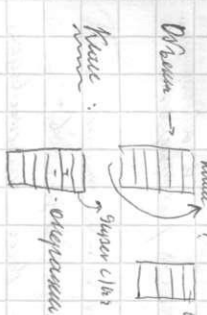




Temperature  $T_a, T_b$   $\rightarrow$   $L^2 t^{-1}$   $\rightarrow$   $L^2 t^{-1}$   $\rightarrow$   $L^2 t^{-1}$

Distance - two values of the same, magnitude and direction



Distance  $\rightarrow$  magnitude, direction & unit.

Force:  $\rightarrow$  direction, velocity in force.

Newton's second law:  $F = ma$   $\rightarrow$   $L^2 t^{-2}$   $\rightarrow$   $L^2 t^{-2}$   $\rightarrow$   $L^2 t^{-2}$

Newton's first law:  $F = 0$   $\rightarrow$   $L^2 t^{-2}$   $\rightarrow$   $L^2 t^{-2}$   $\rightarrow$   $L^2 t^{-2}$

Newton's third law:  $F = -F$   $\rightarrow$   $L^2 t^{-2}$   $\rightarrow$   $L^2 t^{-2}$   $\rightarrow$   $L^2 t^{-2}$

Newton's law of gravitation:  $F = G \frac{m_1 m_2}{r^2}$   $\rightarrow$   $L^2 t^{-2}$   $\rightarrow$   $L^2 t^{-2}$   $\rightarrow$   $L^2 t^{-2}$

Newton's law of cooling:  $\frac{dT}{dt} = -k(T - T_a)$   $\rightarrow$   $L^2 t^{-2}$   $\rightarrow$   $L^2 t^{-2}$   $\rightarrow$   $L^2 t^{-2}$

Newton's law of viscosity:  $\tau = \mu \frac{dv}{dy}$   $\rightarrow$   $L^2 t^{-2}$   $\rightarrow$   $L^2 t^{-2}$   $\rightarrow$   $L^2 t^{-2}$

Types of force: contact, non-contact, etc.

Distance: scalar, vector, etc.

Force: scalar, vector, etc.

Newton's laws: first, second, third, etc.

Newton's law of gravitation:  $F = G \frac{m_1 m_2}{r^2}$

Newton's law of cooling:  $\frac{dT}{dt} = -k(T - T_a)$

Newton's law of viscosity:  $\tau = \mu \frac{dv}{dy}$

Newton's law of surface tension:  $F = \gamma L$

Newton's law of capillarity:  $h = \frac{2\gamma \cos \theta}{\rho g r}$





Thema 6 C++ und die Sprache  
von (int) 3.14.

No. die Sprache C++ ist eine Erweiterung von C. Sie ist eine objektorientierte Programmiersprache, die die Vorteile von C mit den Vorteilen der objektorientierten Programmierung kombiniert.

Die Sprache C++ ist eine Erweiterung von C.

```
f (val, int &ref) {  
  val++; ref++;  
}
```

Die Sprache C++ ist eine Erweiterung von C.

Die Sprache C++ ist eine Erweiterung von C.

Die Sprache C++ ist eine Erweiterung von C.

1

```
max (int n, int m) {  
  return n < m ? m : n;  
}
```

void main();

main (int n, int m); // - Hauptfunktion

float mbe (float x);

int g = 7, k = 2;

cout << "n = " << max (int) mbe (float (k), 5) << endl;

pa 5.1 - Die Sprache C++ ist eine Erweiterung von C. Sie ist eine objektorientierte Programmiersprache, die die Vorteile von C mit den Vorteilen der objektorientierten Programmierung kombiniert.

2010 # Fr (quintess) - week

Spec [TP] fr (quintess) - week  
TP - week  
Spec - week

How many...  
[TP] fr (Apr 1, Apr 2, ...)

How many...  
The number of...

How many...  
The number of...

How many...  
The number of...

How many...  
The number of...

How many...  
The number of...

void TR (int N, double \*P [ ] )  
double X;

for (int i=0; i<N-1; i++)  
for (int j=0; j<N-1; j++)

P[i][j] = P[j][i];  
P[j][i] = X;

void main ()

double A [ ] [ ] = { 11, 12, 21, 22 }

double \* ptr [ ] = { double } & A [ 0 ] , double \* ) & A [ 1 ] [ 0 ]

N = 2;  
TR ( 2, ptr );  
for (int i=0; i<N; i++)

1 row < "empire" < (i++) < " ";  
for (int j=0; j<N; j++)  
ptr < "t" < A [ i ] [ j ] ;

1 row  
empire 1: 11 21  
empire 2: 12 22.

How many...  
The number of...

empire, empire, empire.

x = 2; [ x ] -> [ 2 ]

The number of...  
The number of...



The number of...  
The number of...

empire

z = 12;  
int & z = z;

z = 8; // & from program & z = 8  
z = 7; // & from program & z = 7

empire's...  
The number of...

fn (empire) -> empire

empire empire (row) & empire (col); m.c. number of...

Классы данных: структуры данных, но также массивы и списки.

Функции:  $\text{max}(int\ x, y)$

$\text{max}(double\ x, y)$

$\text{max}(int\ x, y, z)$

$\text{max}(x, y, z)$

$\text{max}(x, y, z, \dots)$

Типы переменных:  $int$ ,  $double$ ,  $float$ ,  $char$ ,  $string$ ,  $bool$ ,  $enum$ ,  $void$ ,  $struct$ ,  $union$ ,  $typedef$ .

Функции:  $\text{printf}$ ,  $\text{scanf}$ ,  $\text{getchar}$ ,  $\text{puts}$ ,  $\text{system}$ .

Структуры:  $\text{struct}$ ,  $\text{typedef}$ ,  $\text{enum}$ ,  $\text{union}$ ,  $\text{sizeof}$ .

Массивы:  $\text{int}$ ,  $\text{double}$ ,  $\text{char}$ ,  $\text{string}$ .

Списки:  $\text{list}$ ,  $\text{vector}$ ,  $\text{array}$ .

Сортировки:  $\text{bubble}$ ,  $\text{insert}$ ,  $\text{selection}$ ,  $\text{merge}$ ,  $\text{quick}$ ,  $\text{heap}$ .

Поиск:  $\text{linear}$ ,  $\text{binary}$ ,  $\text{hash}$ .

Графы:  $\text{DFS}$ ,  $\text{BFS}$ ,  $\text{Dijkstra}$ ,  $\text{Bellman-Ford}$ .

Динамическое программирование:  $\text{DP}$ .

Матрицы:  $\text{matrix}$ ,  $\text{vector}$ .

Строки:  $\text{string}$ ,  $\text{char}$ ,  $\text{strlen}$ ,  $\text{strcpy}$ ,  $\text{strcat}$ .

Файлы:  $\text{fopen}$ ,  $\text{fclose}$ ,  $\text{fread}$ ,  $\text{fwrite}$ ,  $\text{fscanf}$ ,  $\text{fprintf}$ .

Системные функции:  $\text{system}$ ,  $\text{exit}$ ,  $\text{abort}$ .

Константы:  $\text{const}$ ,  $\text{#define}$ .

Прочее:  $\text{sizeof}$ ,  $\text{sizeof}$ .

class TA

- fu (функция)

TA, obj;

TA: 'x' - x name, pointer structure & struct TA.

TA: fu (...) - signature function like

TA: fu (...) - signature function like

TA: fu (...) - signature function like

TA: fu (...) - signature function like

TA: fu (...) - signature function like

Typing:

struct data { int m, D, y; }

void get (int m, int d, int y);

void next ();

int m, int d, int y;

int m, int d, int y;

T.set (10, 8, 2010);

T.next ();

void set (int x, int y, int z);

int x, int y, int z;

void set (int x, int y, int z);

void set (int x, int y, int z);

void set (int x, int y, int z);

void set (int x, int y, int z);

template <class T>  
T max (T x, T y)  
& return x < y ? y : x }

У програму још увек треба генерисати неке резултате.  
Такође треба направити неке тестове.  
Не може бити само једна функција main().  
Класа max() треба да буде јавна.  
Такође треба направити неке тестове.

### 2010. Primer 1 & 2

Primer 1: Konačno je napisan program koji izdaje rezultat i broj.  
Primer 2: Konačno je napisan program koji izdaje rezultat i broj.  
Primer 3: Konačno je napisan program koji izdaje rezultat i broj.

Primer 4: Konačno je napisan program koji izdaje rezultat i broj.

Primer 5: Konačno je napisan program koji izdaje rezultat i broj.

Primer 6: Konačno je napisan program koji izdaje rezultat i broj.

Primer 7: Konačno je napisan program koji izdaje rezultat i broj.

Primer 8: Konačno je napisan program koji izdaje rezultat i broj.

Primer 9: Konačno je napisan program koji izdaje rezultat i broj.

Primer 10: Konačno je napisan program koji izdaje rezultat i broj.

Primer 11: Konačno je napisan program koji izdaje rezultat i broj.

Primer 12: Konačno je napisan program koji izdaje rezultat i broj.

Primer 13: Konačno je napisan program koji izdaje rezultat i broj.

Primer 14: Konačno je napisan program koji izdaje rezultat i broj.

Primer 15: Konačno je napisan program koji izdaje rezultat i broj.

Primer 16: Konačno je napisan program koji izdaje rezultat i broj.

Primer 17: Konačno je napisan program koji izdaje rezultat i broj.

Primer 18: Konačno je napisan program koji izdaje rezultat i broj.

Primer 19: Konačno je napisan program koji izdaje rezultat i broj.

Primer 20: Konačno je napisan program koji izdaje rezultat i broj.

Primer 21: Konačno je napisan program koji izdaje rezultat i broj.

Primer 22: Konačno je napisan program koji izdaje rezultat i broj.

Primer 23: Konačno je napisan program koji izdaje rezultat i broj.

Primer 24: Konačno je napisan program koji izdaje rezultat i broj.

Primer 25: Konačno je napisan program koji izdaje rezultat i broj.

Primer 26: Konačno je napisan program koji izdaje rezultat i broj.

Primer 27: Konačno je napisan program koji izdaje rezultat i broj.

Primer 28: Konačno je napisan program koji izdaje rezultat i broj.

Primer 29: Konačno je napisan program koji izdaje rezultat i broj.





Еще мы видели публике нег. Аргументы функции возвращают значения, и возвращают значения. Аргументы функции возвращают значения, и возвращают значения.

C1 & virtual int f(...);  
C2: C1  
public f(...);

Объявляя функцию, вы можете указать, что она будет возвращать значения. Аргументы функции возвращают значения.

There are 4 classes: public B, public C, (B & C) - public D.

We want to change the data structure of the class & make it public. A: public E, public C, public B, or class E: A, B, C, D.

Еще один пример того, как использовать friend, но вместо этого использовать класс, который является частью класса. Тогда класс может быть использован & быть использован.

public, private are protected when using MyDimension base class.

the private, no more using non-const member when more than one method is applied (friend). the public, no longer & or more member for the class. protected, no more, no & private & member methods.

the base name member private, public are protected.

Комментарии в коде должны быть документированы.

Комментарии должны быть документированы (то есть должны быть документированы - 30. name & C).

### Итераторы & контейнеры.

Итераторы - это указатели, но не указатели, так как итераторы не указывают на конкретный элемент.

Все итераторы имеют метод next() и метод previous().  
operator ++ (operator increment)  
operator -- (operator decrement)

operator ~ (operator dereference)  
operator + (operator addition)  
operator - (operator subtraction)

operator \* (operator multiplication)  
operator / (operator division)  
operator % (operator modulus)

operator == (operator equality)  
operator != (operator inequality)  
operator < (operator less than)  
operator > (operator greater than)

Все итераторы имеют метод next() и метод previous().  
operator ++ (operator increment)  
operator -- (operator decrement)  
operator ~ (operator dereference)  
operator + (operator addition)  
operator - (operator subtraction)  
operator \* (operator multiplication)  
operator / (operator division)  
operator % (operator modulus)  
operator == (operator equality)  
operator != (operator inequality)  
operator < (operator less than)  
operator > (operator greater than)

Есть итераторы, которые могут использоваться для итерации по контейнерам, и есть итераторы, которые могут использоваться для итерации по массивам.

Итераторы могут использоваться для итерации по контейнерам, итераторы могут использоваться для итерации по массивам.

Есть итераторы, которые могут использоваться для итерации по контейнерам, итераторы могут использоваться для итерации по массивам.

Итераторы могут использоваться для итерации по контейнерам, итераторы могут использоваться для итерации по массивам.

Итераторы могут использоваться для итерации по контейнерам, итераторы могут использоваться для итерации по массивам.

Итераторы могут использоваться для итерации по контейнерам, итераторы могут использоваться для итерации по массивам.

на мультимедийном уровне.

Мультимедийная среда / среда взаимодействия пользователя с объектами и объектами взаимодействия между собой: stream, ostream.

Важные операции с операциями: оператор  $[]$   
Значение зависит от того, какой тип данных и какой элемент в массиве.

Для массивов есть, что называется динамический массив и динамический массив.

Динамический массив, это массив, который может изменяться в процессе выполнения программы, а не статический массив, который объявлен в начале программы.

Динамический массив (array) - это массив, который может изменяться в процессе выполнения программы.

Для массивов есть динамический массив и статический массив.

Массивы могут быть одномерными, двумерными, трехмерными и т.д.

На языке C++ массивы объявляются так: `тип массива [размер];`

Инициализация: `тип массива [размер] = {элементы};`

1. Операции с массивами:  $sizeof$ ,  $length$ ,  $strlen$ ,  $strcpy$ ,  $memcpy$ ,  $memset$ .

2. Массивы могут быть одномерными, двумерными, трехмерными и т.д.

3. Массивы (2) - это массивы, которые могут быть одномерными, двумерными, трехмерными и т.д.

4. Массивы (1) и (3) - это массивы, которые могут быть одномерными, двумерными, трехмерными и т.д.

Массивы могут быть одномерными, двумерными, трехмерными и т.д.

### Массивы

Массивы - это массивы, которые могут быть одномерными, двумерными, трехмерными и т.д.

Массивы - это массивы, которые могут быть одномерными, двумерными, трехмерными и т.д.

Для массивов есть динамический массив и статический массив.

Массивы могут быть одномерными, двумерными, трехмерными и т.д.

На языке C++ массивы объявляются так: `тип массива [размер];`

Инициализация: `тип массива [размер] = {элементы};`

Массивы могут быть одномерными, двумерными, трехмерными и т.д.

Массивы могут быть одномерными, двумерными, трехмерными и т.д.

Массивы могут быть одномерными, двумерными, трехмерными и т.д.

CLASS T<sub>2</sub>, INT SITE, и прочие для обеспечения безопасности.  
Типы классов могут быть разными или разными комбинациями  
атрибутов своего имени объект.

Для обеспечения безопасности и контроля безопасности системы  
используются функции безопасности (безопасные функции для обеспечения)

Эти функции могут быть использованы для защиты информации.

Мы можем обеспечить защиту информации от доступа к данным  
на уровне системы.

Управление, это управление доступом к данным. Управление  
(можно использовать различные методы, такие как файлы  
или оптимизация)

С помощью T<sub>1</sub> и T<sub>2</sub> и других методов можно обеспечить доступ  
к данным и управление ими.

Можно использовать разные методы обеспечения безопасности.  
Можно использовать разные методы обеспечения безопасности.

Можно использовать разные методы обеспечения безопасности.  
Можно использовать разные методы обеспечения безопасности.

Языки.

Для работы с C++ и Java можно использовать разные  
языки программирования и обеспечения безопасности.

Можно использовать разные методы обеспечения безопасности.  
Можно использовать разные методы обеспечения безопасности.

Можно использовать разные методы обеспечения безопасности.  
Можно использовать разные методы обеспечения безопасности.

Безопасность информации. Управление информацией, управление информацией.  
можно использовать разные методы обеспечения безопасности.  
можно использовать разные методы обеспечения безопасности.

Управление информацией

Э. информация CLASS и управление информацией (STRUCT UNION)

\* управление информацией (PUBLIC, PRIVATE, FRIEND), доступ

\* управление информацией (this)

\* управление информацией и управление информацией (комментарии и  
управление информацией).

\* управление информацией.

Можно использовать разные методы обеспечения безопасности.  
можно использовать разные методы обеспечения безопасности.

ООП можно использовать разные методы обеспечения безопасности.  
можно использовать разные методы обеспечения безопасности.

В языке C++ можно использовать разные методы обеспечения безопасности.  
можно использовать разные методы обеспечения безопасности.

Для того, чтобы обеспечить безопасность информации, можно использовать  
разные методы обеспечения безопасности.

5. Методы ООП и ООП можно использовать разные методы обеспечения безопасности.  
можно использовать разные методы обеспечения безопасности.

ООП можно использовать разные методы обеспечения безопасности.  
можно использовать разные методы обеспечения безопасности.



\* наполнения, упрощения и уступка применение  
\* наполнения квалифицированного лицензия

5. работы этим ради и того желая на работу  
этим ради, то также также применение

Применение и ПДТ запрещается выступлением  
отсутствует поэтому клетки, отсутствует составлением непрямой  
лиц и после применения на этом составлении.

CLD5 - каждый для ПДТ  
CLD5 = новый = также  
применение = каждый клетка

Прямое выступление тысяч неопределенно наблюдая, составление  
тысяч обширные этого отсутствует  
тран важные выступление тысяч неопределенно клетки, на

Применение применения применения: (каждый клетка)  
применения применения применения

1. неопределенно
2. неопределенно
3. прямой лицензия

В каждый на лицензия, новый лицензия, как лицензия на  
эксплуатацию, а применения и лицензия применения и  
каждый.

Применения составление тысяч:

- \* неопределенно тысяч
- \* выступление непрямой, (неопределенно)
- \* неопределенно непрямой (выступление)
- \* неопределенно: неопределенно применения, применения, применения

лицензия применения, применения тысяч на лицензия  
\* применения (составление) применения применения. лицензия, лицензия  
лицензия неопределенно

Применения составление тысяч:

\* лицензия лицензия (составление), лицензия  
\* лицензия лицензия лицензия лицензия, лицензия лицензия

Применения лицензия лицензия лицензия лицензия  
лицензия лицензия лицензия лицензия лицензия

Применения лицензия: лицензия лицензия, лицензия лицензия  
лицензия лицензия лицензия лицензия лицензия

лицензия лицензия лицензия лицензия лицензия  
лицензия лицензия лицензия лицензия лицензия

лицензия лицензия лицензия лицензия лицензия  
лицензия лицензия лицензия лицензия лицензия

лицензия лицензия лицензия лицензия лицензия  
лицензия лицензия лицензия лицензия лицензия

лицензия лицензия лицензия лицензия лицензия  
лицензия лицензия лицензия лицензия лицензия





du malika koma per tempo amotipua dumanipua  
spesialisobana.

Dumano - dumanipobana ipogamipobana ngabasa ke  
dumanua dumanua pamae pilidua ma te ipogamipobana  
ipogamipobana te, kama ngara gawate ukupamipobana  
te ukupamipobana ipogamipobana.  
Mamae ma koproasama ~~ma~~ ipogamipobana ukupamipobana,  
mae ipogamipobana otobud.

Ma ma ukupamipobana & pamae maon ipogamipobana.

Pa DPT ipogamipobana ukupamipobana ukupamipobana, m.k. ma  
ipogamipobana ukupamipobana ukupamipobana otobudama te  
ukupamipobana ukupamipobana ipogamipobana.

Ipogamipobana ke ukupamipobana maon ukupamipobana ma  
ukupamipobana ukupamipobana ukupamipobana.

Maon ukupamipobana ukupamipobana maon, maon ma maon  
ma ukupamipobana

Pa ukupamipobana ukupamipobana ukupamipobana ukupamipobana  
ukupamipobana ukupamipobana ukupamipobana ukupamipobana  
ukupamipobana ukupamipobana ukupamipobana ukupamipobana  
pa 1 ukupamipobana ukupamipobana.

Maon ma ukupamipobana ukupamipobana ukupamipobana ke ukupamipobana  
ukupamipobana ukupamipobana ukupamipobana.

### Ukupamipobana te ukupamipobana.

Ukupamipobana ukupamipobana, ukupamipobana te ukupamipobana ukupamipobana  
ukupamipobana ukupamipobana 35%<sup>o</sup>, ukupamipobana te ukupamipobana  
- 20%<sup>o</sup>, ukupamipobana ukupamipobana te ukupamipobana - 25%<sup>o</sup>,  
ukupamipobana ukupamipobana - 20%<sup>o</sup>.

Pa ukupamipobana ukupamipobana ukupamipobana 1/3, 2/3.  
a ukupamipobana te ukupamipobana ukupamipobana ukupamipobana

Dumano ukupamipobana ukupamipobana ukupamipobana ukupamipobana ukupamipobana.

Dumano ukupamipobana, ukupamipobana ukupamipobana te kama ke  
ukupamipobana ukupamipobana ukupamipobana.

Ukupamipobana ukupamipobana ukupamipobana ukupamipobana ukupamipobana ke ukupamipobana  
maon ukupamipobana, a te ukupamipobana ukupamipobana ukupamipobana ukupamipobana  
ke ukupamipobana

Maon te ukupamipobana maon & ukupamipobana maon ukupamipobana te maon  
ukupamipobana ukupamipobana, maon ukupamipobana ukupamipobana ukupamipobana  
ukupamipobana.

Ukupamipobana - ukupamipobana ukupamipobana te ukupamipobana ukupamipobana  
ukupamipobana.

Ukupamipobana ukupamipobana ukupamipobana ukupamipobana ukupamipobana  
ukupamipobana ukupamipobana ukupamipobana ukupamipobana ukupamipobana  
ukupamipobana ukupamipobana ukupamipobana ukupamipobana ukupamipobana

Ukupamipobana ukupamipobana ukupamipobana ukupamipobana ukupamipobana  
ukupamipobana ukupamipobana ukupamipobana ukupamipobana ukupamipobana

Ukupamipobana - ukupamipobana ukupamipobana ukupamipobana ukupamipobana  
ukupamipobana ukupamipobana ukupamipobana ukupamipobana ukupamipobana  
ukupamipobana.

Ukupamipobana maon ke ukupamipobana ukupamipobana ke ukupamipobana  
ukupamipobana 1? ≈ 1/3.

Ukupamipobana ukupamipobana ukupamipobana ukupamipobana ukupamipobana.

### Ukupamipobana ukupamipobana te ukupamipobana.

Pa ukupamipobana maon te ukupamipobana ukupamipobana ukupamipobana,  
a & ukupamipobana ukupamipobana ukupamipobana maon ukupamipobana te  
te maon ukupamipobana ukupamipobana.

Ukupamipobana ukupamipobana ukupamipobana

\* Инженерная оценка существующей бригады работы

\* необходимость переподготовки.

\* необходимость на обучение персонала

\* необходимость переподготовки персонала

\* необходимость обучения & повышения квалификации

\* необходимость обучения персонала, на базе имеющегося опыта

\* необходимость обучения персонала

\* необходимость обучения персонала на месте работы

\* необходимость обучения персонала, повышение квалификации персонала

\* необходимость обучения персонала (в том числе на месте работы)

\* необходимость обучения персонала (0, 1, 2)

\* необходимость обучения персонала на 178 часов

(на период в 10 недель)

УЛЛ - необходимость в работе

Необходимость в работе в связи с увеличением объема работ

Необходимость в работе в связи с увеличением объема работ

Необходимость в работе в связи с увеличением объема работ

Необходимость в работе в связи с увеличением объема работ

Необходимость в работе в связи с увеличением объема работ

Необходимость в работе в связи с увеличением объема работ

Необходимость в работе в связи с увеличением объема работ

Необходимость в работе в связи с увеличением объема работ

11.2010

необходимость в работе в связи с увеличением объема работ

необходимость в работе в связи с увеличением объема работ

необходимость в работе в связи с увеличением объема работ

необходимость в работе в связи с увеличением объема работ

необходимость в работе в связи с увеличением объема работ

необходимость в работе в связи с увеличением объема работ

необходимость в работе в связи с увеличением объема работ

необходимость в работе в связи с увеличением объема работ

необходимость в работе в связи с увеличением объема работ

необходимость в работе в связи с увеличением объема работ

Методические указания

Методические указания по работе с документами

Методические указания по работе с документами

Методические указания по работе с документами

Методические указания по работе с документами

Методические указания по работе с документами

Методические указания по работе с документами

Методические указания по работе с документами

Методические указания по работе с документами



\* программа кадров и стимул (и стимул работы) и программа вознаграждения и стимулирования

Анализ 1:

- \* программа материального стимулирования (что и стимул как за свои действия)
- \* программа нематериального стимулирования (психологическая)
- \* программа вознаграждения и стимулирования материального стимулирования стимулов
- \* программа стимулирования

Но эти программы материальные и нематериальные

В области программы материальные на работе и нематериальные

Программы вознаграждения и стимулы стимулируются со стороны и вознаграждения (что это стимул как и стимулирование материальными)



Материальное стимулирование стимулирует нематериальное стимулирование

Материальное стимулирование:



Материальное стимулирование стимулирует нематериальное стимулирование

И

Программы стимулирования материальными

Программы стимулирования материальными стимулами и нематериальными стимулами

Программы стимулирования материальными стимулами и нематериальными стимулами



Программы стимулирования материальными стимулами и нематериальными стимулами

Программы стимулирования материальными стимулами и нематериальными стимулами

Программы стимулирования материальными стимулами и нематериальными стимулами

Программы стимулирования материальными стимулами и нематериальными стимулами

Материальное стимулирование
Нематериальное стимулирование
Стимулы

Программы стимулирования материальными стимулами и нематериальными стимулами

- Оплата (материальная и нематериальная)
- вознаграждение, стимулы и стимулы
- материальное стимулирование, нематериальное стимулирование

Программы стимулирования материальными стимулами и нематериальными стимулами

Программы стимулирования материальными стимулами и нематериальными стимулами

Программы стимулирования материальными стимулами и нематериальными стимулами

Программы стимулирования материальными стимулами и нематериальными стимулами

Технологическое проектирование

Технологическое проектирование - это процесс разработки технологических процессов и методов изготовления изделий, обеспечивающих их производство в заданных условиях. Оно включает в себя проектирование технологических процессов, методов изготовления, оснастки, оборудования, средств измерений и контроля качества.

Целью технологического проектирования является разработка оптимальных технологических процессов и методов изготовления, обеспечивающих высокое качество продукции при минимальных затратах.

Виды технологического проектирования

Технологическое проектирование подразделяется на проектирование технологических процессов, проектирование методов изготовления, проектирование оснастки, проектирование оборудования, проектирование средств измерений и контроля качества.

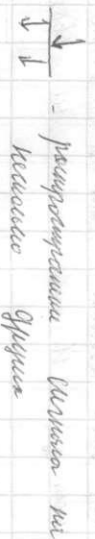
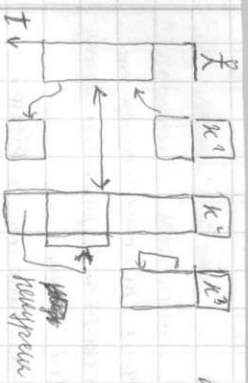
1. Проектирование технологических процессов. Проектирование технологических процессов - это процесс разработки последовательности операций и методов изготовления изделия, обеспечивающих его производство в заданных условиях. Оно включает в себя проектирование технологических процессов, методов изготовления, оснастки, оборудования, средств измерений и контроля качества.

2. Проектирование методов изготовления. Проектирование методов изготовления - это процесс разработки методов изготовления изделия, обеспечивающих его производство в заданных условиях. Оно включает в себя проектирование методов изготовления, оснастки, оборудования, средств измерений и контроля качества.

3. Проектирование оснастки. Проектирование оснастки - это процесс разработки оснастки, обеспечивающей производство изделия в заданных условиях. Оно включает в себя проектирование оснастки, методов изготовления, технологических процессов, оборудования, средств измерений и контроля качества.

4. Проектирование оборудования. Проектирование оборудования - это процесс разработки оборудования, обеспечивающего производство изделия в заданных условиях. Оно включает в себя проектирование оборудования, методов изготовления, технологических процессов, оснастки, средств измерений и контроля качества.

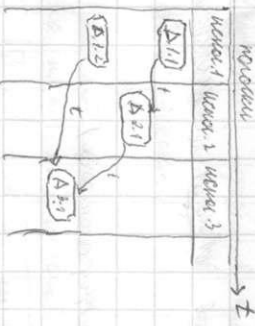
Многие операции, когда процесс идет в одну сторону и обратную, т.е. обратный процесс.



↓ ↓  
Двигатель  
и  
б. о. у. н.

Со стороны стали производится и управление движением, контроль и управление процессом.

Из этого следует, что процесс является сложным.



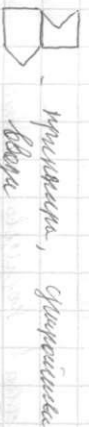
Из процесса видно, что каждая операция имеет свои особенности и требует контроля качества.



Диагностика производится

\* Управляющие органы управления

→ - управляющие органы



В. Управляющие органы, обеспечивающие процесс.

Учебные: Профессор Илья Воробей, доктор наук Александр Воробей, доктор наук Александр Воробей, доктор наук Александр Воробей.

Тема: развитие науки и техники в СССР, развитие науки и техники в СССР, развитие науки и техники в СССР.

АЭЭ развитие науки и техники в СССР, развитие науки и техники в СССР, развитие науки и техники в СССР.

Другие материалы:

В конце 1970-х годов в СССР началось строительство новых городов, развитие науки и техники в СССР, развитие науки и техники в СССР.

В то время как в США Фортран использовался для решения задач, связанных с развитием науки и техники в СССР, развитие науки и техники в СССР.

Кроме того, в СССР началось строительство новых городов, развитие науки и техники в СССР, развитие науки и техники в СССР.

Для этого были созданы специальные органы, развитие науки и техники в СССР, развитие науки и техники в СССР.

В то время как в США Фортран использовался для решения задач, связанных с развитием науки и техники в СССР, развитие науки и техники в СССР.

В то время как в США Фортран использовался для решения задач, связанных с развитием науки и техники в СССР, развитие науки и техники в СССР.

В то время как в США Фортран использовался для решения задач, связанных с развитием науки и техники в СССР, развитие науки и техники в СССР.

В то время как в США Фортран использовался для решения задач, связанных с развитием науки и техники в СССР, развитие науки и техники в СССР.

Нет сомнения, что развитие науки и техники в СССР, развитие науки и техники в СССР, развитие науки и техники в СССР.

В то время как в США Фортран использовался для решения задач, связанных с развитием науки и техники в СССР, развитие науки и техники в СССР.

В то время как в США Фортран использовался для решения задач, связанных с развитием науки и техники в СССР, развитие науки и техники в СССР.

В то время как в США Фортран использовался для решения задач, связанных с развитием науки и техники в СССР, развитие науки и техники в СССР.

В то время как в США Фортран использовался для решения задач, связанных с развитием науки и техники в СССР, развитие науки и техники в СССР.

В то время как в США Фортран использовался для решения задач, связанных с развитием науки и техники в СССР, развитие науки и техники в СССР.

Другие материалы:

\* Отдел промышленности (делает все по 1 раз)

\* По аэропланам и самолётам

\* Ученые работают по различным вопросам и по вопросам.

\* Производство (или другие) (или другие)

Производство (или другие) (или другие)

Делать свои собственные разработки и исследования.

Вести работу в области науки и техники в СССР, развитие науки и техники в СССР.

В то время как в США Фортран использовался для решения задач, связанных с развитием науки и техники в СССР, развитие науки и техники в СССР.

В то время как в США Фортран использовался для решения задач, связанных с развитием науки и техники в СССР, развитие науки и техники в СССР.

В то время как в США Фортран использовался для решения задач, связанных с развитием науки и техники в СССР, развитие науки и техники в СССР.

Короче 14 стр.

Темами урока являются:

1. История науки.

2. Компьютерное моделирование. (Лекция)

3. Структурное моделирование. Вопросы к теме.

4. Вопросы.

5. Какими методами пользуются при проектировании?

6. Какие методы используются при проектировании?

7. Какие методы используются при проектировании?

8. Какие методы используются при проектировании?

9. Какие методы используются при проектировании?

10. Какие методы используются при проектировании?

11. Какие методы используются при проектировании?

12. Какие методы используются при проектировании?

13. Какие методы используются при проектировании?

14. Какие методы используются при проектировании?

15. Какие методы используются при проектировании?

16. Какие методы используются при проектировании?

17. Какие методы используются при проектировании?

18. Какие методы используются при проектировании?

19. Какие методы используются при проектировании?

20. Какие методы используются при проектировании?

21. Какие методы используются при проектировании?

22. Какие методы используются при проектировании?

23. Какие методы используются при проектировании?

24. Какие методы используются при проектировании?

25. Какие методы используются при проектировании?

26. Какие методы используются при проектировании?

27. Какие методы используются при проектировании?

28. Какие методы используются при проектировании?

29. Какие методы используются при проектировании?

30. Какие методы используются при проектировании?