

# Математическое и лингвистическое обеспечение САПР

# Математическое обеспечение

- **Математическое обеспечение** – это совокупность математических моделей, методов и алгоритмов решения задач автоматизированного проектирования.
- **Математическое обеспечение** при автоматизированном проектировании в явном виде не используется, а применяется компонент, производный от него – **программное обеспечение**.

# Математическое обеспечение

- **Математическое обеспечение** является самым сложным этапом создания САПР.
- **От успешности** математического обеспечения в наибольше степени **зависит эффективность** использования САПР.

# Состав МО

**Математическое обеспечение делится на 2 части:**

- 1. Математические методы и модели, описывающие объекты проектирования, их свойства и параметры.**
- 2. Формализованное описание технологии автоматизированного проектирования.**

# Математические методы и модели

- Эта часть математического обеспечения наиболее специфична и зависит от предметной области объекта проектирования.

$$\lambda \equiv 4,12 \times \sqrt[3]{\frac{(\alpha + \beta)^2}{\alpha^2 + \beta^2}}$$

$$\Delta x_i = h = \frac{b - a}{n}$$

$$\vec{a} = \frac{F_1 + F_2 + \dots + F_n}{m}$$

$$4(1 + \xi^2) = 3(1 + \xi) + \frac{1}{(1 + \xi)^3}$$

# Формализованное описание технологии

- При решении задач в данной части должна быть формализована вся технология проектирования, в том числе логика взаимодействия проектировщиков друг с другом и со средствами автоматизации.
- Проблемы, которые касаются логики взаимодействия проектировщиков решаются опытным путем.

# Направления совершенствования

**Направления совершенствования  
математического обеспечения:**

- 1. Развитие методов получения оптимальных проектных решений. Это касается первой части математического обеспечения.**
- 2. Совершенствование и стандартизация процессов автоматизированного проектирования. Это касается второй части математического обеспечения.**

# Направления развития

Математическое обеспечение должно обеспечить не просто решение, а **оптимальное решение**.

**Для этого есть несколько направлений развития:**

- 1. Разработка критериев эффективности проектных решений.**
- 2. Выбор наиболее эффективных методов получения проектных решений.**
- 3. Создание Банков Знаний – фондов описания объектов и методов проектирования.**



# Лингвистическое обеспечение

- Лингвистическое обеспечение – это совокупность языков, используемых в процессе проектирования.
- Язык – это совокупность знаков, используемых для общения.



# Классификация языков

**Языки, используемые в САПР делятся на 3 вида:**

1. Языки программирования
2. Языки проектирования
3. Языки управления

# Языки программирования

- **Языки программирования** служат для разработки и редактирования системного и прикладного программного обеспечения САПР.
- **Эти языки базируются на алгоритмических языках.**

# Уровни языков программирования

- Языки программирования делятся на 2 уровня:

1. Языки высокого уровня
2. Языки низкого уровня

- Первые приближены к пользователю

- Вторые приближены к машинным языкам

# Преобразования

- Программа, записанная на языке высокого уровня, называется исходной программой.
- Для исполнения она должна быть преобразована в машинную форму.
- Подобные преобразования выполняются специальными программами, которые называются языковыми процессорами.



# Языковые процессоры

- **Языковые процессоры делятся на 2 типа:**
  1. Трансляторы
  2. Интерпретаторы
- **При трансляции** программа сначала преобразуется с одного языка в другой, а потом выполняется.
- **При интерпретации** перевод исходной программы в рабочую совмещен по времени с её исполнением.

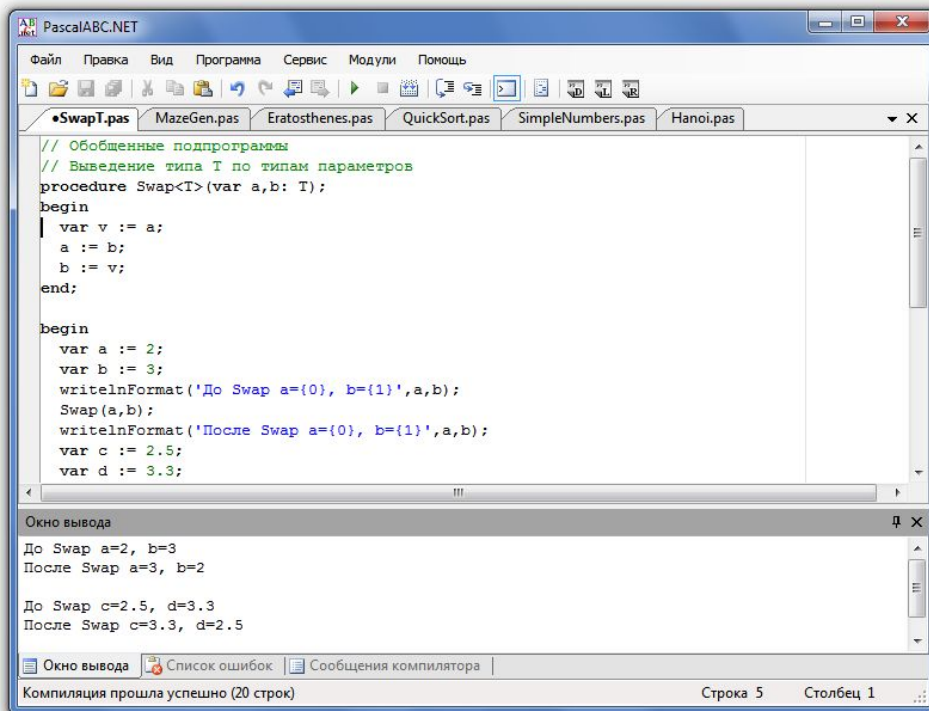
# Виды трансляторов

**Транслятор называется:**

- **Компилятором**, если исходный язык высокого уровня преобразуется в машинный.
- **Ассемблером**, если исходный язык машинно-ориентированный (автокод) преобразуется в машинный.
- **Конвертером**, если преобразуются языки одного уровня.

# Система программирования

- Совокупность языка программирования и соответствующего ему языкового процессора называется системой программирования.



The screenshot shows the PascalABC.NET IDE with a Pascal program in the main editor. The program defines a procedure `Swap` that takes two variables of type `T` and swaps their values. It then calls this procedure with various numerical values and prints the results before and after the swap.

```
// Обобщенные подпрограммы
// Выведение типа T по типам параметров
procedure Swap<T>(var a,b: T);
begin
  var v := a;
  a := b;
  b := v;
end;

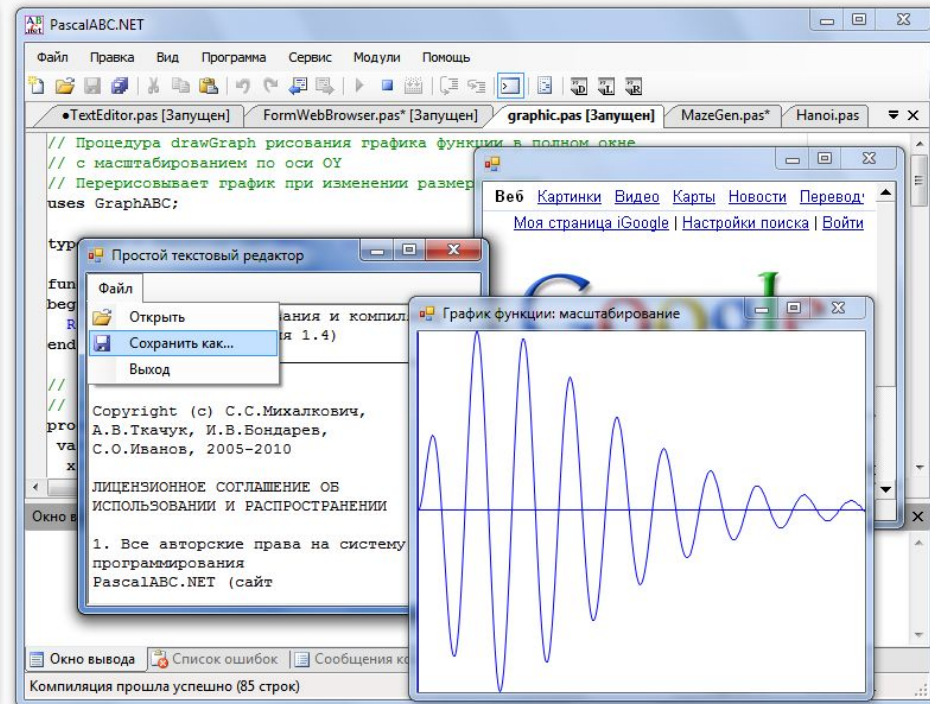
begin
  var a := 2;
  var b := 3;
  writelnFormat('До Swap a={0}, b={1}',a,b);
  Swap(a,b);
  writelnFormat('После Swap a={0}, b={1}',a,b);
  var c := 2.5;
  var d := 3.3;
end;
```

The output window shows the following results:

```
До Swap a=2, b=3
После Swap a=3, b=2

До Swap c=2.5, d=3.3
После Swap c=3.3, d=2.5
```

At the bottom, it indicates: "Компиляция прошла успешно (20 строк) Строка 5 Столбец 1".



The screenshot shows the PascalABC.NET IDE with a Pascal program for drawing a graph. The program uses the `drawGraph` procedure to plot a function. A context menu is overlaid on the editor, and a window titled "График функции: масштабирование" displays a graph of a sine wave.

```
// Процедура drawGraph рисования графика функции в полном окне
// с масштабированием по оси OY
// Перерисовывает график при изменении размеров
uses GraphABC;
```

The context menu shows options: "Открыть...", "Сохранить как...", "Выход".

The graph window displays a sine wave with the title "График функции: масштабирование".

At the bottom, it indicates: "Компиляция прошла успешно (85 строк)".



# Языки проектирования

- Языки проектирования делятся на:
  - Входные
  - Выходные
  - Языки сопровождения
  - Промежуточные
  - Внутренние

# Входные языки

- **Входные языки** служат для задания исходной информации об объекте и цели проектирования.
- **Во входных языках выделяют две части:**
  1. **Процедурную** – описывает цели проектирования
  2. **Непроцедурную** – описывает объекты проектир-я
- **Процедурную часть** описывают языки описания заданий
- **Непроцедурную часть** описывают языки описания объектов

# Языки описания объектов

Языки описания объектов делятся на:

- схемные
- графические
- языки моделирования

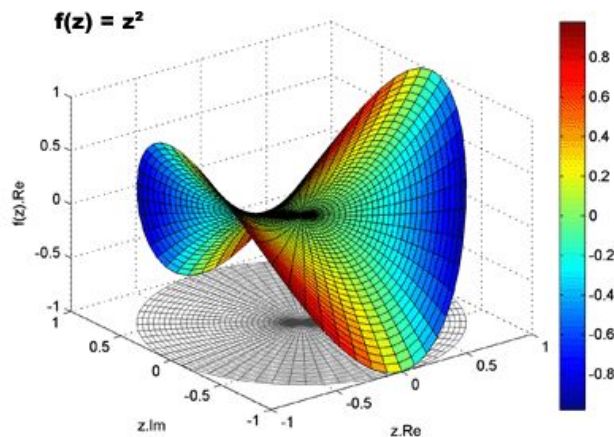
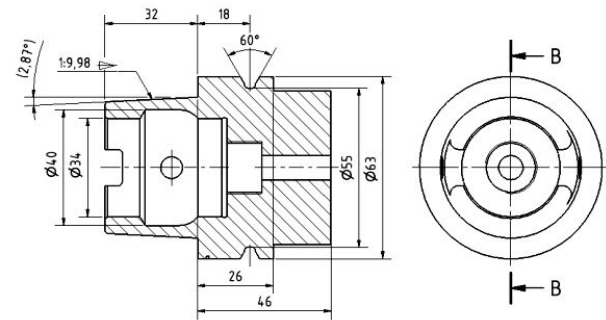
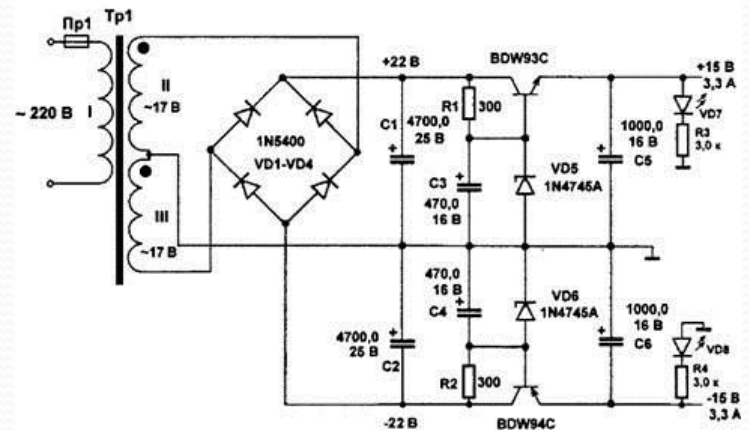
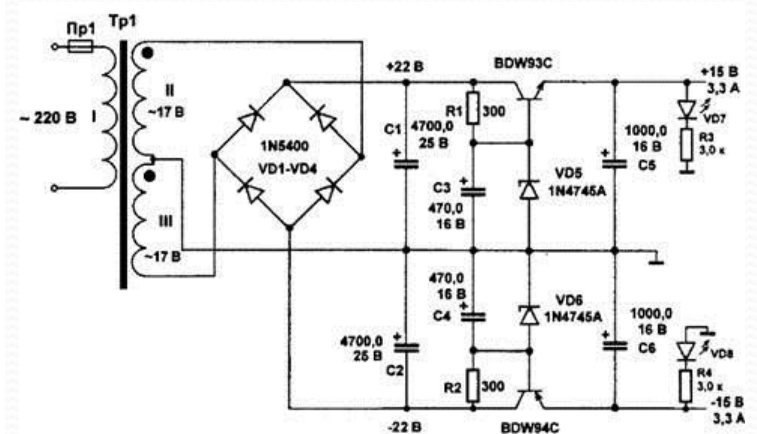
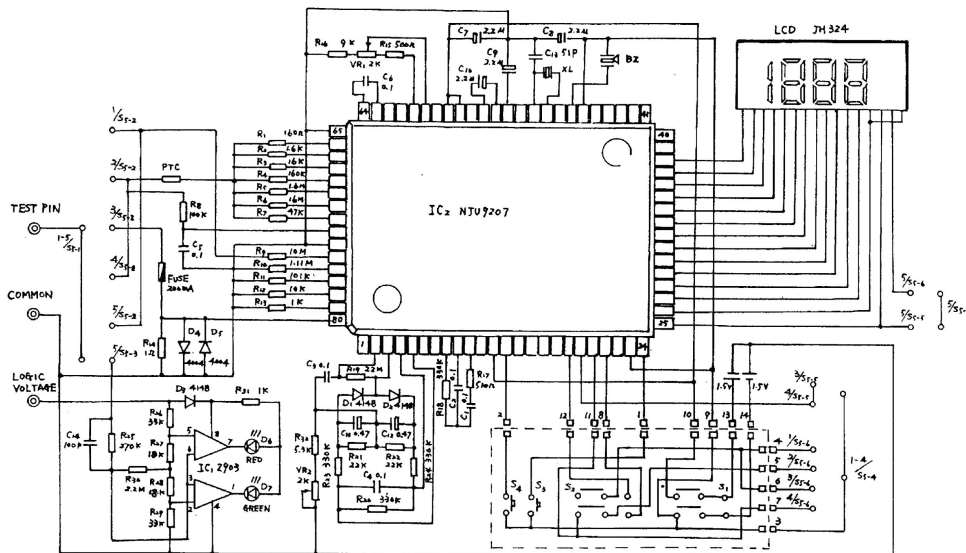


График комплексной функции  
в четырехмерном (4D) пространстве



# Схемные языки

- Схемные языки применяются для описания особенностей объектов проектирования, например, электронных схем.



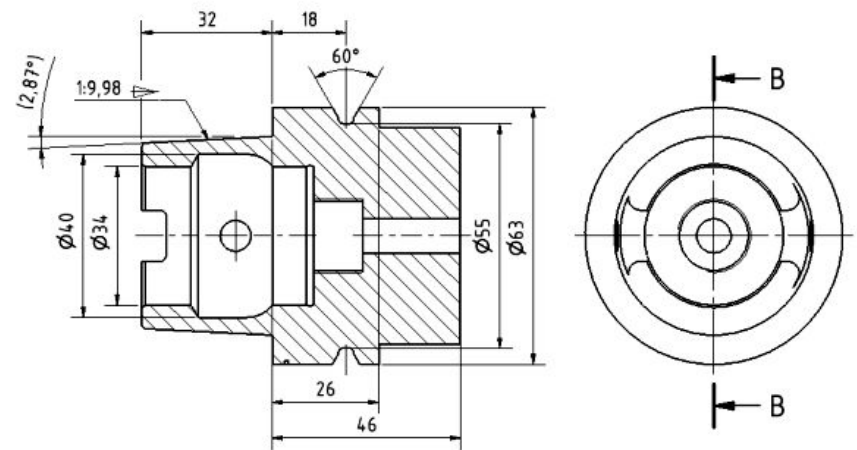
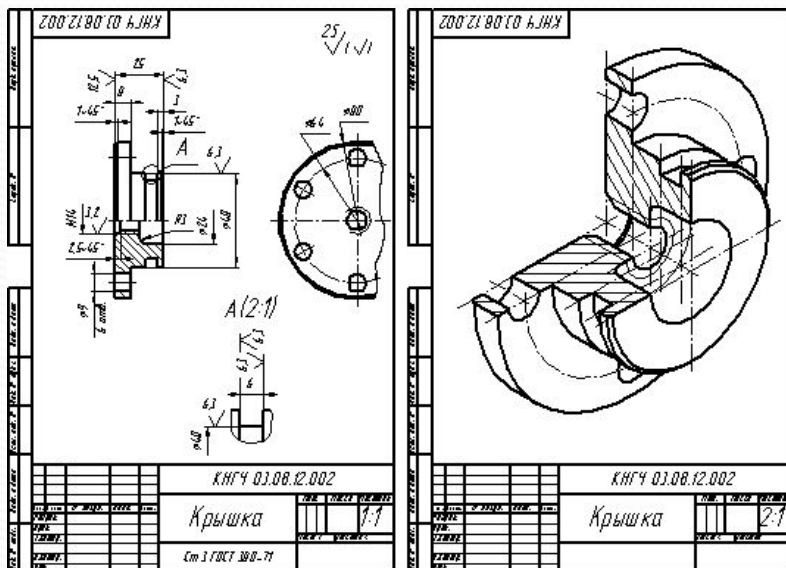
S <sub>F</sub>	1	2	3	4	5
L	DCV	Ω	DCA	⊖	LOGIC
R	ACV	⊕	ACA	⊖	LOGIC

S<sub>1</sub>, POWER  
 S<sub>2</sub>, DC A/AC A  
 S<sub>3</sub>, RANGE  
 S<sub>4</sub>, DATA HOLD  
 S<sub>5</sub>, FUNCTION SELECT

MODEL: 3211D

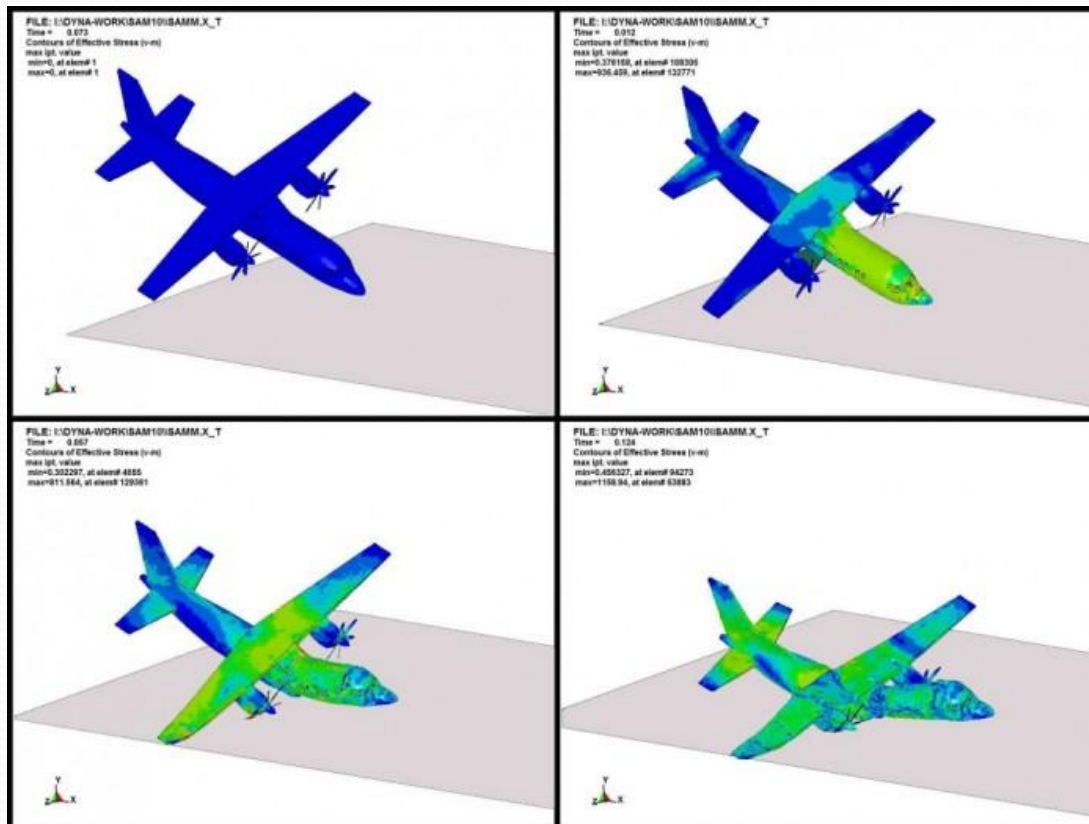
# Графические языки

- Графические языки применяются для описания чертежей, геометрических изображений и т.д.
- Эти языки стандартизованы и могут быть использованы в различных предметных областях.



# Языки моделирования

- Языки моделирования применяются для описания процессов, протекающих в моделируемом объекте.



# Выходные языки

- **Выходные языки** используются для представления **результата** проектирования в удобной для проектировщика форме (чертежи, схемы, таблицы, диаграммы, графики, текст).
- **Выходная информация** должна быть **стандартизована**.

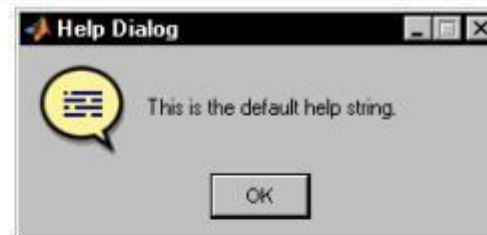
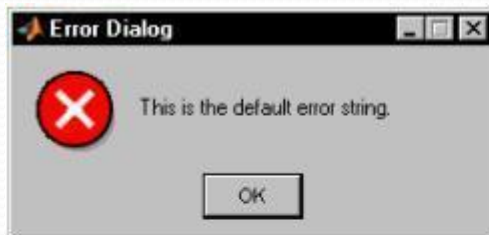


Depth		NDL time																									
m	ft	PADI	NAUI	NOAA	SSI	USN	DCI	EM	RUS	YM	CA	BEES/	VD	FFE	Lifras	NOB	PDI	Bara	Buhl	Com	HU	Min	Max	Av.			
10	33						100							330								100	330	215			
11	36			310						220												165	165	310	232		
12	39	147	130	200	130	200	90	300	150	140	165	135	100	150	195			140		165	135	90	360	161			
13	43																										
14	46	98																					98	98	98		
15	49		80	100	70	100	70	105	80	72	80	75	100	75	90	85	80	75	80	75	60	105	82				
16	52	72																				72	72	72			
17	56																										
18	59	56	55	60	50	60	50	45	50	45	50	50	60	50	55	60			51	50	50	45	60	53			
19	63																					40	45	42			
20	66	45																				35	50	41			
21	69		45		40	50	35	35	40	31	35			50	35	45	40		35	35	40	31	60	39			
22	72	37		50										35								35	50	41			
23	75																										
24	79		35		30	40	25	25	30	23	25			40	25	35	30		25	25	30	23	40	30			
25	82	29		40										20							20	20	40	29			
26	85																										
27	89		25		30		20	20	20	18	20			30	20	25	25		20	20	25	18	30	22			
28	92													15								15	30	23			
29	95																					30	30	30			
30	99	20	22												20	25	15	15	20	20	20	17	15	20	10	25	18
31	102			25																		25	25	25			
32	106																										
33	109																										
34	112		15		15	20	12	15	13					12			10	12	15	15	14	12	15	12	20	15	
35	119	14																				20	20	20			
36	118		12		10		10	10	10	10	10			15	10	10	10		12	10	10	10	10	14	12		
37	121			15		15																15	15	15			
38	125													5								5	5	5			
39	128				5		8	10	5	9	8			10	8	5	10		10	8	5	8	10	8			
40	131	9	8	10		10								5								5	10	8			
41	134																										



# Языки сопровождения

- Языки сопровождения служат для общения пользователя с ЭВМ.
- Такие языки должны поддерживать диалоговый режим.





# Промежуточные языки

- Промежуточные языки используются для поэтапной трансляции исходных программ.
- Промежуточные языки должны быть адаптивны к проявлению новых языков высокого уровня.

```
0050A02F: 0F8565030000    jne 0050A39A1
0050A035: 8D852E040000    lea eax, [ebp+0000042E]
0050A03B: 50              push eax
0050A03C: FF954D0F0000    call [ebp+00000F4D]
0050A042: 898526040000    mov [ebp+00000426], eax
0050A048: 8BF8           mov edi, eax
0050A04A: 8D5D5E         lea ebx, [ebp+5E]
0050A04D: 53            push ebx
0050A04E: 50            push eax
0050A04F: FF95490F0000    call [ebp+00000F49]
0050A055: 89854D050000    mov [ebp+0000054D], eax
0050A05B: 8D5D6B         lea ebx, [ebp+6B]
0050A05E: 53            push ebx
0050A05F: 57            push edi
0050A060: FF95490F0000    call [ebp+00000F49]
0050A066: 898551050000    mov [ebp+00000551], eax
0050A06C: 8D4577         lea eax, [ebp+77]
0050A06F: FFE0          jmp eax
0050A071: 56            push esi
```

# Внутренние языки

- Внутренние языки устанавливают единую форму представления данных для различных подсистем САПР.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <category uniqueID="1">
  <title />
- <category uniqueID="1_1">
  <title>Geek jamboree</title>
- <category uniqueID="1_1_1">
  <title>Articles and sample book chapters</title>
- <category uniqueID="1_1_1_1">
  <title>Collab, P2P, Swarming and NW</title>
+ <item uniqueID="1_1_1_1_1">
- <item uniqueID="1_1_1_1_2">
  <title>CypherSpace : P2P, Resilient systems and anonymity</title>
  <link>http://www.cypherspace.org/</link>
</item>
+ <item uniqueID="1_1_1_1_3">
+ <item uniqueID="1_1_1_1_4">
- <item uniqueID="1_1_1_1_5">
  <title>Fcast : Multicast file transfer</title>
  <link>http://research.microsoft.com/barc/mbone/fcast.asp</link>
</item>
+ <item uniqueID="1_1_1_1_6">
+ <item uniqueID="1_1_1_1_7">
```

# Заключение

- В данной презентации были рассмотрены различные языки, используемые в САПР.
- Была приведена классификация и подробное описание каждого языка.

**Спасибо за внимание!**