1,J1:J2:2,K2:K1:-4)=arr1(I1:I2:1,J1:J2:2,K2:K1:-4) ! присваивание элементов с шагом 1, 2 и -4 (назад), заданное  
индексным триплетом  
  
frr=(/(J, J=1,100)/) ! присваивание одномерного массива посредством циклического списка  
  
Forall(i=I1:I2, j=J1:J2, k=K1:K2, arr(i,j,k)>0.) brr(i,j,k)=Log(arr(i,j,k)) ! замена циклов и условных операторов и  
конструкций. Маскирование присваивания (маска – arr(i,j,k)>0.)  
  
Forall(i=1:N, j=1:N, k=1:N) crr(i,j,k)=Sin(0.5*(i+j)-k) ! расширение возможностей сечений  
  
Forall(i=1:100) ! конструкция Forall для нескольких операторов присваивания  
  drr(i,i)=0.    ! доступ к диагонали матрицы  
  err(i,i,i)=1.  ! и диагонали трёхмерного массива  
End Forall
```

Группа FORTRAN

FORTRAN-95 (1997) – FORTRAN 2008 (2010)

```
module class_Circle
  implicit none
  private
  public :: Circle, circle_area, circle_print

  real :: pi = 3.1415926535897931d0 ! Class-wide private constant

  type Circle
    real :: radius
  end type Circle
contains
  function circle_area(this) result(area)
    type(Circle), intent(in) :: this
    real :: area
    area = pi * this%radius**2
  end function circle_area

  subroutine circle_print(this)
    type(Circle), intent(in) :: this
    real :: area
    area = circle_area(this) ! Call the circle_area function
    print *, 'Circle: r = ', this%radius, ' area = ', area
  end subroutine circle_print
end module class_Circle
```

Группа FORTRAN

BASIC (Бейсик)

```
10 CLS 'Очистка экрана
20 PRINT "Добро пожаловать!" 'Заголовок в первой строке
30 'Цикл, выводящий линию под заголовком, на всю ширину экрана
40 FOR I=1 TO 80
50 PRINT "=";
60 NEXT I
65 'Ввод символьных данных от пользователя (комментарий добавлен после ввода нижних строк)
70 INPUT "Имя: ",N$
80 INPUT "фамилия: ",S$
90 INPUT "Отчество: ",T$
95 'Вырезаем копию первых символов из имени и отчества
100 N2$=LEFT$(N$,1)
110 T2$=LEFT$(T$,1)
120 'Выводим результат
130 PRINT "Ваше имя кратко: ";S$;" ";N2$;" ";T2$;"."
140 INPUT "Повторить программу? (Y/N) ",U$
150 IF U$="Y" THEN GOTO 10
160 END
```

Группа FORTRAN

VISUAL BASIC

```
Dim strMessage As String 'объявление переменной символьного типа
Dim iProductCount As Integer 'объявление переменной целочисленного типа
Dim dt70YearsOfVictory As Date 'объявление переменной для хранения даты и времени
Dim otherValue ' тип не указан, переменная типа Variant
```

```
Public Sub Main()
    strMessage = "Привет, Мир!"
    iProductCount = 125
    dt70YearsOfVictory = #5/9/2011#

    otherValue = 12.5 ' тип переменной
    otherValue = "Википедия" ' тип переменной

End Sub
```

```
'Класс ITransport.cls
Public Function GetMaxSpeed() As Long
End Function
```

```
'Класс CAuto.cls
Implements ITransport 'Наследуем интерфейс

Private Function ITransport_GetMaxSpeed() As Long 'Реализация метода
    ITransport_GetMaxSpeed = 240
End Function
```

```
'Класс CFly.cls
Implements ITransport

Private Function ITransport_GetMaxSpeed() As Long
    ITransport_GetMaxSpeed = 700
End Function
```

Группа FORTRAN

LOTUS SCRIPT

```
If Instr(LicolF1Item.Text, "удостоверение") > 1 Then
stroka = Fulltrim(LicolF1Item.Text)
stroka = " " + stroka
Call docGSS.ReplaceItemValue("LicolF1_1",stroka)
LicolF1_1$=""
While stroka <> ""
pos = Instr(stroka,"удостоверение")
If pos <> 0 Then
fio = Left (stroka,pos-1)
fio = Strrightback(fio," ",1,4)
LicolF1_1$ = LicolF1_1$ + Cstr(fio) + " // "
stroka = Right(stroka,Len(stroka)-pos-1)
Else
stroka = ""
End If
Wend
End If
```

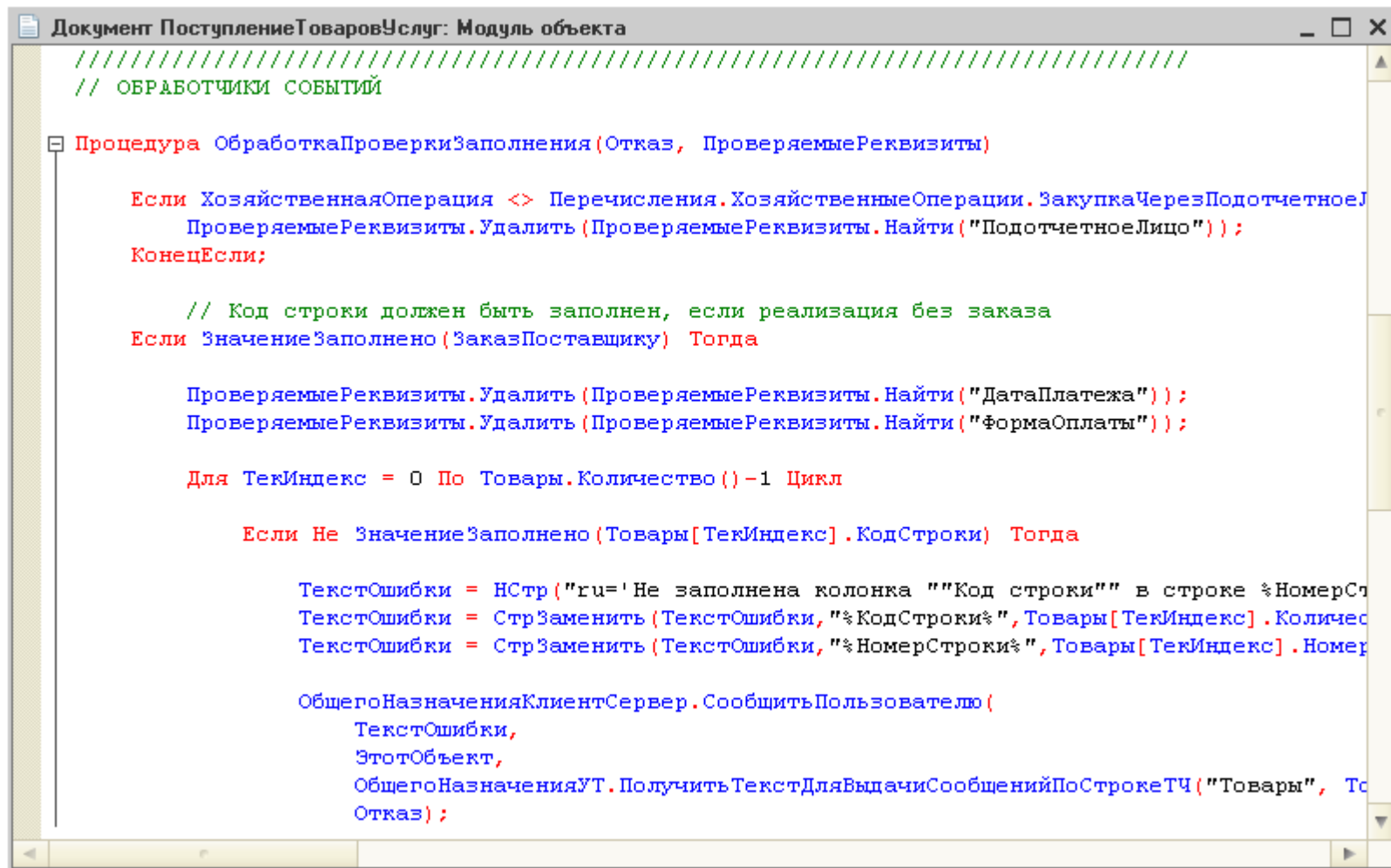
```
Sub Click(Source As Button)
Dim session As New NotesSession
Dim db As NotesDatabase
Dim dc As NotesDocumentCollection
Set db = session.CurrentDatabase
Dim MUNdb As NotesDataBase
Dim doc As NotesDocument
Dim docGSS As NotesDocument
Set MUNdb = session.Getdatabase("B/i", "O\r\G.nsf")
Dim formula As String
formula$ = {Form = "(form3_2ipoteka)"}
Set dc=MUNdb.Search(formula$,Nothing,0)

For i=1 To dc.count
Set doc = dc.GetNthDocument(i)
Set docGSS = doc.CopyToDatabase(db)
Next

End Sub
```


Группа FORTRAN

Встроенный язык 1С



```
Документ ПоступлениеТоваровУслуг: Модуль объекта
////////////////////////////////////
// ОБРАБОТЧИКИ СОБЫТИЙ

□ Процедура ОбработкаПроверкиЗаполнения(Отказ, ПроверяемыеРеквизиты)

    Если ХозяйственнаяОперация <> Перечисления.ХозяйственныеОперации.ЗакупкаЧерезПодотчетноеЛ
        ПроверяемыеРеквизиты.Удалить(ПроверяемыеРеквизиты.Найти("ПодотчетноеЛицо"));
    КонецЕсли;

    // Код строки должен быть заполнен, если реализация без заказа
    Если ЗначениеЗаполнено(ЗаказПоставщику) Тогда

        ПроверяемыеРеквизиты.Удалить(ПроверяемыеРеквизиты.Найти("ДатаПлатежа"));
        ПроверяемыеРеквизиты.Удалить(ПроверяемыеРеквизиты.Найти("ФормаОплаты"));

        Для ТекИндекс = 0 По Товары.Количество() - 1 Цикл

            Если Не ЗначениеЗаполнено(Товары[ТекИндекс].КодСтроки) Тогда

                ТекстОшибки = НСтр("ru='Не заполнена колонка ""Код строки"" в строке %НомерСт
                ТекстОшибки = СтрЗаменить(ТекстОшибки, "%КодСтроки%", Товары[ТекИндекс].Количес
                ТекстОшибки = СтрЗаменить(ТекстОшибки, "%НомерСтроки%", Товары[ТекИндекс].Номер

                ОбщегоНазначенияКлиентСервер.СообщитьПользователю(
                    ТекстОшибки,
                    ЭтотОбъект,
                    ОбщегоНазначенияУТ.ПолучитьТекстДляВыводаСообщенийПоСтрокеТЧ("Товары", То
                    Отказ);
```

Группа FORTRAN

Встроенный язык 1C

```
Result = Query.Execute();
If Not Result.IsEmpty() Then
    Message("Insufficient goods:");
    Selection = Result.Choose();
    While Selection.Next() Do
        Message(Selection.Presentation + ": required: " + Row(Selection.Quantity)
            + ", available: " + Row(Selection.QuantityBalance));
    EndDo;
EndIf;
```

```
SELECT
CASE
    WHEN ItemsAccounting.Recorder REFS Document.SalesOrder
    THEN SalesOrder.Number
    WHEN ItemsAccounting.Recorder REFS Document.PurchaseOrder
    THEN PurchaseOrder.Number
END,
ItemsAccounting.Count
FROM
    AccumulationRegister.ItemsAccounting AS ItemsAccounting
    LEFT JOIN Document.SalesOrder AS SalesOrder
    ON ItemsAccounting.Recorder = SalesOrder.Ref
    LEFT JOIN Document.PurchaseOrder AS PurchaseOrder
    ON ItemsAccounting.Recorder = PurchaseOrder.Ref
```

Группа ALGOL

ALGOL (Алгол) 60 – 68

```
procedure Absmax(a) Size:(n, m) Result:(y) Subscripts:(i, k);
  value n, m; array a; integer n, m, i, k; real y;
  comment Наибольший элемент матрицы a, размера n на m
  передаётся в виде результата в y, а его индексы – в параметры i и k;
begin integer p, q;
  y := 0; i := k := 1;
  for p:=1 step 1 until n do
    for q:=1 step 1 until m do
      if abs(a[p, q]) > y then
        y := abs(a[p, q]); i := p; k := q;
    end
  end
end Absmax
```

```
# Next day date - english variant
mode date = struct(Int day, string month, Int year);
proc the day following = (date x) date:
  If day of
  then (day of month + 1, month, year)
  elif month = "December"
  then (1, "January", year + 1)
  else (1, "January", year)
  fi;
```

```
int initial foot width = 5;
mode foot = struct(
  string name,
  sema width,
  bits toe & packed vector of BOOL &
);

foot left foot:= foot("Left", level initial foot width, 2r11111),
  right foot:= foot("Right", level initial foot width, 2r11111);

& 10 round clip in a 1968 Colt Python .357 Magnum &
sema rounds = level 10;

& the Magnum needs more barrels to take full advantage of parallelism &
sema acquire target = level 1;

prio ^:= = 1;
op ^:= = (ref bits lhs, bits rhs)ref bits: lhs := lhs ^ rhs;
```

Группа ALGOL

PASCAL (Паскаль)

```
program Printing;  
  
var i : integer;  
  
procedure Print(j : integer);  
begin  
    ...  
end;  
  
begin { main program }  
    ...  
    Print(i);  
end.
```

```
unit UnitName2;  
interface  
    ...  
  
implementation  
    ...  
  
initialization  
    ...  
  
finalization  
    ....  
  
end.
```

```
while a <> b do WriteLn('Waiting');  
  
if a > b then WriteLn('Condition met') {no semicolon allowed!}  
    else WriteLn('Condition not met');  
  
for i := 1 to 10 do {no semicolon for single statements allowed!}  
    WriteLn('Iteration: ', i);  
  
repeat  
    a := a + 1  
until a = 10;  
  
case i of  
    0 : Write('zero');  
    1 : Write('one');  
    2 : Write('two');  
    3,4,5,6,7,8,9,10: Write('?')  
end;
```

Группа ALGOL

DELPHI (Дельфи)

```
program Project32;  
  
{$APPTYPE CONSOLE}  
  
{$R *.res}  
  
uses  
    System.SysUtils;
```

```
begin  
    try  
        { TODO -oUse  
    except  
        on E: Except  
            Writeln (E  
    end;  
end.
```

```
type  
    TAnimal = class abstract  
    protected  
        FPersonalName: string;  
    public  
        constructor Create(const PersonalName: string); virtual; abstract;  
        function GetSpecieName: string; virtual; abstract; //возвращает биологический вид живот  
        property Name: string read FPersonalName;  
    end;  
  
    TAnimalClass = class of TAnimal; //метакласс, могущий ссылаться на любой класс, унаследов  
  
    ...  
  
    function CreateAnAnimal(const FactAnimalClass: TAnimalClass; const Name: string): TAnimal;  
    begin  
        Result := FactAnimalClass.Create(Name); //функция не знает, животное какого именно вида б  
        Конкретная реализация вида скрыта.  
    end;
```

Группа ALGOL

ЭЛЬ-76

```
программа
  процедура печатьэл = проц(ф32 эл) начало
    ф32 к;
    для к от 1 до эл[0] цикл печать(эл[к]) повторить
  конец;
  процедура печатьсп = проц(ф32 список) начало
    если не естьпусто список то
      печатьэл(список[0]);
      печатькс();
      печатьсп(список[1])
    все
  конец;
  процедура добавсп = проц(ф32 список, знач) начало
    если не естьпусто список[1] то
      добавсп(список[1], знач)
    иначе
      ф32 с;
      список[1] := лок вект [2] ф32;
      с := список[1];
      с[0] := знач
    все
  конец;
  процедура вариант1 = проц(ф32 список) начало
    если не естьпусто список то
```

Группа ALGOL

ADA (Ада)

```
-- Sample of the parallel calculations.
-- Distributed under the GNU General Public License
-- Author: Dmitriy Anisimkov.

with Ada.Numerics.Generic_Elementary_Functions;

function SSqrt
  (Vector : in Vector;
   Tasks : in Positive)
is
  package GEL is
    -- Generic elementary functions.
  end package GEL;

  function Square_Sum (First, Last : in Positive) return Number;

  task type Calculator is
    entry Start;
    -- First and Last parameters.

    entry Stop;
    -- Get the result.

  end;

  task body Calculator is
    First, Last : Positive;
    Sum : Number;

  begin
    accept Start (First, Last : Positive) do
      -- Copy parameters to the local variables for be able start
      -- calculation after rendezvous.

      Calculator.First := First;
      Calculator.Last := Last;

    end Start;

    Sum := Square_Sum (First, Last);

    accept Stop;
    -- Return the result.

    Result := Sum;
  end Stop;
end Calculator;

function Square_Sum (First, Last : in Positive) return Number is
  Sum : Number := 0.0;

begin
  for J in First .. Last loop
    Sum := Sum + Vector (J) ** 2;
  end loop;

  return Sum;
end Square_Sum;

Sub_Length : constant Positive := Vector'Length / Tasks;

Calcs : array (1 .. Tasks - 1) of Calculator;
-- Tasks - 1 is to rest some elements for the calculation in the main task.

Index : Positive := Vector'First;
Next : Positive;
Sum : Number;
Subsum : Number;
```

Группа ALGOL

PL/SQL

```
DECLARE
-- Описания блока, переменные, типы, курсоры и т. п. (опционально)
BEGIN
-- Непосредственно код программы
EXCEPTION
-- Обработка исключений (с)
END;
/* Многострочные
комментарии... */
-- Однострочный комментарий
```

```
DECLARE
    TYPE GenericCursor IS REF CURSOR;
    c1 GenericCursor;
    empname VARCHAR2(200);
BEGIN
    OPEN c1 FOR SELECT ename
    LOOP
        FETCH c1 INTO empname;
        EXIT WHEN c1%NOTFOUND;
        -- работа со строками
    END LOOP;
    CLOSE c1;
END;
```

```
DECLARE
    TYPE GenericCursor IS REF CURSOR;
    c1 GenericCursor;
    sel VARCHAR2(4000);
    bind_var VARCHAR2(200);
    result VARCHAR2(200);
BEGIN
    sel := 'SELECT name FROM employees WHERE 1 = 1';
    IF ... THEN
        sel := sel || ' AND id = :1';
        bind_var := 12;
    ELSE
        sel := sel || ' AND city = :1';
        bind_var := 'Магадан';
    END IF;

    OPEN c1 FOR sel USING bind_var;
        FETCH c1 INTO result;
    CLOSE c1;
END;
```


Группа ALGOL

FORTH (Форт)

```
\ Напечатать знак числа
: .SIGN ( n -- )
  ?DUP 0= IF
    ." НОЛЬ"
  ELSE
    0> IF
      ." ПОЛОЖИТЕЛЬНОЕ ЧИСЛО" ELSE
      ." ОТРИЦАТЕЛЬНОЕ ЧИСЛО" THEN
    THEN
;

```

```
\ Создать "константу" из строки
: S-CONSTANT ( c-addr u "<spaces>name" -- )
  CREATE
  DUP , 0 ?DO
    DUP C@ C, CHAR+
  LOOP DROP 0 C,
  DOES>
  DUP CELL+ SWAP @
;

```

```
\ Некоторый размер данных в килобайтах
16 CONSTANT size

\ Напечатать отчёт о пересчёте килобайтов в байты
: report ( -- )
  size . ." килобайт эквивалентны "
  [ size 1024 * ] LITERAL . ." байтам"
;

```

Группа ALGOL

POSTSCRIPT

```
%!PS-Adobe-2.0
%%Creator: dvips(k) 5.78 Copyright 1998 Radical Eye Software
%%Title: rep.dvi
%%Pages: 9%%PageOrder: Ascend
%%BoundingBox: 0 0 596 842
%%EndComments
```

```
% По умолчанию, постскрипт использует единицу измерения 1 пункт=1/72 дюйма,
% а нам нравятся миллиметры.
72 25.4 div          % 1 мм = 72/25.4 пунктов
dup                  % дублировать значение на вершине стека
scale                 % растянуть в это количество раз по обеим координатам
100 100 translate    % установить начало координат в (100мм, 100мм)

/Times-Roman findfont % взять шрифт Times-Roman
10 scalefont          % растянуть до размера 10 (у нас - мм!)
setfont                % установить выбранный шрифт

0 30 330 {            % цикл для углов от 0 до 330 с шагом 30
  gsave                % запомнить текущее преобразование координат
    rotate              % повернуть систему координат (угол в градусах берём с вершины стека)
    15 0 moveto          % перейти в точку (15мм, 0мм)
    (Wikipedia) show     % написать слово текущим шрифтом
  grestore              % вернуть преобразование координат
} for                  % конец цикла for

showpage               % вывести страницу
```

Группа С

С (Си)

```
#include <stdio.h>
#include <stdlib.h>

void main(void)
{
    int *p, i;
    p = (int *) malloc(100 * sizeof(int));

    if (!p)
    {
        printf("Недостаточно памяти\n");
        exit(1);
    }
    for (i = 0; i < 100; ++i) *(p+i) = i;
    for (i = 0; i < 100; ++i) printf("%d",
        free(p);
}
```

```
#include <stdio.h>
#include <values.h>
#include <process.h>

long factorial(int value)          /* Рекурсивная функция */
{
    long result = 1;

    if (value != 0)
    {
        result = factorial(value - 1);
        /* Проверка возможности вычисления факториала */
        if (result > MAXLONG / (value + 1))
        {
            fprintf(stderr, "Очень большое число\n");
            getch( );              /* Ожидание нажатия клавиши */
            exit (1);
        }
        result *= value;
    }
    return(result);
}

/* Рекурсивное вычисление факториала числа value */
void main(void)
{
    int value;                     /* факториал этого значения вычисляется */
    long result;                   /* Переменная для результата */

    puts("факториал какого числа?");
    scanf("%d", &value);
    result = factorial(value);
    printf("Результат: %ld\n", result);
    getch( );                      /* Ожидание нажатия клавиши */
}
```

Группа С

C++

```
namespace Foo
```

```
{
    const int x=5;
    typedef int** T;
    void f(int y) {return y*x};
    double g(T);
    ...
}
```

```
void Print(int x);
void Print(double x);
void Print(int x, int y);
```

```
class Matrix {
    ...
    friend Matrix Multiply(Matrix m1, Matrix m2);
    ...
};

Matrix Multiply(Matrix m1, Matrix m2) {
    ...
}
```

```
class Figure
{
    ...
    virtual void Draw() const;
    ...
};
```

```
class Square : public Figure
```

```
...
void Draw() const;
```

```
Figure* figures[10];
figures[0] = new Square(1, 2, 1);
figures[1] = new Circle(3, 5, 8);
...
for (int i = 0; i < 10; i++)
    figures[i]->Draw();
```

```
#include <iostream> // для использования std::
#include <vector> // для std::vector<>
#include <map> // для std::map<> и std::pair<>
#include <algorithm> // для std::for_each()
#include <string> // для std::string
```

```
using namespace std; // используем пространство имен std
```

```
void display_item_count(pair < string const, int> item) const
{
    // person - это пара двух объектов: person.first - это имя человека,
    // person.second - это список его предметов
    cout << person.first << " is carrying " << item.second << endl;
}
```

```
int main()
{
    // Объявляем ассоциативный контейнер со std::map
    map< string, vector<string> > items;

    // Добавим в этот ассоциативный контейнер несколько предметов
    items["Anya"].push_back("scarf");
    items["Dmitry"].push_back("tickets");
    items["Anya"].push_back("puppy");

    // Переберём все объекты в контейнере
    for_each(items.begin(), items.end(), display_item_count);
}
```

Группа С

JAVA (Ява)

```
import java.util.Scanner;

class Point {
    public double x; // абсцисса точки
    public double y; // ордината точки

    // возвращает строку с описанием точки
    public String toString() {
        return "("+x+";"+y+";";
    }
    // выводит на экран описание точки
    public void print() {
        System.out.print(this.toString());
    }
    // метод перемещает точку на указанный вектор
    public void move(double a, double b) {
        x = x + a;
        y = y + b;
    }
    // метод изменяет координаты точки на указанные
    public void set(double a, double b) {
        x = a;
        y = b;
    }
}
```

```
public Point() {
    boolean err;
    do {
        err = false;
        System.out.print("Введите абсциссу точки: ");
        Scanner scan = new Scanner(System.in);
        if(scan.hasNextDouble()) {
            x = scan.nextDouble();
        } else {
            System.out.println("Вы ввели не число, попробуйте снова");
            err = true;
        }
    } while (err);
    do {
        err = false;
        Scanner scan = new Scanner(System.in);
        System.out.print("Введите ординату точки: ");
        if(scan.hasNextDouble()) {
            y = scan.nextDouble();
        } else {
            System.out.println("Вы ввели не число, попробуйте снова");
            err = true;
        }
    } while (err);
}
// конструктор, создающий точку с указанными координатами
public Point(double a, double b) {
    x = a;
    y = b;
}
```

Группа С

С# (Си шарп)

```
using System;

namespace Example
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Hello World");
            Console.ReadKey(); // Ожидание нажатия клавиши
        }
    }
}
```

```
using System;
using System.Drawing;
using System.Windows.Forms;

namespace WindowsForms
{
    public class Program
    {
        [STAThread]
        public static void Main()
        {
            new DemoForm().ShowDialog();
        }

        public class DemoForm : Form
        {
            Label label = new Label();

            public DemoForm()
            {
                label.Text = "Hello World!";
                this.Controls.Add(label);
                this.StartPosition = FormStartPosition.CenterScreen;
                this.BackColor = Color.White;
                this.FormBorderStyle = FormBorderStyle.Fixed3D;
            }
        }
    }
}
```

Группа С

JavaScript (JS)

```
window.onload = function() {  
    var linkWithAlert = document.getElementById("alertLink");  
    linkWithAlert.onclick = function() {  
        return confirm('Вы уверены?');  
    };  
};
```

```
var fruits, text, fLen, i;  
  
fruits = ["Banana", "Orange", "Apple", "Mango"];  
fLen = fruits.length;  
text = "<ul>";  
for (i = 0; i < fLen; i++) {  
    text += "<li>" + fruits[i] + "</li>";  
}
```

```
var obj;  
obj = document.getElementById("demo");  
obj.innerHTML = "Hello";
```

```
xhttp.open("GET", "ajax_info.txt", false);  
xhttp.send();  
document.getElementById("demo").innerHTML = xhttp.responseText;
```

```
<script>  
window.onload = downScripts;  
  
function downScripts() {  
    var element = document.createElement("script");  
    element.src = "myScript.js";  
    document.body.appendChild(element);  
}  
</script>
```

Группа С

JQuery

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js">
</script>
<script src="my_jquery_functions.js"></script>
</head>
```

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("[href]").hide();
});
</script>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<a href="http://www.w3schools.com/html/">HTML Tutorial</a>
<a href="http://www.w3schools.com/css/">CSS Tutorial</a>

</body>
```

```
$("#button").click(function(){
    $("h1, h2, p").addClass("blue");
    $("div").addClass("important");
});
```

```
$("#button").click(function(){
    $.get("demo_test.asp", function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```


Группа С

PHP

```
<!DOCTYPE html>
<html>
<body>
```

```
<?php
$color = "red";
echo "My car is " . $color;
echo "My house is " . $color;
echo "My boat is " . $color;
?>
```

```
</body>
</html>
```

```
<?php
// Start the session
session_start();
?>
```

```
<!DOCTYPE html>
<html>
<body>
```

```
<?php
// Set session variables
$_SESSION["favcolor"] = "red";
$_SESSION["favourite"] = "cat";
echo "Session variables: " . print_r($_SESSION, true);
?>
```

```
</body>
</html>
```

```
<?php
//error handler function
function customError($errno, $errstr) {
    echo "<b>Error:</b> [$errno] $errstr<br>";
    echo "Ending Script";
    die();
}

//set error handler
set_error_handler("customError",E_USER_WARNING);

//trigger error
$test=2;
if ($test>=1) {
    trigger_error("Value must be 1 or below",E_USER_WARNING);
}
?>
```

Группа С

PERL

```
my $file = 'file.txt'; #имя файла, который читаем
open FH, $file; #открываем файл
my $text = _parseFile(*FH); #передаем файловый дескриптор
print $text;
close FH; #обязательно закрываем файл
sub _parseFile
{
    my $fh = shift; #получаем файловый дескриптор
    my $text = join('', <$fh>); #получаем текст
    return $text;
}
```

```
package LSystem;
use warnings;
use base ('Turtle');

sub new($%)
{
    my $class=shift;
    my $self=Turtle->new;
    $self->{condition}=shift;
    %{$self->{rules}}=@_;
    return bless $self, $class;
}

sub iterate($)
{
    my $self=shift;
    my $n=shift;
    $self->{condition}=join
        (
            '',
            map $self->{rules}{$_}//$_,
              split(//, $self->{condition})
        )
        while $n--;
```

Группа С

Python (Питон)

```
# для версии меньше 3
"строка"
'строка'
"""тоже строка"""
u"Юникод-строка"
True or False # булевы ли
3.14 # число с плавающей
0o12 + 0xA # числа в вось
1 + 2j # комплексное числ
[1, 2, "a"] # список
(1, 2, "a") # кортеж
{'a': 1, 'b': 'B'} # слов
{'a', 6, 8.8} # множество
lambda x: x**2 # анонимна
```

```
class A(object):
    def __init__(self, x):
        self._x = x

    def getx(self):
        return self._x

    def setx(self, value):
        self._x = value

    def delx(self):
        del self._x

    x = property(getx, setx, de

a = A(5)
print a.x # Синтаксис дост
a.x = 5
```

```
class Multiplyable:
    def __init__(self, value):
        self.value = value

    def __mul__(self, y):
        return self.value * y

    def __rmul__(self, x):
        return x * self.value

    def __imul__(self, y):
        return Multiplyable(self.value * y)

    def __str__(self):
        return "Multiplyable(%s)" % self.value

>>> m = Multiplyable(1)
>>> print m
Multiplyable(1)
>>> m *= 3
>>> print m
Multiplyable(3)
```

Группа С

R

```
# В переменную a поместить список всех оценок:  
a <- c(4,3,3,3,3,4,4,4,4,4,5,4,4,4,5,5,5,5,+  
3,5,5,4,4,3,3,4,4,3,5,5,4,3,3,4,4,3,3,5,4,5,5)
```

```
# В переменную n поместить количество оценок:  
length(a) -> n
```

```
# Средний балл:  
m <- mean(a)
```

```
# Таблица (горизонтальная) с подсчётом количества оценок  
t <- table(a)
```

```
# Преобразование в более удобный формат (в матрицу)  
f <- as.data.frame(t)
```

```
# Вычисление процентной доли и запись в матрицу  
mapply(function(r) r*100/n, f[,2])
```

```
# Заголовки столбцов:  
colnames(f) <- c("Оценка", "Кол-во")
```

```
# Вывод результатов:
```

```
a  
n  
m  
f
```

```
n <- 1000  
x <- sample(c(0,1), n, replace=T)  
x <- sample(c(0,1), n, replace=T, prob=c(.3,.7))  
x <- runif(n) >.3  
x <- rbinom(n, 1, .2)
```

```
N <- 10000  
x <- rpois(N, 1)  
x <- rpois(N, 3)  
x <- rpois(N, 100)
```

```
require(graphics)  
## Annette Dobson (1990) "An Introduction to Generalized Linear Models".  
## Page 9: Plant Weight Data.  
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)  
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)  
group <- gl(2,10,20, labels=c("Ctl","Trt"))  
weight <- c(ctl, trt)  
lm.D9 <- lm(weight ~ group)  
lm.D90 <- lm(weight ~ group - 1) # omitting intercept  
  
anova(lm.D9)  
summary(lm.D90)  
  
opar <- par(mfrow = c(2,2), oma = c(0, 0, 1.1, 0))  
plot(lm.D9, las = 1) # Residuals, Fitted, ...  
par(opar)
```

Определение данных

SQL, T-SQL, PL/SQL

```
CREATE TABLE table_name  
(  
  column_name1 data_type(size),  
  column_name2 data_type(size),  
  column_name3 data_type(size),  
  ....  
);
```

```
CREATE OR REPLACE VIEW view_name AS  
SELECT column_name(s)  
FROM table_name  
WHERE condition
```

```
CREATE ASSEMBLY utf8string  
AUTHORIZATION [dbo]  
FROM 0x4D... ;  
GO  
CREATE TYPE Utf8String  
EXTERNAL NAME utf8string.[Microsoft.Samples.SqlServer.utf8string] ;  
GO
```

```
CREATE TYPE address_typ AS OBJECT (  
  street          VARCHAR2(30),  
  city            VARCHAR2(20),  
  state           CHAR(2),  
  postal_code     VARCHAR2(6) );  
/  
CREATE TYPE employee_typ AS OBJECT (  
  employee_id     NUMBER(6),  
  first_name      VARCHAR2(20),  
  last_name       VARCHAR2(25),  
  email           VARCHAR2(25),  
  phone_number    VARCHAR2(20),  
  hire_date       DATE,  
  job_id          VARCHAR2(10),  
  salary          NUMBER(8,2),  
  commission_pct  NUMBER(2,2),  
  manager_id      NUMBER(6),  
  department_id   NUMBER(4),  
  address         address_typ,  
  MAP MEMBER FUNCTION get_idno RETURN NUMBER,  
  MEMBER PROCEDURE display_address ( SELF IN  
/  

```

Определение данных

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="country">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="country_name" type="xs:string"/>
        <xs:element name="population" type="xs:decimal"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<grammar xmlns="http://relaxng.org/ns/structure/1.0">
  <start>
    <element name="book">
      <oneOrMore>
        <ref name="page"/>
      </oneOrMore>
    </element>
  </start>
  <define name="page">
    <element name="page">
      <text/>
    </element>
  </define>
</grammar>
```

Определение данных

OWL

```
<owl:Class rdf:ID="Винодельня"/>
<owl:Class rdf:ID="Регион"/>
<owl:Class rdf:ID="ПродуктПитания"/>
```

```
<owl:Class rdf:ID="Вино">
  <rdfs:subClassOf rdf:resource="&food;Напиток"/>
  <rdfs:label xml:lang="en">wine</rdfs:label>
  <rdfs:label xml:lang="ru">вино</rdfs:label>
  <rdfs:label xml:lang="fr">vin</rdfs:label>
  ...
</owl:Class>

<owl:Class rdf:ID="Паста">
  <rdfs:subClassOf rdf:resource="#СъедобнаяВещь" />
  ...
</owl:Class>
```

```
<owl:ObjectProperty rdf:ID="расположенВ">
  <rdf:type rdf:resource="&owl;TransitiveProperty" />
  <rdfs:domain rdf:resource="&owl;Thing" />
  <rdfs:range rdf:resource="#Регион" />
</owl:ObjectProperty>
```

```
<Регион rdf:ID="РегионГорыСантаКруз">
  <расположенВ rdf:resource="#РегионКалифорния" />
</Регион>
```

```
<Регион rdf:ID="РегионКалифорния">
  <расположенВ rdf:resource="#РегионКалифорния" />
</Регион>
```

```
<owl:Class rdf:about="#Бургундское">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Вино" />
    <owl:Restriction>
      <owl:onProperty rdf:resource="#расположенВ" />
      <owl:hasValue rdf:resource="#РегионБургонь" />
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Разметка

HTML (HyperText Markup Language)

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>HTML Document</title>
  </head>
  <body>
    <p>
      <b>
        Этот текст будет полужирным,
        <i>а этот - ещё и курсивным</i>
      </b>
    </p>
  </body>
</html>
```

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
```

```
<p><a href="http://www.w3schools.com" target="iframe_a">W3Schools.com</a></p>
```


Разметка

HTML5 (HyperText Markup Language 5)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>
      (Это title) Пример страницы на HTML5
    </title>
  </head>
  <body>
    <header>
      <hgroup>
        <h1>
          Заголовок "h1" из hgroup
        </h1>
        <h2>
          Заголовок "h2" из hgroup
        </h2>
      </hgroup>
    </header>
    <nav>
      <menu>
        <li>
```

```
<video poster="movie.jpg" controls>
  <source src='movie.webm' type='video/webm; codecs="vp8.0, vorbis"'/>
  <source src='movie.ogv' type='video/ogg; codecs="theora, vorbis"'/>
  <source src='movie.mp4' type='video/mp4; codecs="avc1.4D401E, mp4a.40.2"'/>
  <p>This is fallback content</p>
</video>
```

Разметка

XML

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE recipe>
<recipe name="хлеб" preptime="5min" cooktime="180min">
  <title>
    Простой хлеб
  </title>
  <composition>
    <ingredient amount="3" unit="стакан">Мука</ingredient>
    <ingredient amount="0.25" unit="грамм">Дрожжи</ingredient>
    <ingredient amount="1.5" unit="стакан">Тёплая вода</ingredient>
  </composition>
  <instructions>
    <step>
      Смешать все ингредиенты и тщательно замесить.
    </step>
    <step>
      Закрыть тканью и оставить на один час в тёплом помещении.
    </step>
  </instructions>
</recipe>
```

```
<w:document xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main">
  <w:body>
    <w:p>
      <w:r>
        <w:t>Example text.</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

Разметка

CSS (Cascading Style Sheets, каскадные таблицы стилей)

```
p.Big {  
    font-family: arial, helvetica, sans-serif;  
    color: maroon;  
}  
div#First {  
    background-color: silver;  
}
```

```
p {  
    font-family: arial, helvetica, sans-serif;  
}  
h2 {  
    font-size: 20pt;  
    color: red;  
    background: white;  
}  
.note {  
    color: red;  
    background-color: yellow;  
    font-weight: bold;  
}  
p#paragraph1 {  
    padding-left: 10px;  
}  
a:hover {  
    text-decoration: none;  
}  
#news p {  
    color: blue;  
}  
[type="button"] {  
    background-color: green;  
}
```

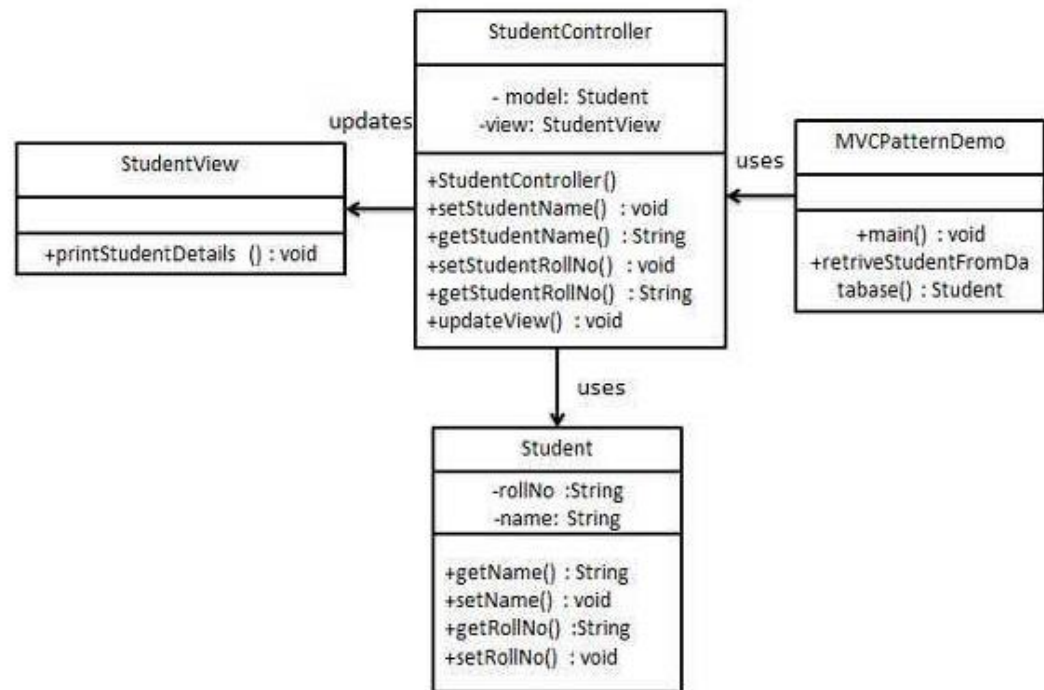
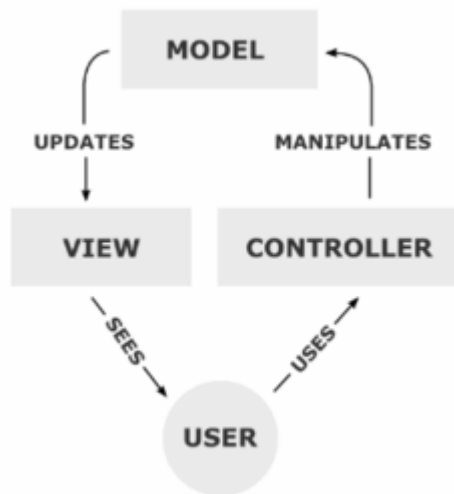
Фреймворк

XMI

```
<?xml version="1.0"?>
<XMI xmi.version="1.2" xmlns:UML="org.omg/UML/1.4">
  <XMI.header>
    <XMI.documentation>
      <XMI.exporter>ananas.org stylesheet</XMI.exporter>
    </XMI.documentation>
    <XMI.metamodel xmi.name="UML" xmi.version="1.4"/>
  </XMI.header>
  <XMI.content>
    <UML:Model xmi.id="M.1" name="address" visibility="public"
      isSpecification="false" isRoot="false"
      isLeaf="false" isAbstract="false">
      <UML:Namespace.ownedElement>
        <UML:Class xmi.id="C.1" name="address" visibility="public"
          isSpecification="false" namespace="M.1" isRoot="true"
          isLeaf="true" isAbstract="false" isActive="false">
          <UML:Classifier.feature>
            <UML:Attribute xmi.id="A.1" name="name" visibility="private"
              isSpecification="false" ownerScope="instance"/>
            <UML:Attribute xmi.id="A.2" name="street" visibility="private"
              isSpecification="false" ownerScope="instance"/>
            <UML:Attribute xmi.id="A.3" name="zip" visibility="private"
              isSpecification="false" ownerScope="instance"/>
            <UML:Attribute xmi.id="A.4" name="region" visibility="private"
              isSpecification="false" ownerScope="instance"/>
            <UML:Attribute xmi.id="A.5" name="city" visibility="private"
              isSpecification="false" ownerScope="instance"/>
          </UML:Classifier.feature>
        </UML:Class>
      </UML:Namespace.ownedElement>
    </UML:Model>
  </XMI.content>
</XMI>
```

Фреймворк

Model-view-controller (MVC, «модель-представление-контроллер», «модель-вид-контроллер») — схема использования нескольких [шаблонов проектирования](#), с помощью которых модель приложения, [пользовательский интерфейс](#) и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения [архитектурного каркаса](#), когда переходят от теории к реализации в конкретной предметной области



Фреймворк

Шаблон проектирования или **паттерн** ([англ. design pattern](#)) в [разработке программного обеспечения](#) — повторяемая [архитектурная конструкция](#), представляющая собой решение проблемы [проектирования](#) в рамках некоторого часто возникающего [контекста](#).

Обычно шаблон не является законченным образцом, который может быть прямо преобразован в [код](#); это лишь пример решения задачи, который можно использовать в различных ситуациях.

[Объектно-ориентированные](#) шаблоны показывают отношения и [взаимодействия](#) между [классами](#) или [объектами](#), без определения того, какие конечные классы или объекты приложения будут использоваться., когда переходят от теории к реализации в конкретной предметной области