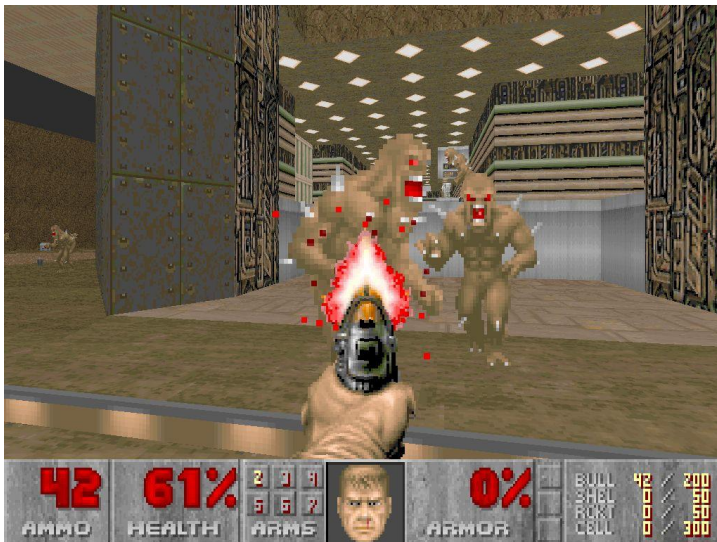


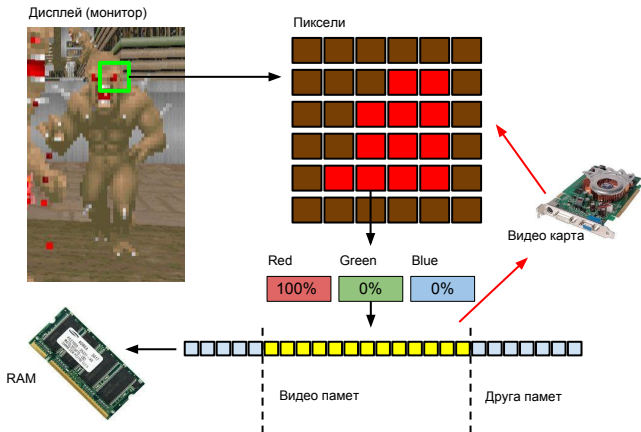
Въведение в курса. Програми, езици от високо ниво,  
базова структура на програма, променливи, вход и  
изход, условен оператор и цикъл, бройни системи,  
Машины на Тюринг

18 октомври 2022 г.

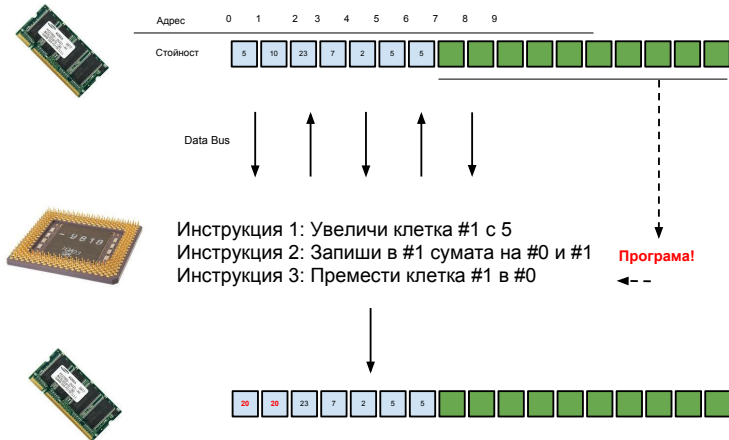
# Програмиране?



# Как работи?



# Програми



# Език от високо ниво

```
int a = 5;
int b = 10;
```

```
b = b + 5;
b = a + b;
a = b;
```

КОМПИЛАТОР--->



# Променливи

- Стойност

```
int a = 5;  
int b = 10;
```



- Адрес

- Присвояване на стойност

```
b = a + b;
```

- Последователност на операциите

```
int a = 5;  
int b = 10;  
b = a + b;  
b = 17 + b;
```

# Език за програмиране C++

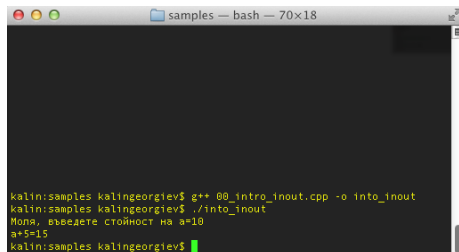
# Базова структура. Вход/изход

```
#include <iostream>
using namespace std;

int main ()
{
    int a;

    cout << "Please, input the value of a=";
    cin >> a;
    cout << "a+5=" << a+5 << endl;

    return 0;
}
```



```
kalin:samples kalingeorgiev$ g++ @0_intro_inout.cpp -o into_inout
kalin:samples kalingeorgiev$ ./into_inout
Моля, въведете стойност на a=10
a+5=15
kalin:samples kalingeorgiev$
```

- Променлива
- Константи: низови, числови
- Аритметични операции
- Конзола



# Пример: Средно аритметично

```
#include <iostream>
using namespace std;

int main ()
{
    int a,b,c;    //DOUBLE!!!

    cout << "Please ,_input_the_value_of_a=";
    cin >> a;
    cout << "Please ,_input_the_value_of_b=";
    cin >> b;
    cout << "Please ,_input_the_value_of_c=";
    cin >> c;

    cout << "average=" << (a+b+c)/3 << endl;

    return 0;
}
```

# Условен оператор

- Проверка на условие:

```
int a;  
  
cin >> a;  
  
if (a > 5) {  
    cout << "a_is_greater_than_5";  
} else {  
    cout << "a_is_less_than_or_equal_to_5";  
}
```

- Прости видове условия за числа: >, >=, <, <=, ==, !=

# Вложени оператори

```
int a = 701;

if (a > 20){
    if (a < 200){
        cout << "CASE_1";
    } else if (a < 700) {
        cout << "CASE_2";
    } else {
        cout << "CASE_3";
    }
}
```

# Вложени оператори

```
int a = 701;

if (a > 20){
    if (a < 200){
        cout << "CASE_1";
    } else if (a < 700) {
        cout << "CASE_2";
    } else {
        cout << "CASE_3";
    }
}
```

```
int a = 701;

if (a > 20){
    if (a < 200){
        cout << "CASE_1";
    }
    else
        if (a < 700) {
            cout << "CASE_2";
        }
}
else {
    cout << "CASE_3";
}
```

# Пример: най-голямото от 3 числа

```
if (a > b){
    if (b > c){
        cout << "max_=" << a << endl;
    } else if (a > c)
        cout << "max_=" << a << endl;
    } else {
        cout << "max_=" << c << endl;
    }
} else //b >= a {
    if (a > c){
        cout << "max_=" << b << endl;
    } else if (b > c)
        cout << "max_=" << b << endl;
    } else {
        cout << "max_=" << c << endl;
    }
}
```

```
if (a > b){
    if (b > c || a > c){
        cout << "max_=" << a << endl;
    } else {
        cout << "max_=" << c << endl;
    }
} else {
    if (a > c || b > c){
        cout << "max_=" << b << endl;
    } else {
        cout << "max_=" << c << endl;
    }
}

if (a > b && a > c){
    cout << "max_=" << a << endl;
} else if (b > a && b > c) {
    cout << "max_=" << b << endl;
} else {
    cout << "max_=" << c << endl;
}
```

*Как може допълнително да се опрости тази програма?*

# Пример: най-голямото от 3 числа

```

if (a > b){
    if (b > c){
        cout << "max_=" << a << endl;
    } else if (a > c)
        cout << "max_=" << a << endl;
    } else {
        cout << "max_=" << c << endl;
    }
} else //b >= a {
    if (a > c){
        cout << "max_=" << b << endl;
    } else if (b > c)
        cout << "max_=" << b << endl;
    } else {
        cout << "max_=" << c << endl;
    }
}

```

```

if (a > b){
    if (b > c || a > c){
        cout << "max_=" << a << endl;
    } else {
        cout << "max_=" << c << endl;
    }
} else {
    if (a > c || b > c){
        cout << "max_=" << b << endl;
    } else {
        cout << "max_=" << c << endl;
    }
}

if (a > b && a > c){
    cout << "max_=" << a << endl;
} else if (b > a && b > c) {
    cout << "max_=" << b << endl;
} else {
    cout << "max_=" << c << endl;
}

```

Как може допълнително да се опрости тази програма?

# Пример: най-голямото от 3 числа

```
if (a > b){
    if (b > c){
        cout << "max_=" << a << endl;
    } else if (a > c)
        cout << "max_=" << a << endl;
    } else {
        cout << "max_=" << c << endl;
    }
} else //b >= a {
    if (a > c){
        cout << "max_=" << b << endl;
    } else if (b > c)
        cout << "max_=" << b << endl;
    } else {
        cout << "max_=" << c << endl;
    }
}
```

```
if (a > b){
    if (b > c || a > c){
        cout << "max_=" << a << endl;
    } else {
        cout << "max_=" << c << endl;
    }
} else {
    if (a > c || b > c){
        cout << "max_=" << b << endl;
    } else {
        cout << "max_=" << c << endl;
    }
}

if (a > b && a > c){
    cout << "max_=" << a << endl;
} else if (b > a && b > c) {
    cout << "max_=" << b << endl;
} else {
    cout << "max_=" << c << endl;
}
```

Как може допълнително да се опрости тази програма?

# Булеви (логически) операции AND ( $\wedge$ ) и OR ( $\vee$ )

$\&\&$	true	false
true	true	false
false	false	false

$\ \ $	true	false
true	true	true
false	true	false



## Пример: Корени на $ax^2 + bx + c = 0$

```
double a, b, c;

cin >> a >> b >> c;

double D = b*b - 4*a*c;

if (D < 0){
    cout << "NO_roots!";
} else if (D == 0) {
    cout << "ONE_root, x=" << (-b)/2*a << endl;
} else {
    cout << "TWO_roots, x1=" << (-b-sqrt(D))/2*a << endl <<
        "x2=" << (-b+sqrt(D))/2*a << endl;
}
```

## Циклични процеси

# Пример: Средно аритметично (отново)

```
#include <iostream>
using namespace std;

int main ()
{

    int a,b,c;    //DOUBLE!!!

    cout << "Please ,_input_the_value_of_a=";
    cin >> a;
    cout << "Please ,_input_the_value_of_b=";
    cin >> b;
    cout << "Please ,_input_the_value_of_c=";
    cin >> c;

    cout << "average=" << (a+b+c)/3 << endl;

    return 0;

}
```

# Пример: Средно аритметично на 10 числа

```
#include <iostream>
using namespace std;

int main ()
{
    int number, sum = 0;

    for (int counter = 0; counter < 10; counter++){

        cout << "Please_enter_number_" << counter << ":";
        cin >> number;
        sum = sum + number;

    }

    cout << "The_average_is_" << sum / 10;

}
```

## Пример: Средно аритметично на N числа

```
#include <iostream>
using namespace std;

int main ()
{
    int number, sum = 0, numbersCount;

    cout << "Numbers count = ";
    cin >> numbersCount;

    for (int counter = 0; counter < numbersCount; counter++){

        cout << "Please enter number #" << counter << ":";
        cin >> number;
        sum = sum + number;

    }

    cout << "The average is " << sum / numbersCount;

}
```

## Съвсем малко теория

# Бройни системи

## Число (0x10)

$$2 \mid 3 \mid 4 = 2 * 10^2 + 3 * 10 + 4$$

- Какво става, ако имаме не 10, а 16 цифри
- 0,1,2,3,4,5,6,7,8,9,... ???
- A,B,C,D,E,F

## Число (0x16)

$$2 \mid 3 \mid 4 = 2 * 16^2 + 3 * 16 + 4$$

- Ами ако имаме само две цифри?

## Число (Binary)

$$1 \mid 0 \mid 1 = 1 * 2^2 + 0 * 2 + 1$$

- Защо бихме се ограничили до две цифри?

# Бройни системи

## Число (0x10)

$$2 \mid 3 \mid 4 = 2 * 10^2 + 3 * 10 + 4$$

- Какво става, ако имаме не 10, а 16 цифри
- 0,1,2,3,4,5,6,7,8,9,... ???
- A,B,C,D,E,F

## Число (0x16)

$$2 \mid 3 \mid 4 = 2 * 16^2 + 3 * 16 + 4$$

- Ами ако имаме само две цифри?

## Число (Binary)

$$1 \mid 0 \mid 1 = 1 * 2^2 + 0 * 2 + 1$$

- Защо бихме се ограничили до две цифри?



# Бройни системи

## Число (0x10)

$$2 \mid 3 \mid 4 = 2 * 10^2 + 3 * 10 + 4$$

- Какво става, ако имаме не 10, а 16 цифри
- 0,1,2,3,4,5,6,7,8,9,... ???
- A,B,C,D,E,F

## Число (0x16)

$$2 \mid 3 \mid 4 = 2 * 16^2 + 3 * 16 + 4$$

- Ами ако имаме само две цифри?

## Число (Binary)

$$1 \mid 0 \mid 1 = 1 * 2^2 + 0 * 2 + 1$$

- Защо бихме се ограничили до две цифри?

# Бройни системи

## Число (0x10)

$$2 \mid 3 \mid 4 = 2 * 10^2 + 3 * 10 + 4$$

- Какво става, ако имаме не 10, а 16 цифри
- 0,1,2,3,4,5,6,7,8,9,... ???
- A,B,C,D,E,F

## Число (0x16)

$$2 \mid 3 \mid 4 = 2 * 16^2 + 3 * 16 + 4$$

- Ами ако имаме само две цифри?

## Число (Binary)

$$1 \mid 0 \mid 1 = 1 * 2^2 + 0 * 2 + 1$$

- Защо бихме се ограничили до две цифри?

# Бройни системи

## Число (0x10)

$$2 \mid 3 \mid 4 = 2 * 10^2 + 3 * 10 + 4$$

- Какво става, ако имаме не 10, а 16 цифри
- 0,1,2,3,4,5,6,7,8,9,... ???
- A,B,C,D,E,F

## Число (0x16)

$$2 \mid 3 \mid 4 = 2 * 16^2 + 3 * 16 + 4$$

- Ами ако имаме само две цифри?

## Число (Binary)

$$1 \mid 0 \mid 1 = 1 * 2^2 + 0 * 2 + 1$$

- Защо бихме се ограничили до две цифри?

# Бройни системи

## Число (0x10)

$$2 \mid 3 \mid 4 = 2 * 10^2 + 3 * 10 + 4$$

- Какво става, ако имаме не 10, а 16 цифри
- 0,1,2,3,4,5,6,7,8,9,... ???
- A,B,C,D,E,F

## Число (0x16)

$$2 \mid 3 \mid 4 = 2 * 16^2 + 3 * 16 + 4$$

- Ами ако имаме само две цифри?

## Число (Binary)

$$1 \mid 0 \mid 1 = 1 * 2^2 + 0 * 2 + 1$$

- Защо бихме се ограничили до две цифри?

# Бройни системи

## Число (0x10)

$$2 \mid 3 \mid 4 = 2 * 10^2 + 3 * 10 + 4$$

- Какво става, ако имаме не 10, а 16 цифри
- 0,1,2,3,4,5,6,7,8,9,... ???
- A,B,C,D,E,F

## Число (0x16)

$$2 \mid 3 \mid 4 = 2 * 16^2 + 3 * 16 + 4$$

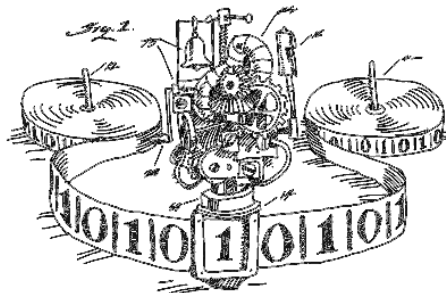
- Ами ако имаме само две цифри?

## Число (Binary)

$$1 \mid 0 \mid 1 = 1 * 2^2 + 0 * 2 + 1$$

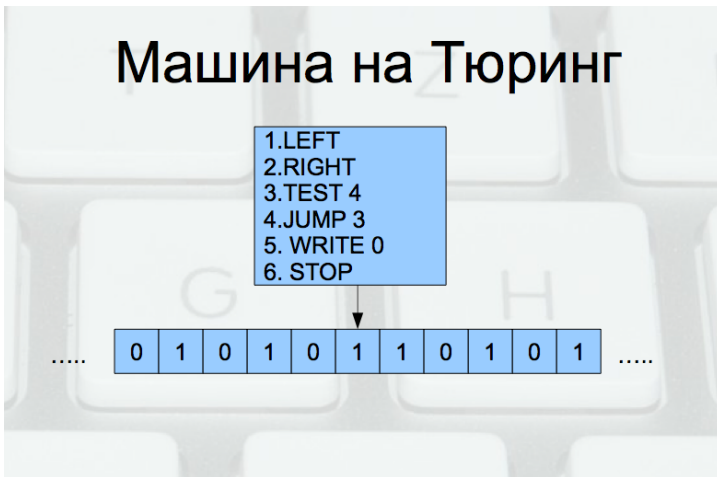
- Защо бихме се ограничили до две цифри?

# Машины на Тюринг



Машина на Тюринг  
изображение от <http://www.worldofcomputing.net/>

# Машина на Тюринг



- Пълно множество от операции

# Изчислимост

P:  
LEFT  
RIGHT  
TEST 4  
JUMP 3  
WRITE 0  
STOP

LEFT = 0  
RIGHT = 1  
TEST = 2  
JUMP = 3  
WRITE = 4  
STOP = 5

$$P = 0, 1, 2, 4, 3, 3, 4, 0, 5$$

$$N_P = 2^0 \times 3^1 \times 5^2 \times 7^4 \times 11^4 \times 13^3 \dots$$

- Всяка програма се кодира с едно естествено число
- За всички функции ли могат да се намерят програми? Изчислими и неизчислими функции



# Диагоналният метод на Кантор, 1891

$$\begin{array}{lcl}
 s_1 & = & 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \dots \\
 s_2 & = & 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ \dots \\
 s_3 & = & 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ \dots \\
 s_4 & = & 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ \dots \\
 s_5 & = & 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ \dots \\
 s_6 & = & 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ \dots \\
 s_7 & = & 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ \dots \\
 s_8 & = & 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ \dots \\
 s_9 & = & 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ \dots \\
 s_{10} & = & 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots \\
 s_{11} & = & 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ \dots \\
 \vdots & & \vdots\ \vdots\ \vdots\ \vdots\ \vdots\ \vdots\ \vdots\ \vdots\ \vdots\ \vdots\ \vdots\ \ddots
 \end{array}$$

$$s = 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ \dots$$

# Диагоналният метод на Кантор, 1891

- Следователно безкрайните редици от числа са “повече от числата”
- Какво значи “повече”?

# Неизчислими функции

- Нека  $\mathcal{A}$  е множеството на всички безкрайни редици над  $N$ .
- За всяка редица  $\{a\}_{\infty} \in \mathcal{A}$  да построим функция  $f_a : N \rightarrow N$  по следния начин:  $f_a(n) = a_n$  за  $\forall n \in N$
- Тези функции са “повече” от естествените числа
- А програмите колко са?

# Разрешими и неразрешими проблемы

- Разрешими и неразрешими проблемами. СТОП проблем

Благодаря за вниманието!