



ПЪРВИ СЪПКИ С АРДУИНО

**ЗАНЯТИЕ №6 КОМУНИКАЦИЯ НА АРДУИНО
С КОМПЮТЪР**

ВЕНЦИСЛАВ НАЧЕВ





КАКВО ЩЕ ПРАВИМ ДНЕС?

- ❖ Разработка на графични приложения с “Processing”
- ❖ Серийна комуникация с Ардуино
- ❖ Връзка на Ардуино с Processing приложение

КАКВО Е PROCESSING?

- ❖ Java "диалект" създаден за разработка на графични приложения;
- ❖ Processing е безплатен софтуер – Windows, Linux, Mac OS...
- ❖ Разработен през 2001г. в **M**assachusetts **I**nstitute of **T**echnology – MIT.



ЗАЩО PROCESSING?

- ❖ Много, много лесен за научаване и работа;
- ❖ Безплатен;
- ❖ Има своя среда за разработка Processing IDE – **I**ntegrated **D**evelopment **E**nvironment;
- ❖ С големи възможности: позволява разработка от много прости до много сложни приложения;

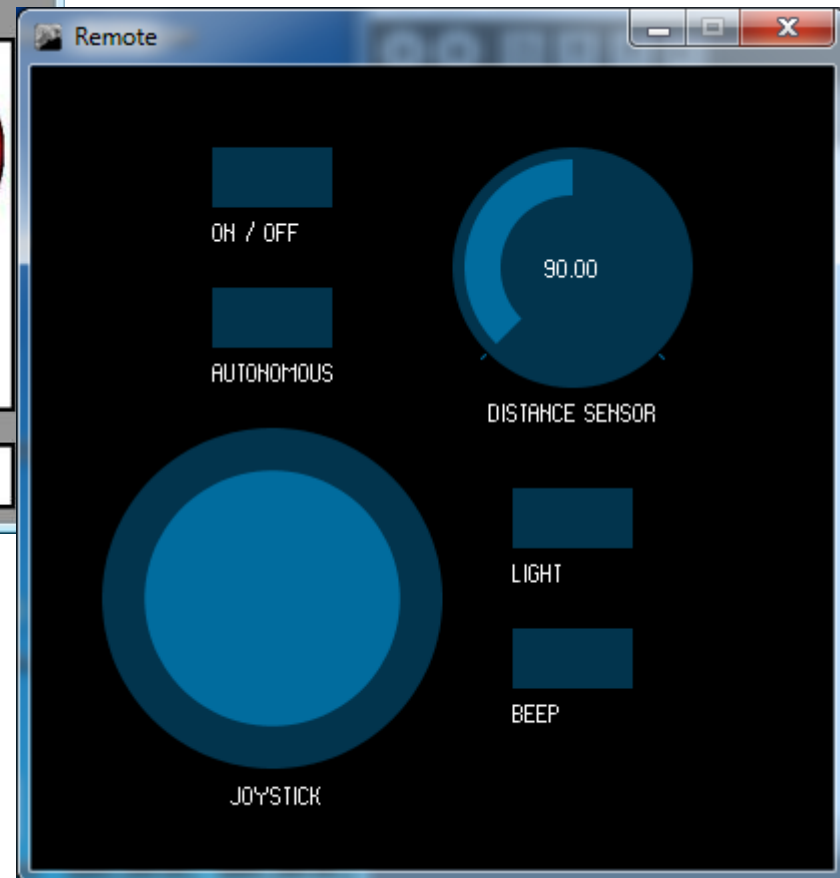
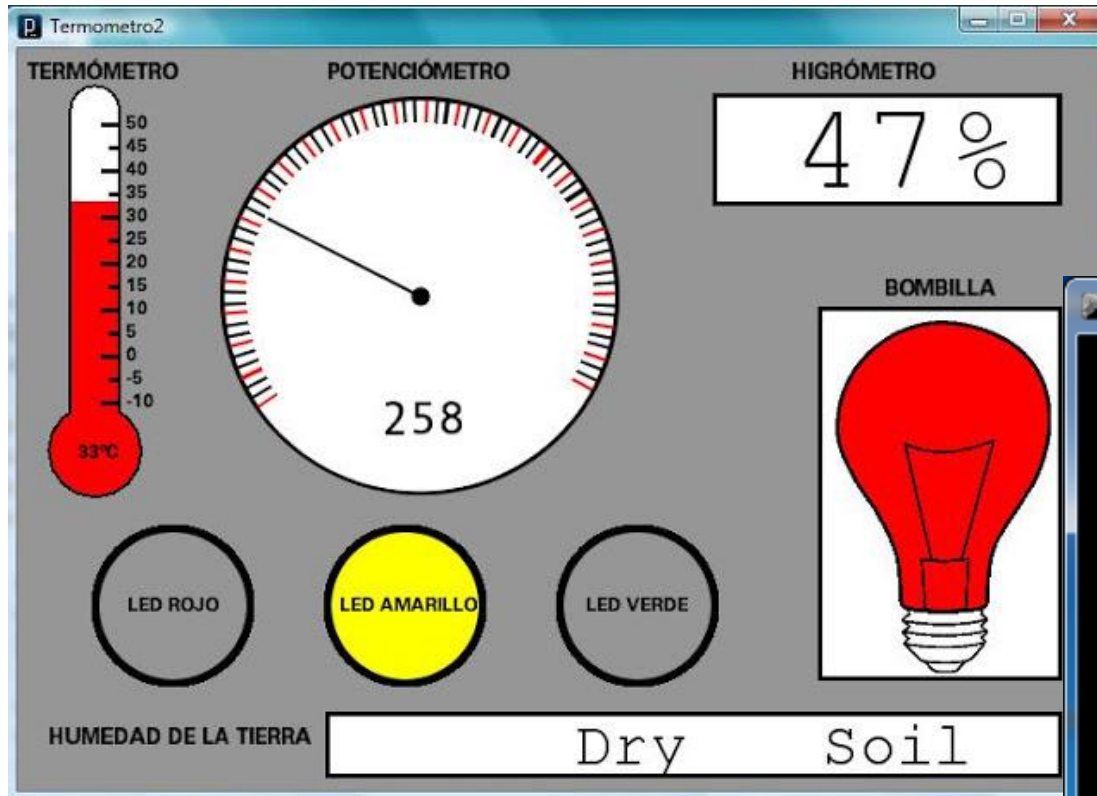


ЗАЩО PROCESSING?

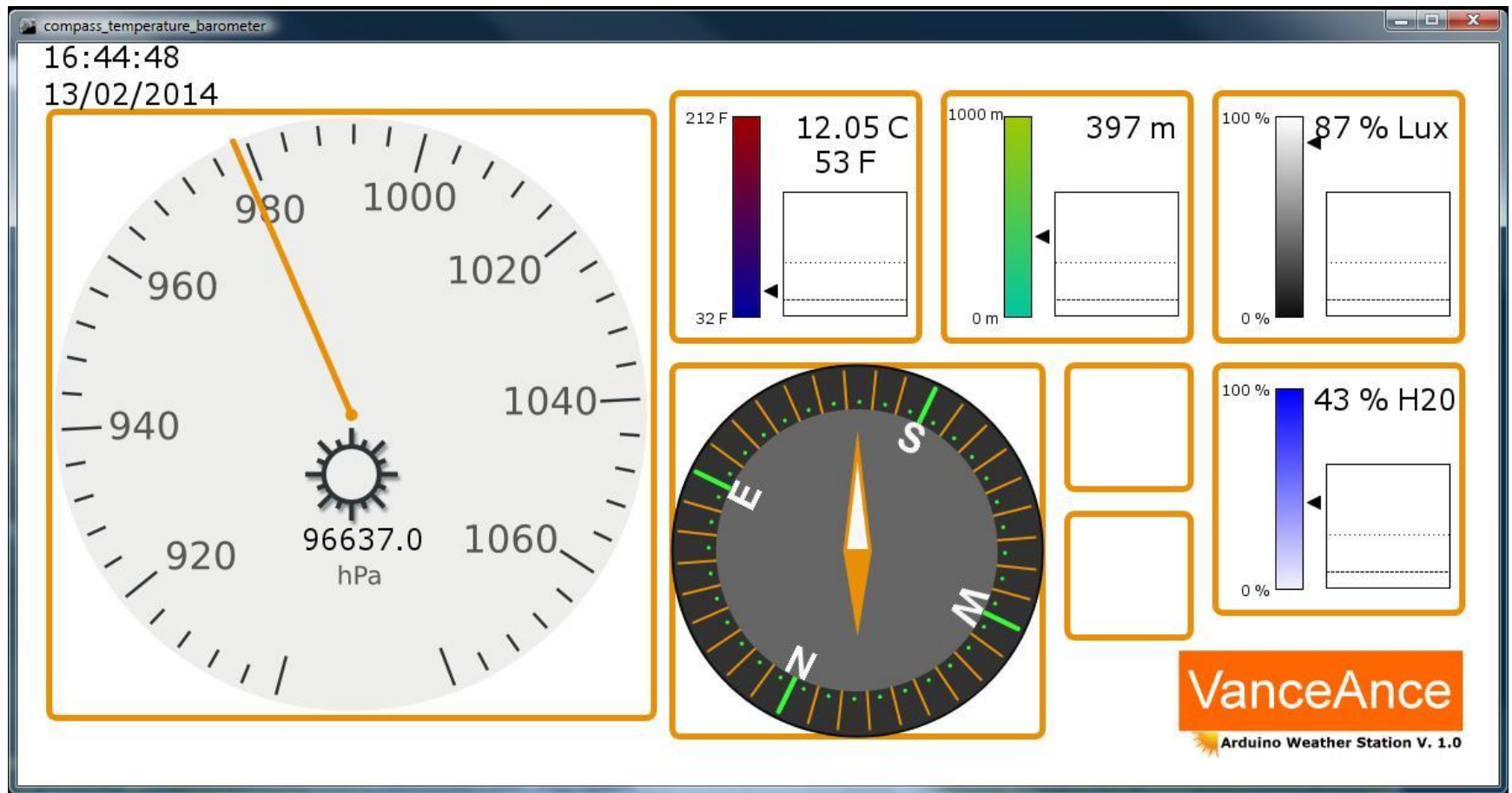
- ❖ Възможност да се използват Java библиотеки;
- ❖ Позволява различни подходи на програмиране: основен, процедурен, обектно-ориентиран;
- ❖ Възможност за разработка на мобилни приложения(Android) и приложения за изпълнение в браузър;
- ❖ Лесна комуникация с хардуерни устройства и микроконтролери;



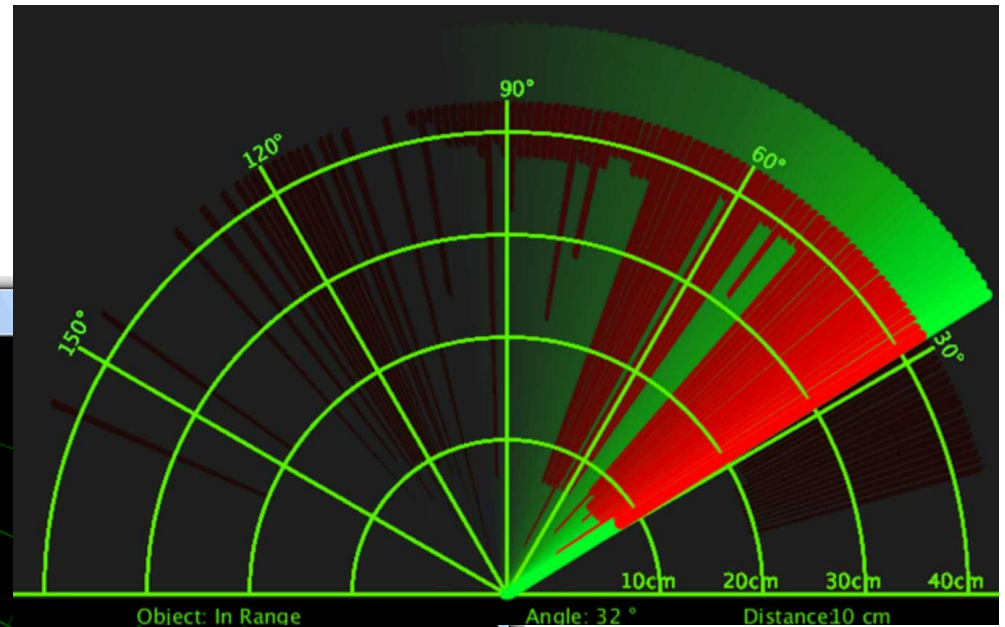
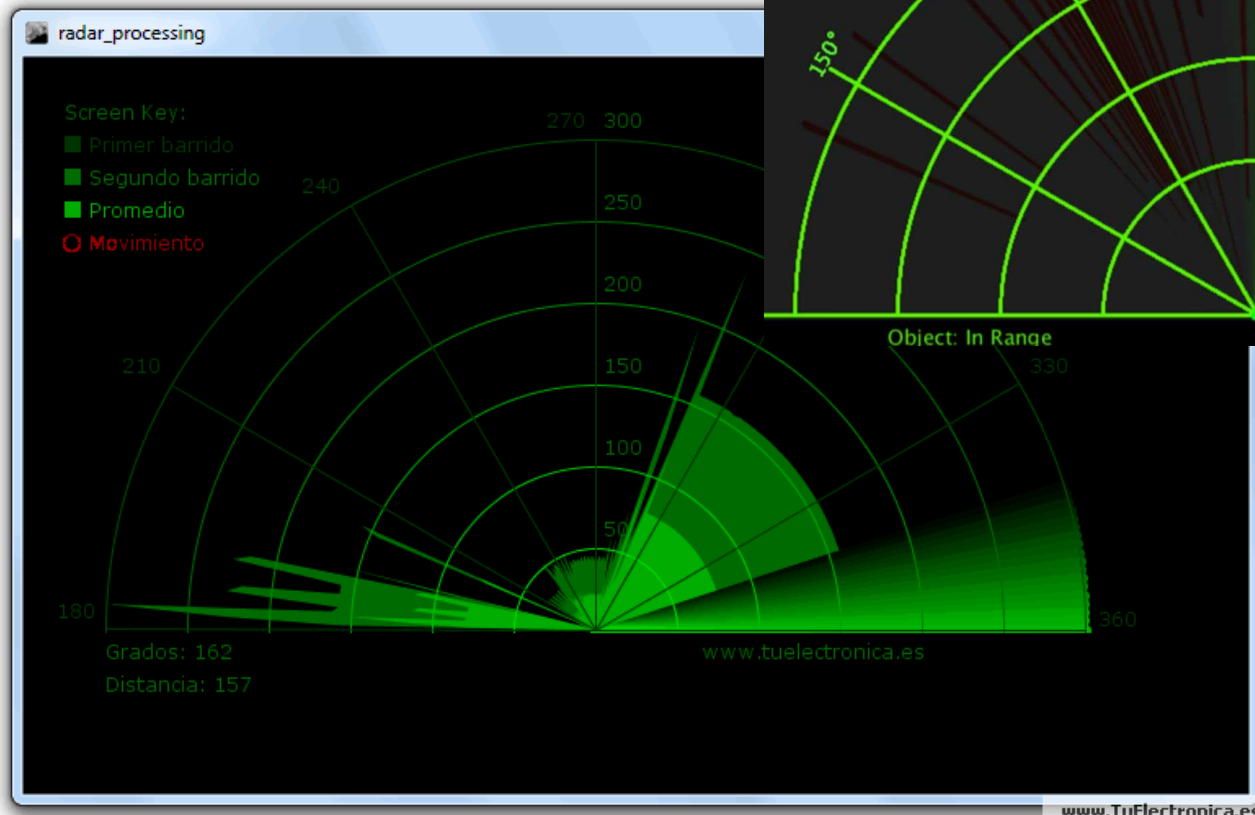
PROCESSING



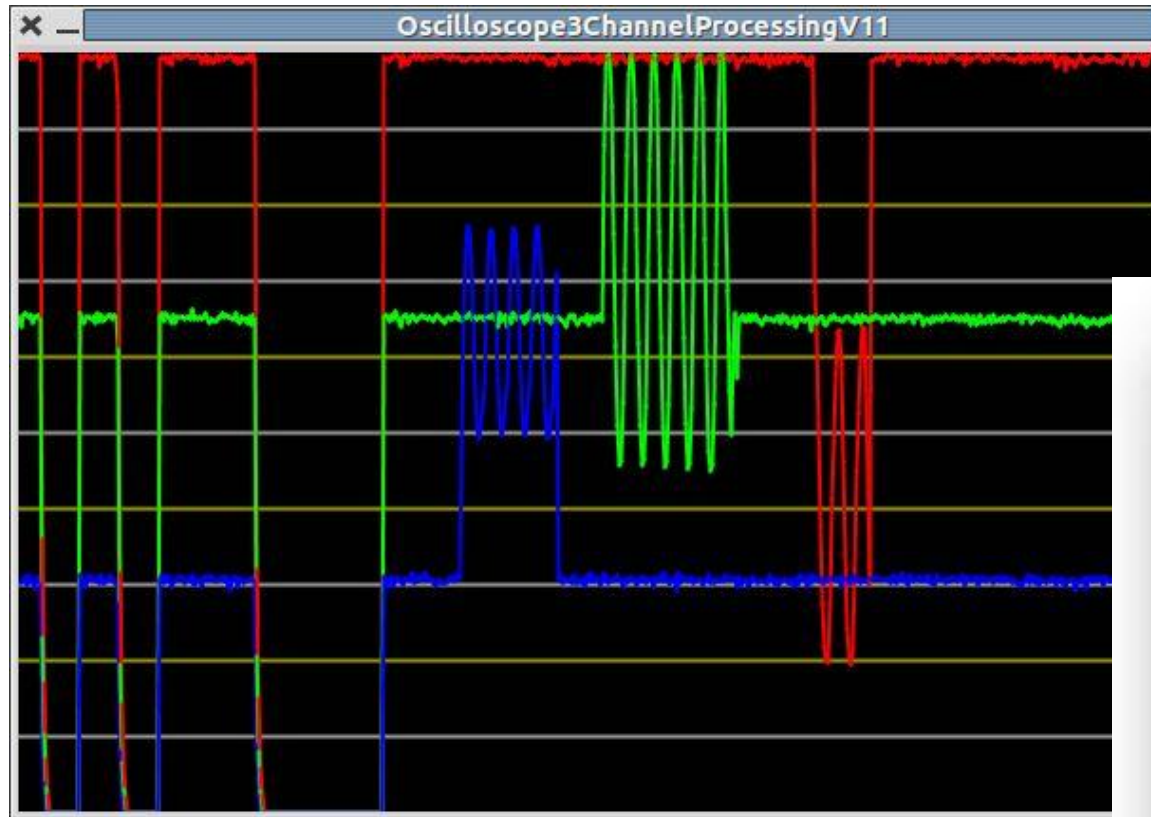
PROCESSING



PROCESSING



PROCESSING



ИНСТАЛИРАНЕ И КОНФИГУРИРАНЕ

1. СВАЛИ <https://processing.org/download/>

Download Processing. Processing is available for Linux, Mac OS X, and Windows. Select your choice to download the software below.



3.2.3 (7 November 2016)

Windows 64-bit
Windows 32-bit

Linux 64-bit
Linux 32-bit
Linux ARMv6hf

Mac OS X

2. РЕЗАРХИВИРАЙ



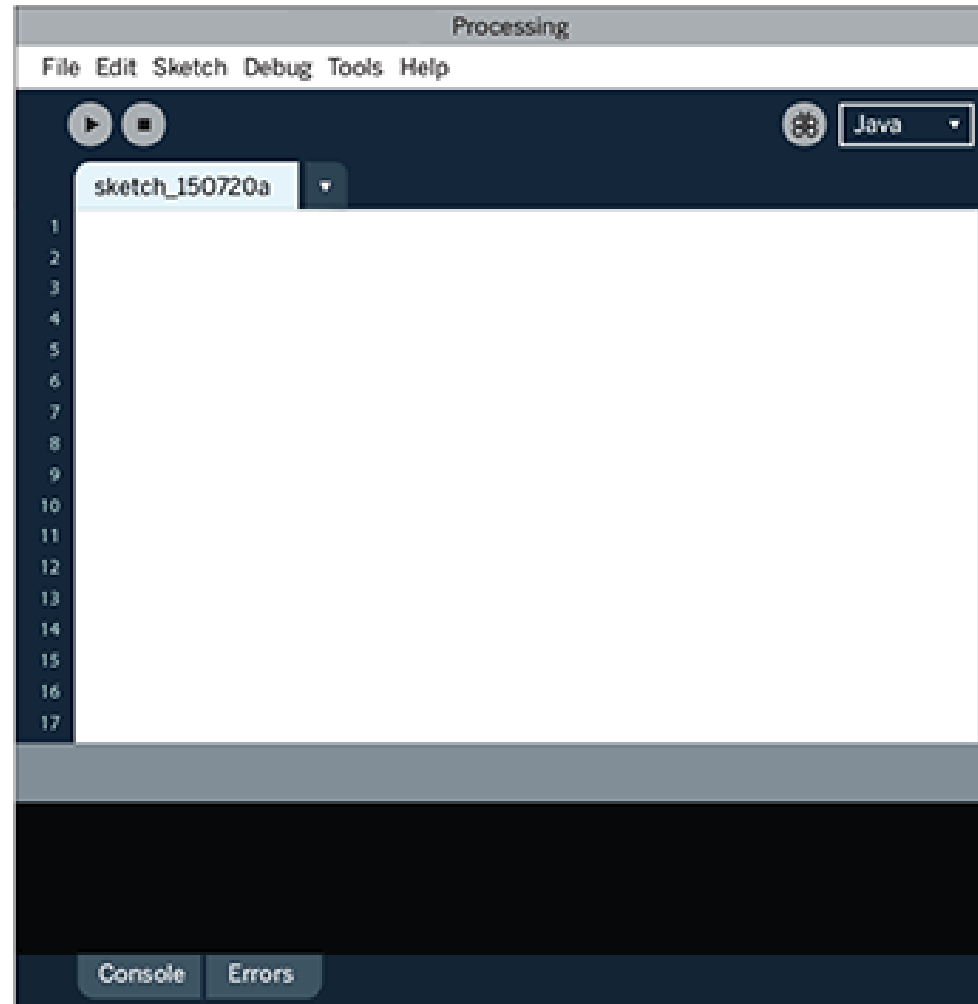
3. ГОТОВО



PROCESSING IDE



Display Window



Menu

Toolbar

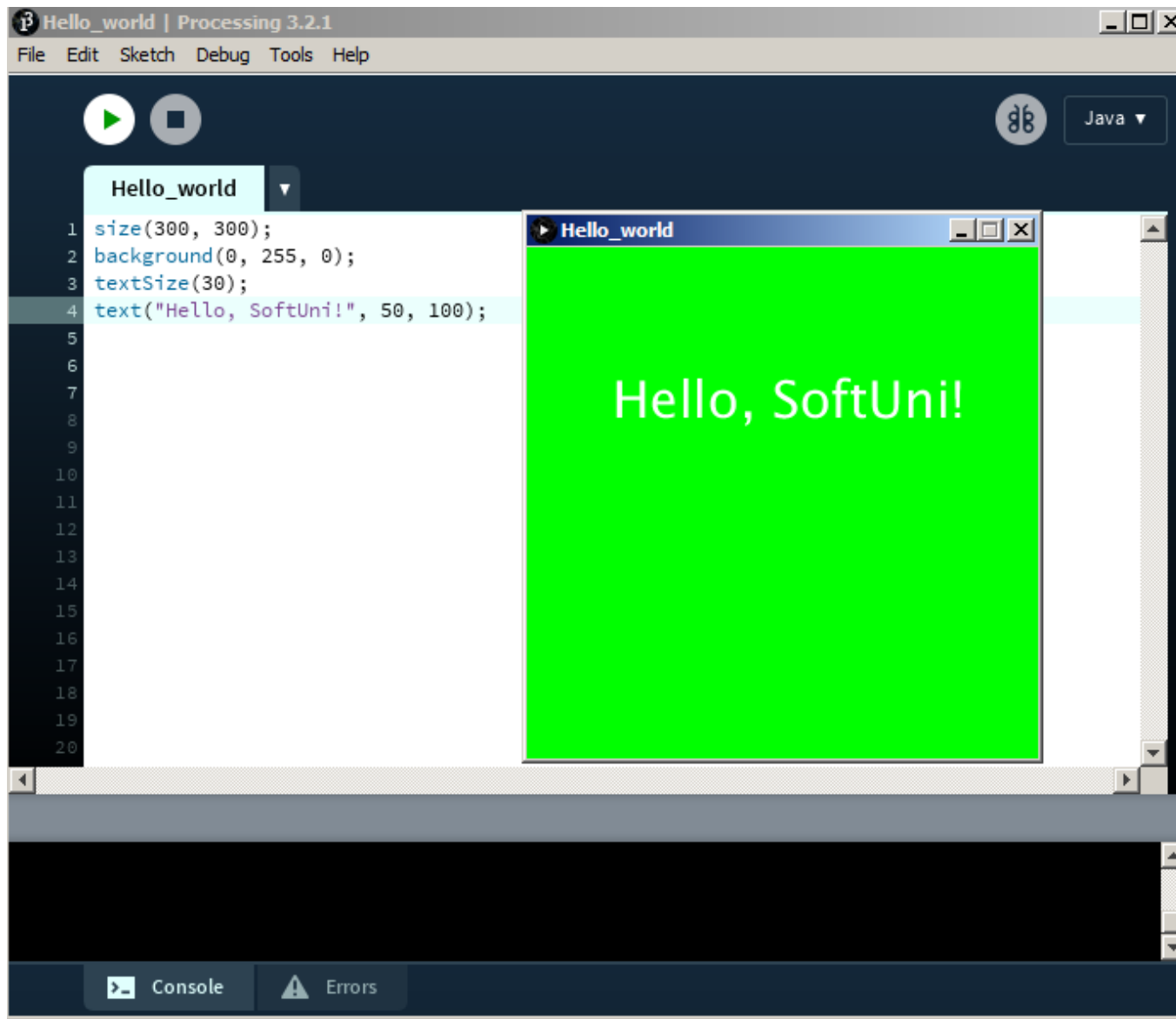
Tabs

Text Editor

Message Area

Console

“HELLO, WORLD!”





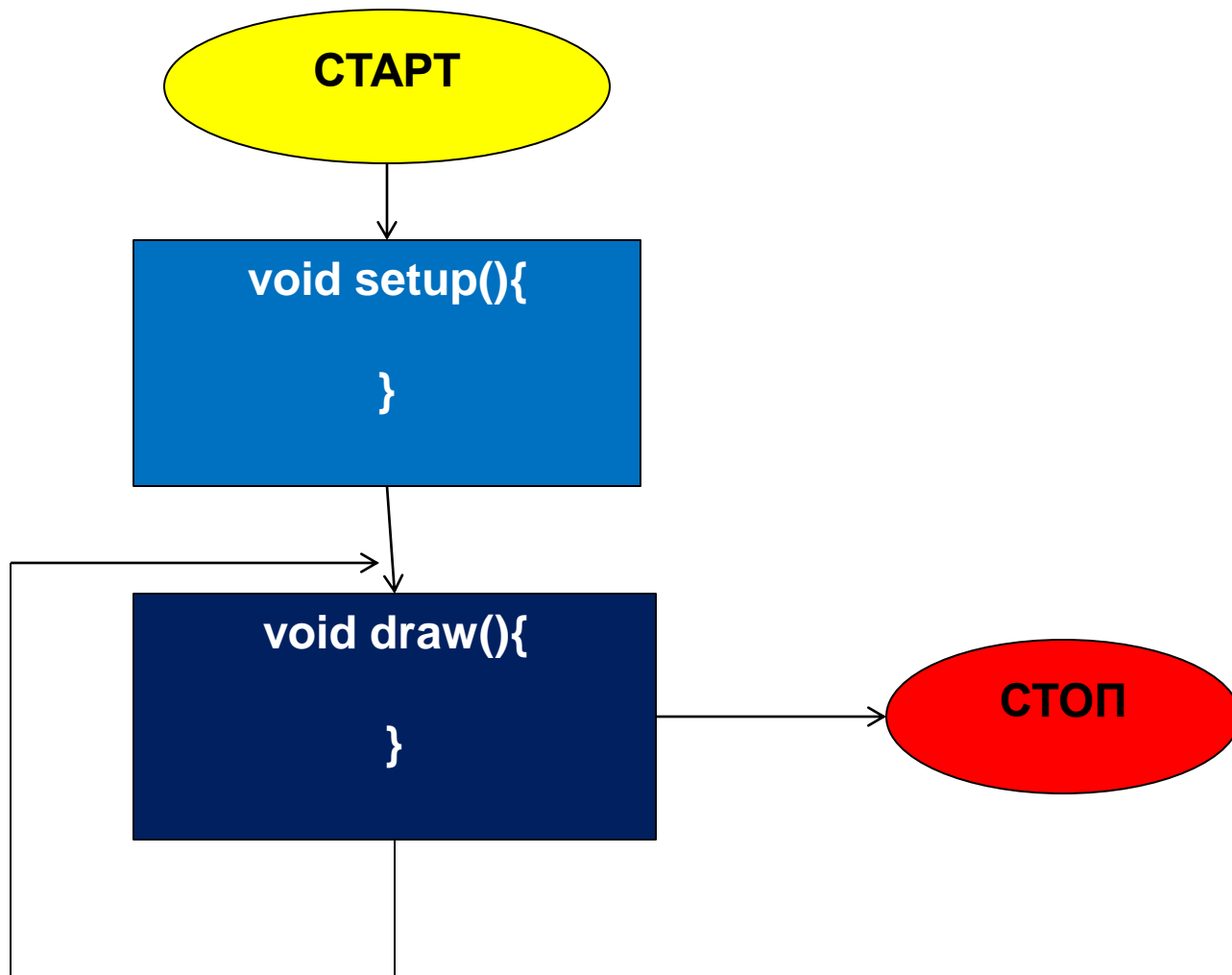
“HELLO, WORLD!”

```
size(300, 300);  
background(0, 255, 0);  
textSize(30);  
text("Hello, SoftUni!", 50, 100);
```

СТРУКТУРА НА PROCESSING ПРОГРАМА

```
void setup(){  
    // изпълнява се веднъж  
}  
  
void draw(){  
    // изпълнява се докато не спрем програмата  
}
```

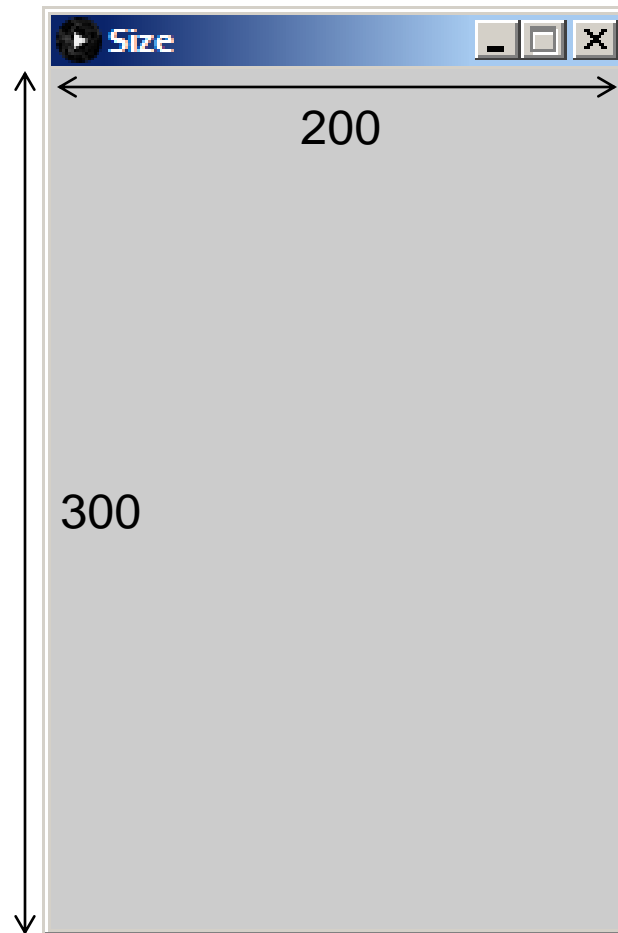
СТРУКТУРА НА PROCESSING ПРОГРАМА



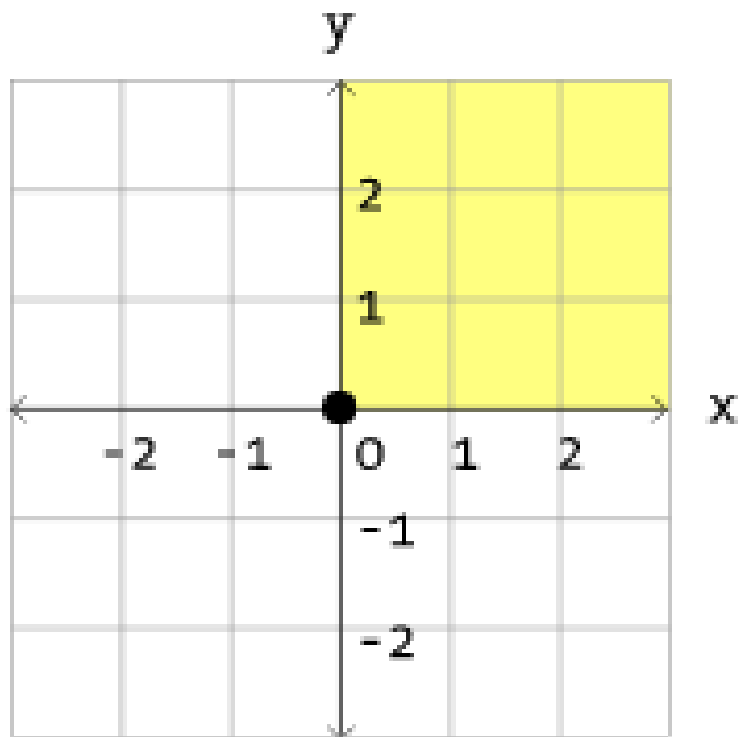
ОСНОВНИ ФУНКЦИИ

size(широчина, височина);

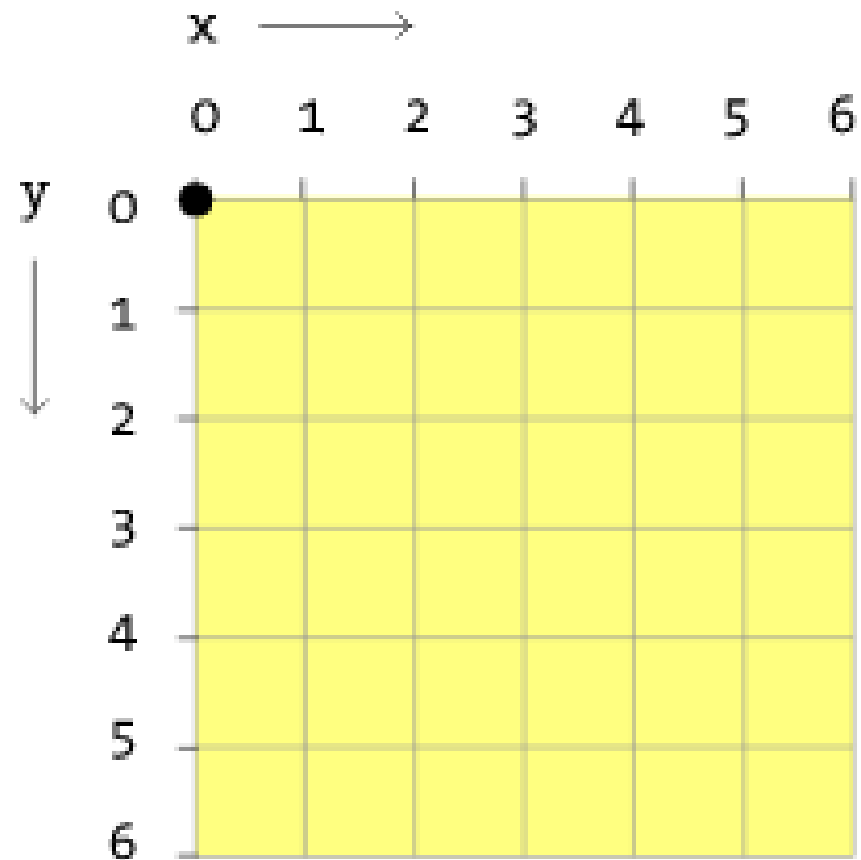
```
void setup(){  
  size(200, 300);  
}  
  
void draw(){  
}
```



КООРДИНАТНА СИСТЕМА



в училище



в Processing

ОСНОВНИ ФИГУРИ



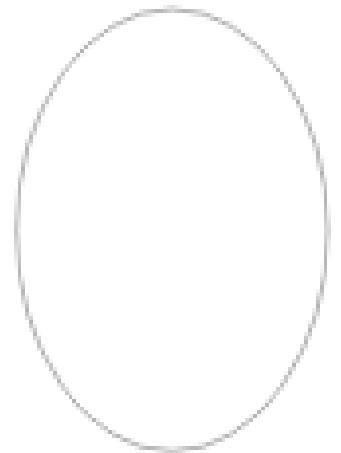
Point



Line



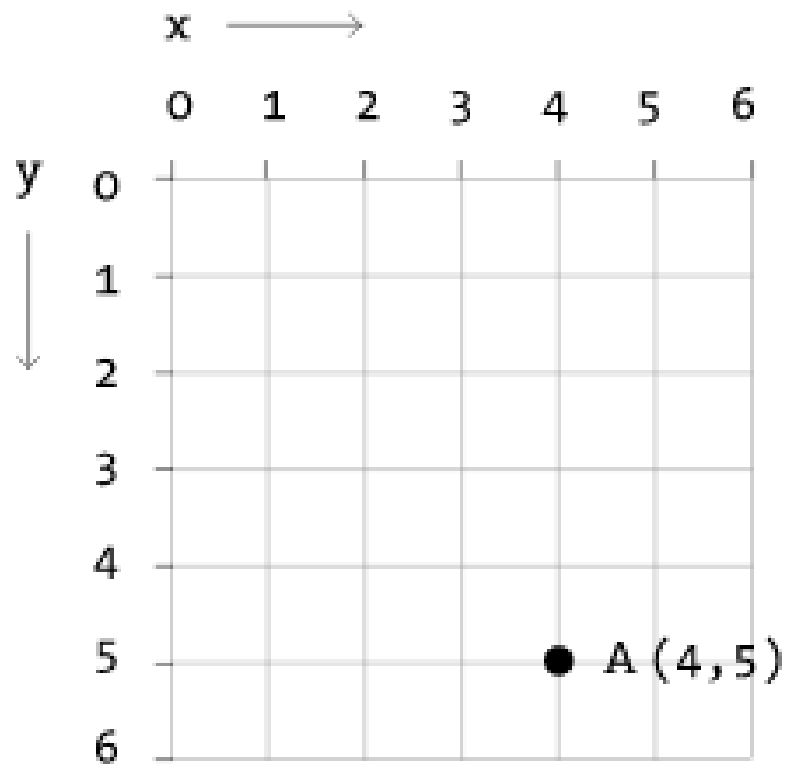
Rectangle



Ellipse

ТОЧКА

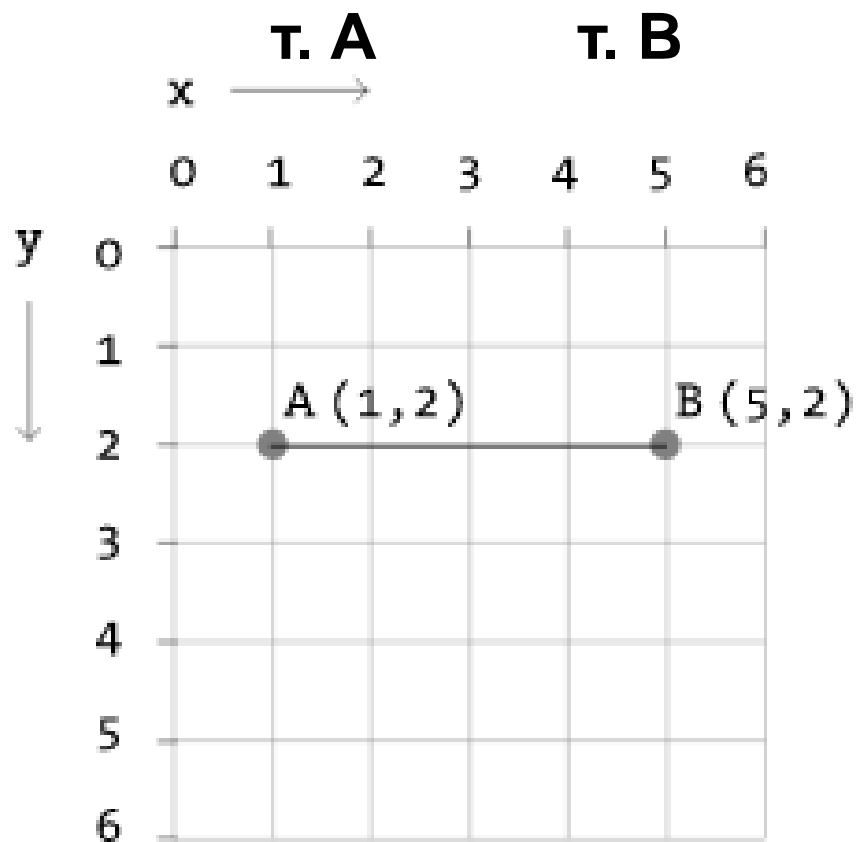
point(x, y);



Example: A (4, 5) ;

ЛИНИЯ

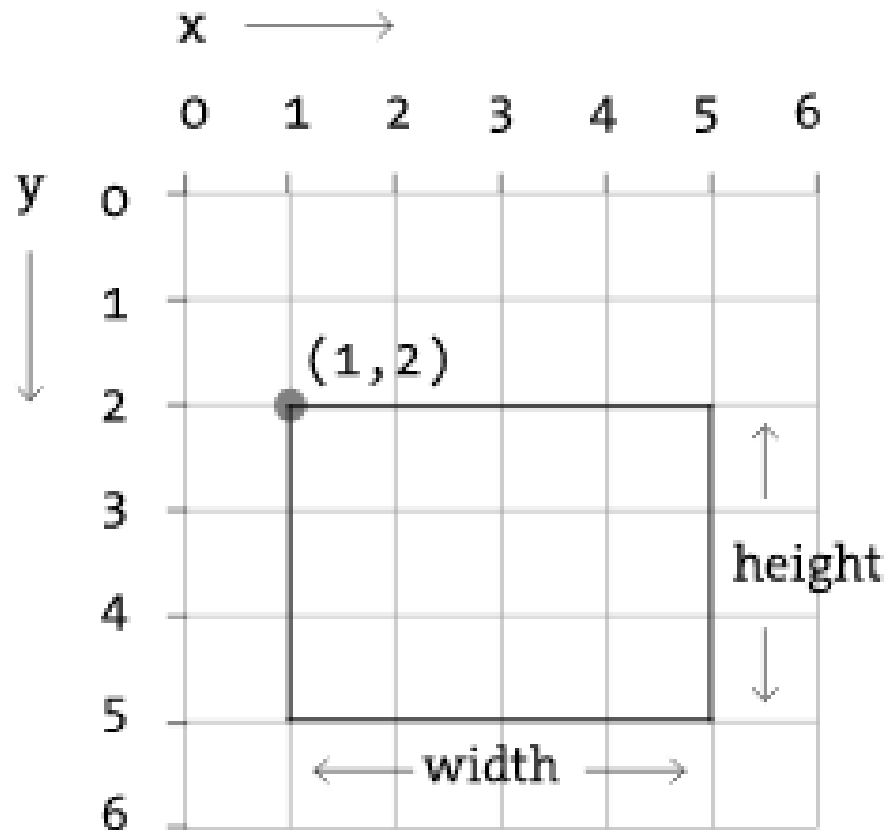
line(x1, y1, x2, y2);



Example: line (1, 2, 5, 2) ;

ПРАВОЪГЪЛНИК

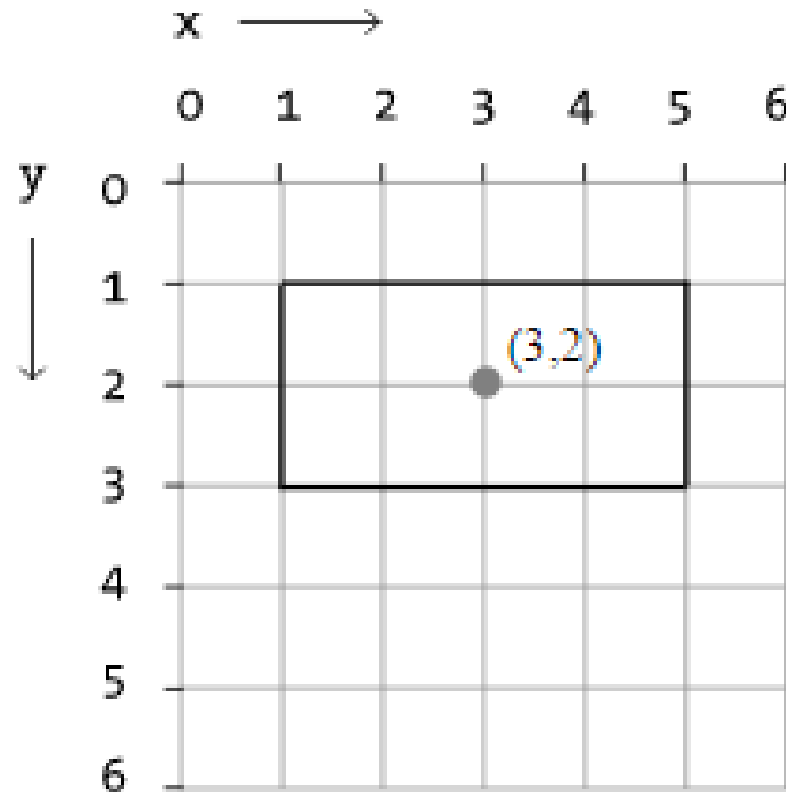
rect(*x, y, ширина, височина*);



Example: `rect (1, 2, 4, 3) ;`

ПРАВОЪГЪЛНИК

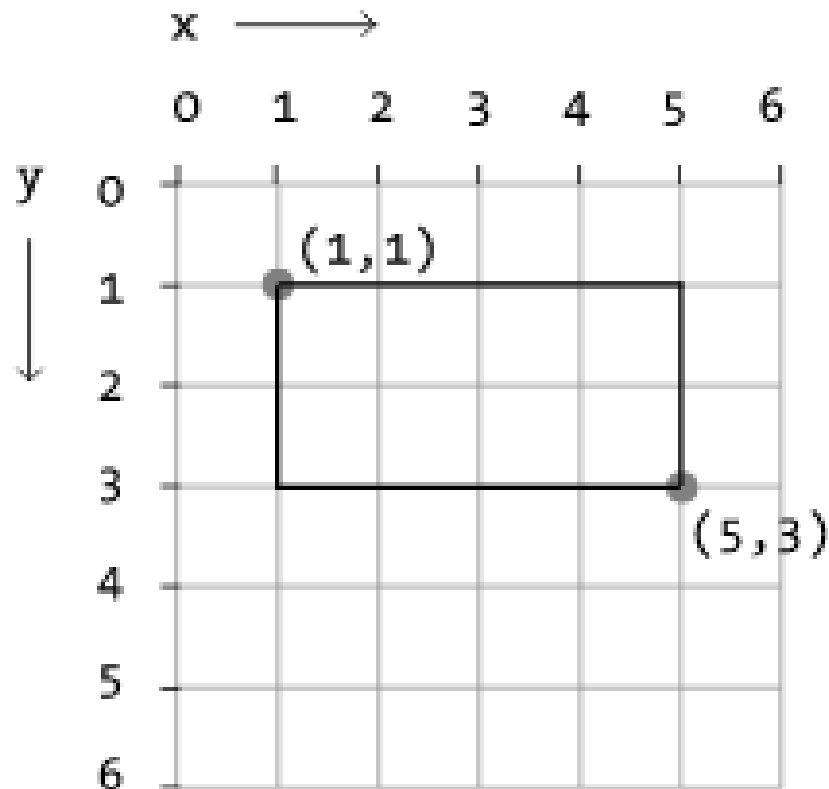
rectMode(CENTER);



Example: `rectMode (CENTER) ;`
`rect (3 , 2 , 4 , 2) ;`

ПРАВОЪГЪЛНИК

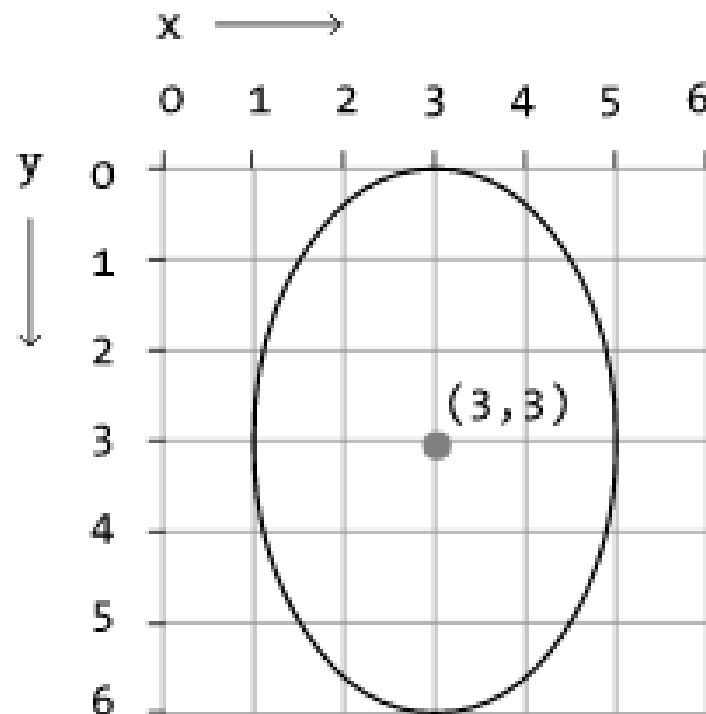
rectMode(CORNERS);



Example: `rectMode (CORNERS) ;`
`rect (1, 1, 5, 3) ;`

ЕЛИПСА

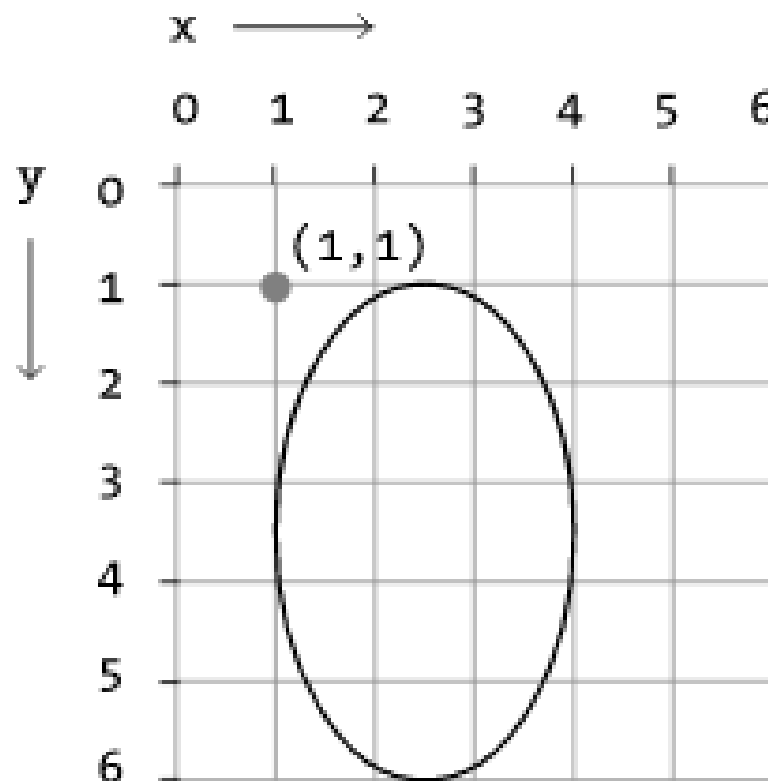
ellipse(*x*, *y*, *диаметър по x*, *диаметър по y*);



Example: `ellipseMode (CENTER);`
`ellipse (3, 3, 4, 6);`

ЕЛИПСА

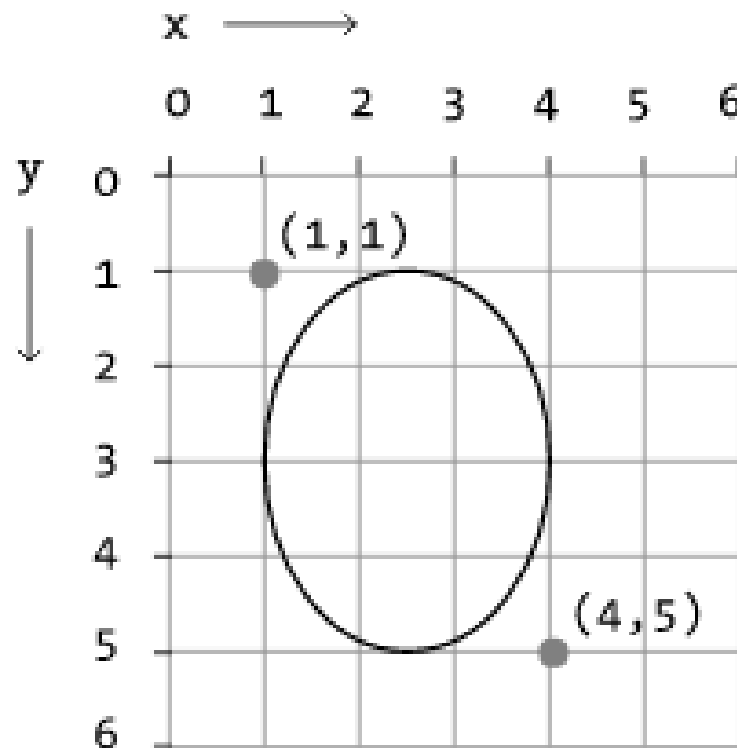
ellipseMode(CORNER);



Example: `ellipseMode (CORNER) ;`
`ellipse (1, 1, 3, 5) ;`

ЕЛИПСА

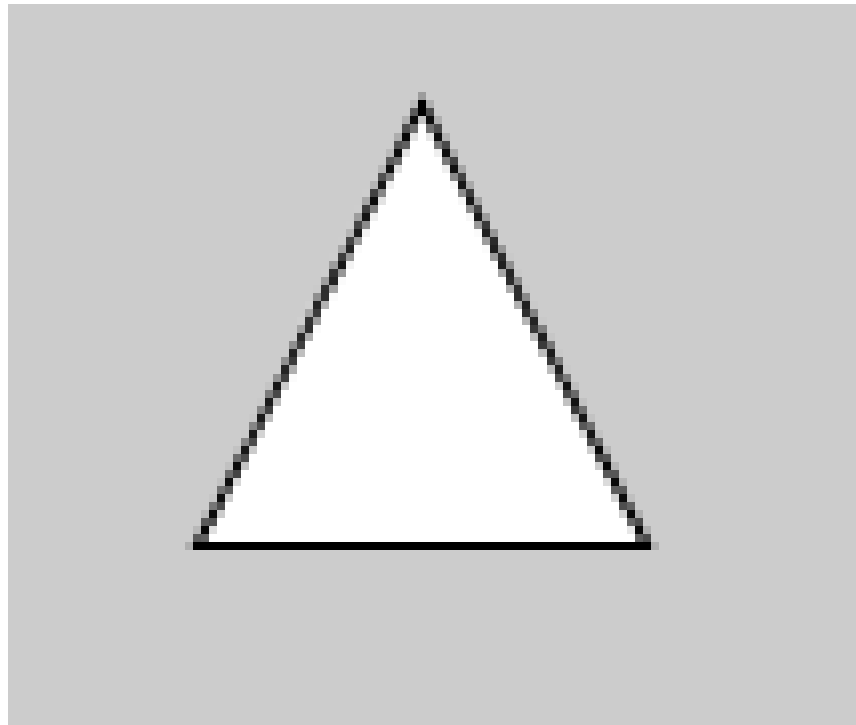
ellipseMode(CORNERS);



Example: `ellipseMode (CORNERS) ;`
`ellipse (1, 1, 4, 5) ;`

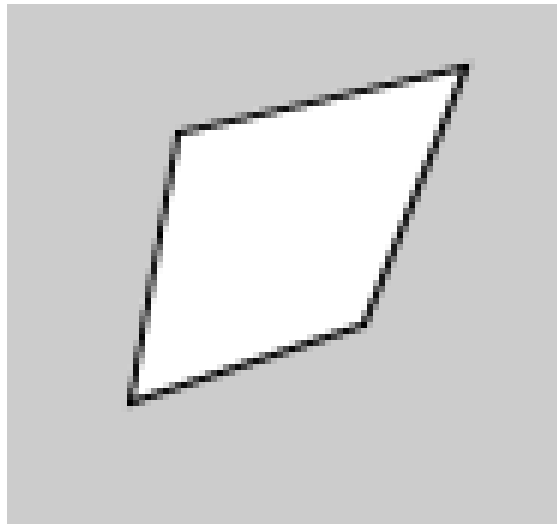
ТРИЪГЪЛНИК

triangle(x1, y1, x2, y2, x3, y3);



ЧЕТИРИЪГЪЛНИК

quad(x1, y1, x2, y2, x3, y3, x4, y4);

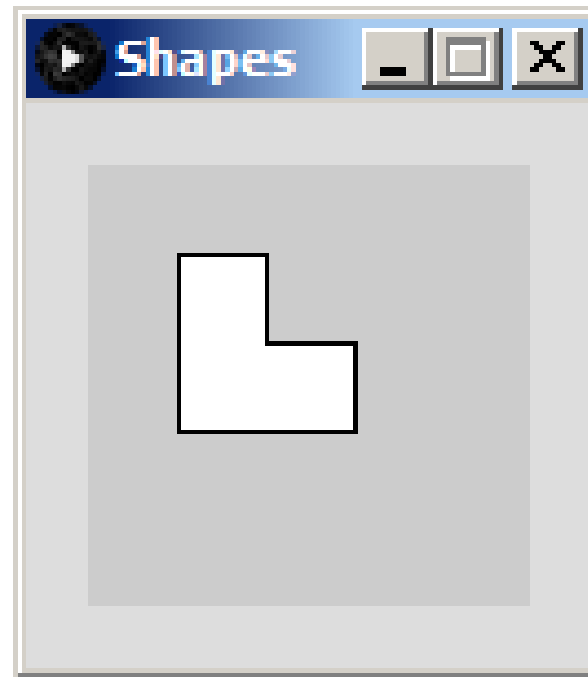


ПРОИЗВОЛНА ФИГУРА

```
beginShape();  
vertex( $x1$ ,  $y1$ );  
vertex( $x2$ ,  $y2$ );  
vertex( $x3$ ,  $y3$ );  
  
.....  
vertex( $xn$ ,  $yn$ );  
endShape(CLOSE);
```

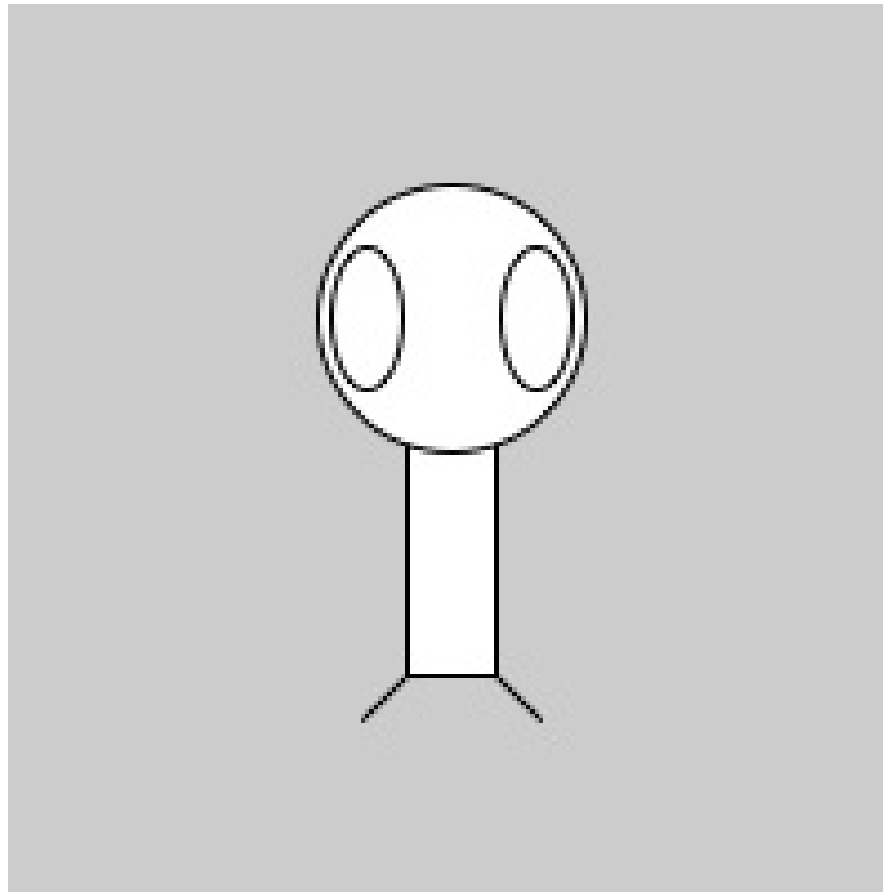
ПРИМЕР:

```
void setup(){  
  size(100, 100);  
  beginShape();  
  vertex(20, 20);  
  vertex(40, 20);  
  vertex(40, 40);  
  vertex(60, 40);  
  vertex(60, 60);  
  vertex(20, 60);  
  endShape(CLOSE);  
}
```



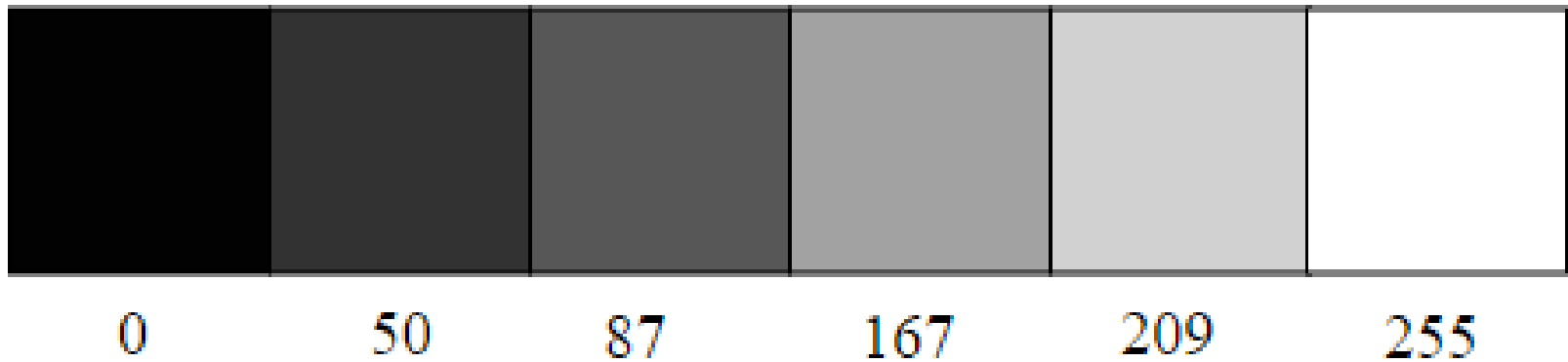
ЗАДАЧА 6.1

Да се нарисува с код следната картинка:



ДОБАВЯНЕ НА ЦВЯТ

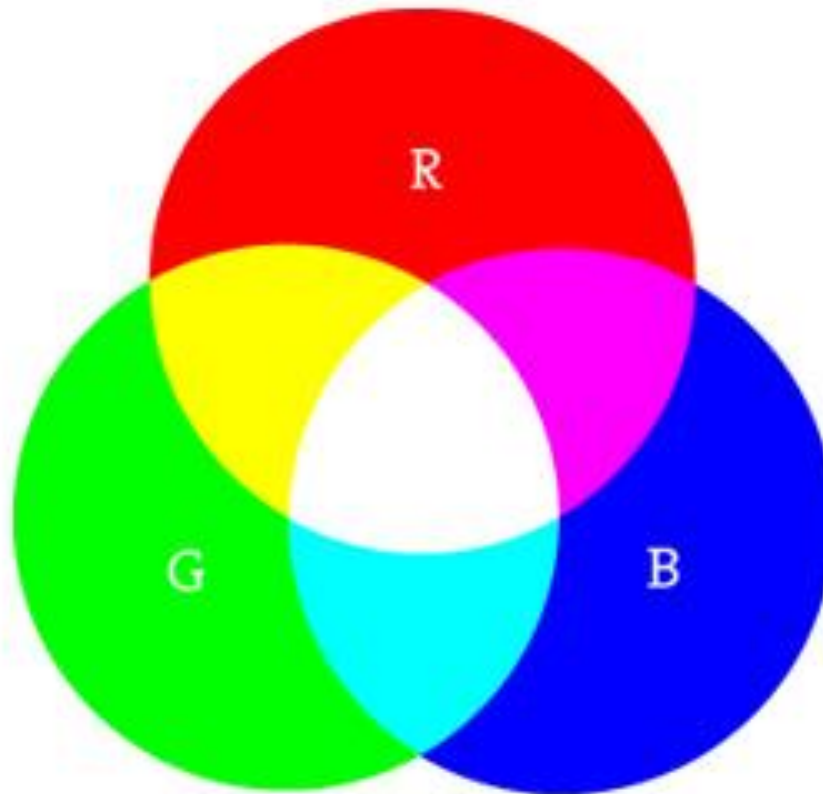
background(0 - 255);



Red – Green – Blue

background(0 – 255, 0 – 255, 0 – 255);

R G B



ДОБАВЯНЕ НА ЦВЯТ

fill(0 – 255, 0 – 255, 0 – 255);

-> оцветяване на фигури

stroke(0 – 255, 0 – 255, 0 – 255);

-> оцветяване на контури

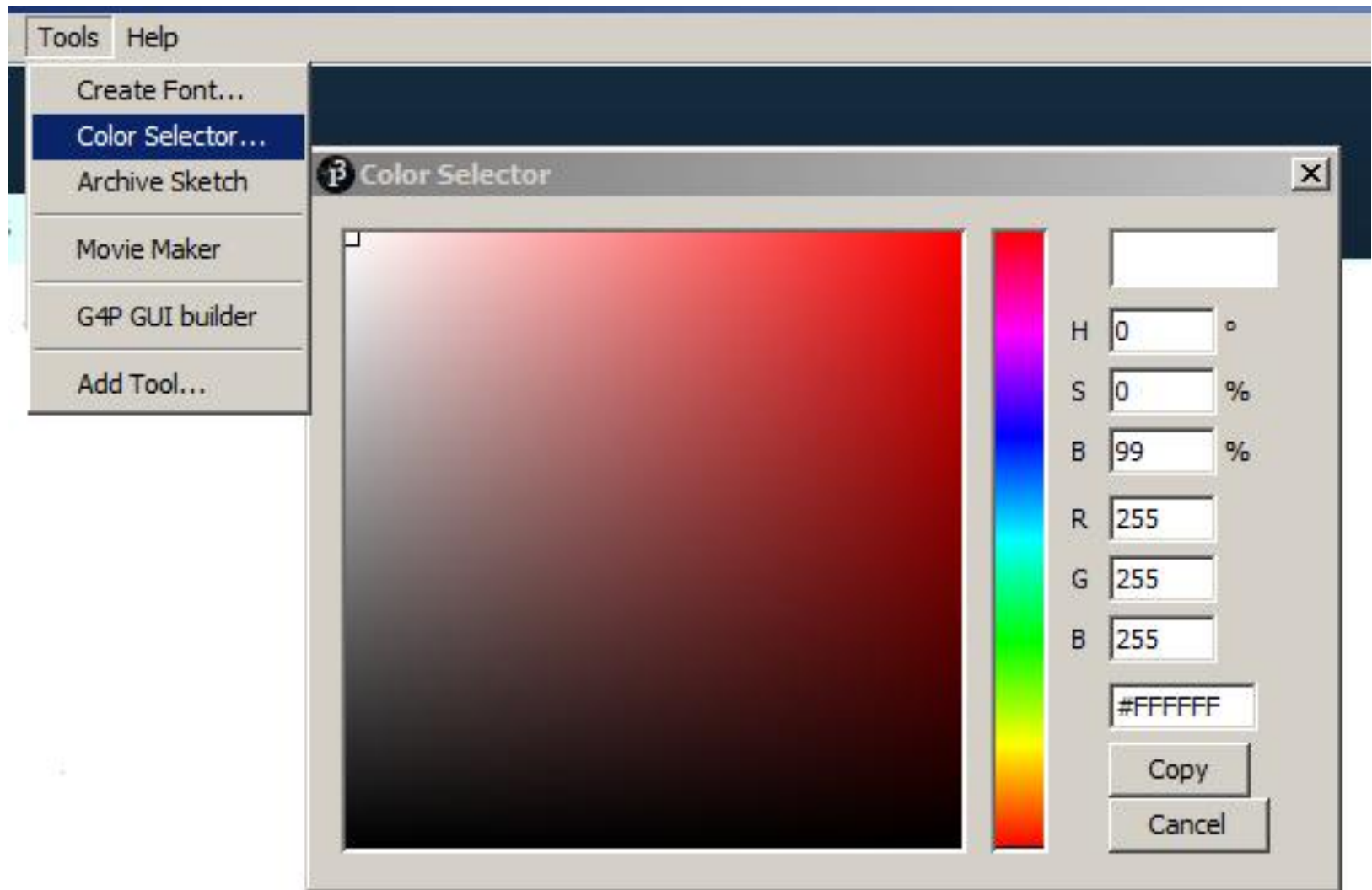
noFill();

-> без оцветяване на фигури

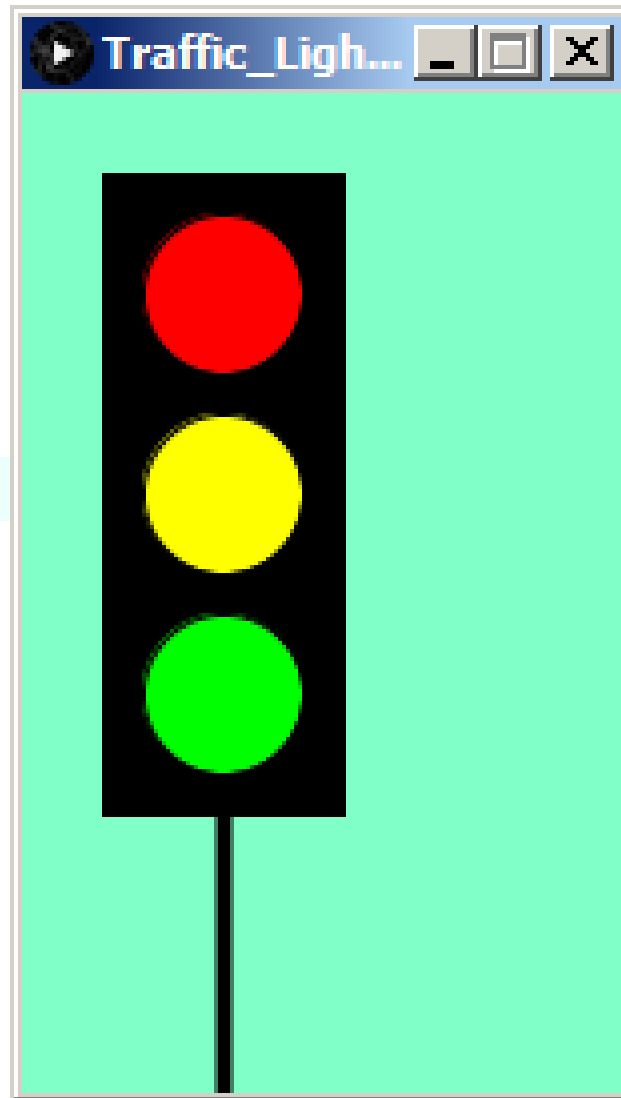
noStroke();

-> без оцветяване на контури

COLOR SELECTOR



ЗАДАЧА 6.2

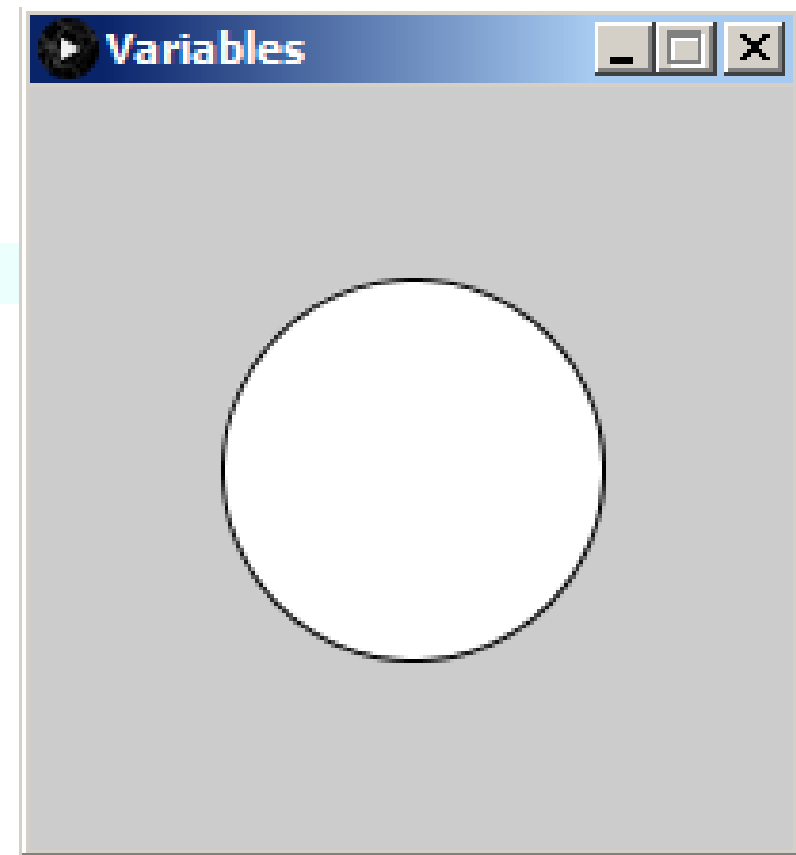


ТИПОВЕ ДАННИ И ПРОМЕНЛИВИ

- ❖ **int** – цяло число;
- ❖ **float** – число с десетична запетая;
- ❖ **char** – СИМВОЛ;
- ❖ **String** – текст.
- ❖ **boolean** – логическа стойност – **true/false**.

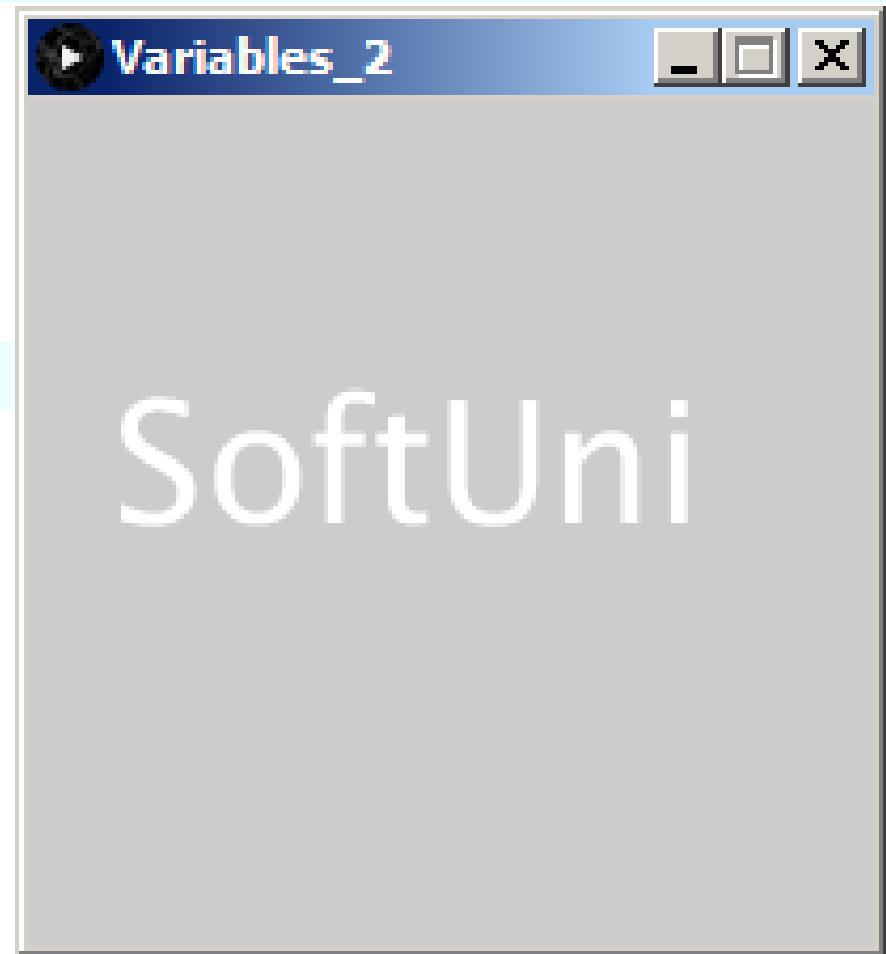
ПРИМЕР - INT

```
void setup(){  
  size(200, 200);  
  int d = 100;  
  ellipse(width / 2, height / 2, d, d);  
}  
  
void draw(){  
}
```



ПРИМЕР - STRING

```
void setup(){  
  size(200, 200);  
  String txt = "SoftUni";  
  textSize(40);  
  text(txt, 20, 100);  
}  
  
void draw(){  
  
}
```

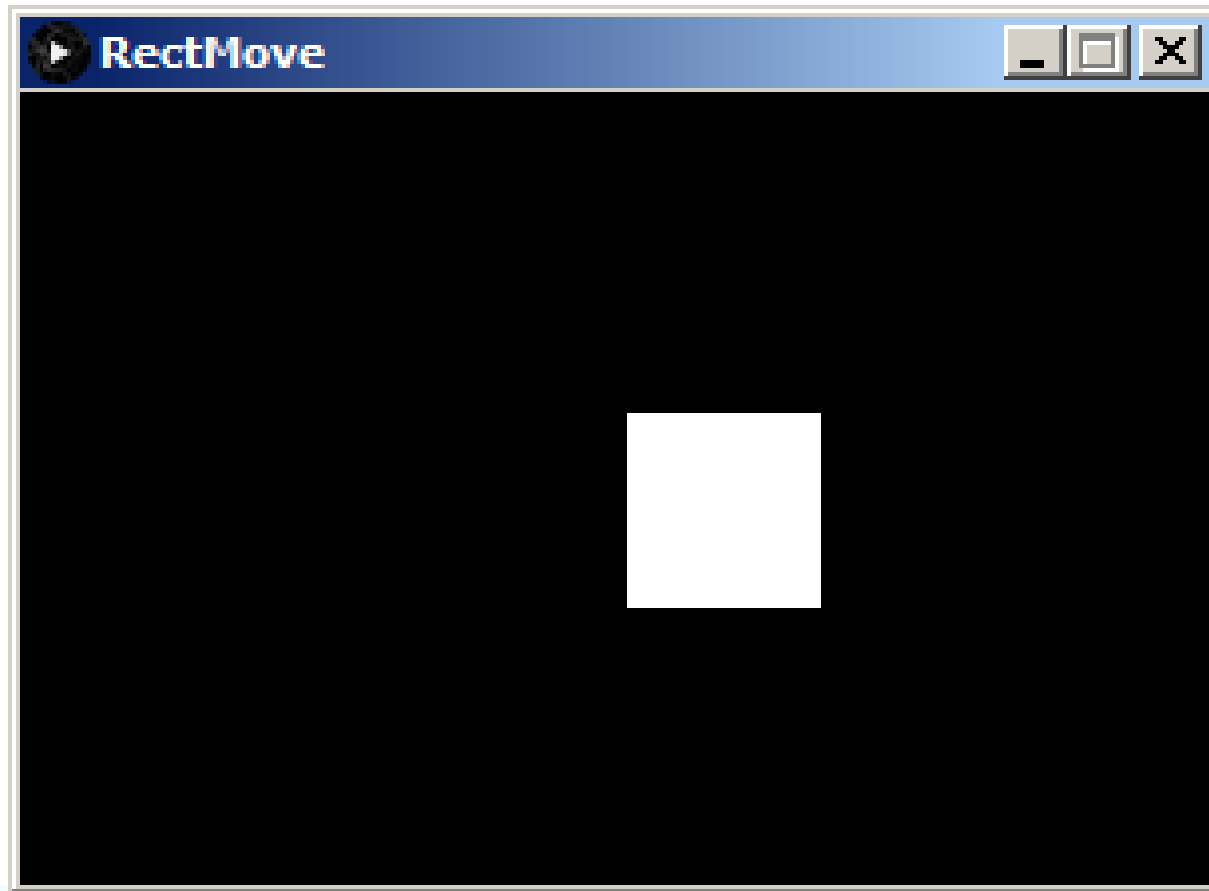


ПРИМЕР – MOVE RECTANGLE

```
int x, y;

void setup(){
  size(300, 200);
  x = 0;
  y = 80;
}

void draw(){
  background(0);
  x++;
  rect(x, y, 50, 50);
}
```

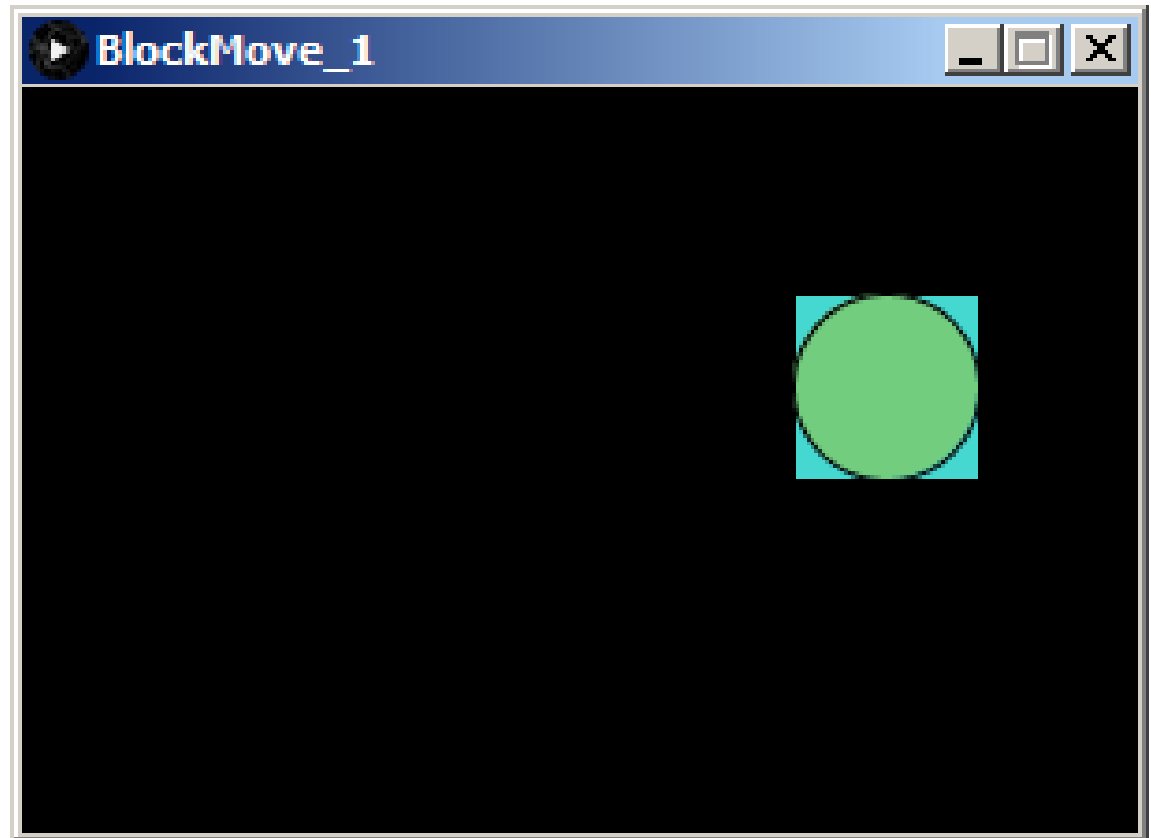


ПРИМЕР – MOVE OBJECT

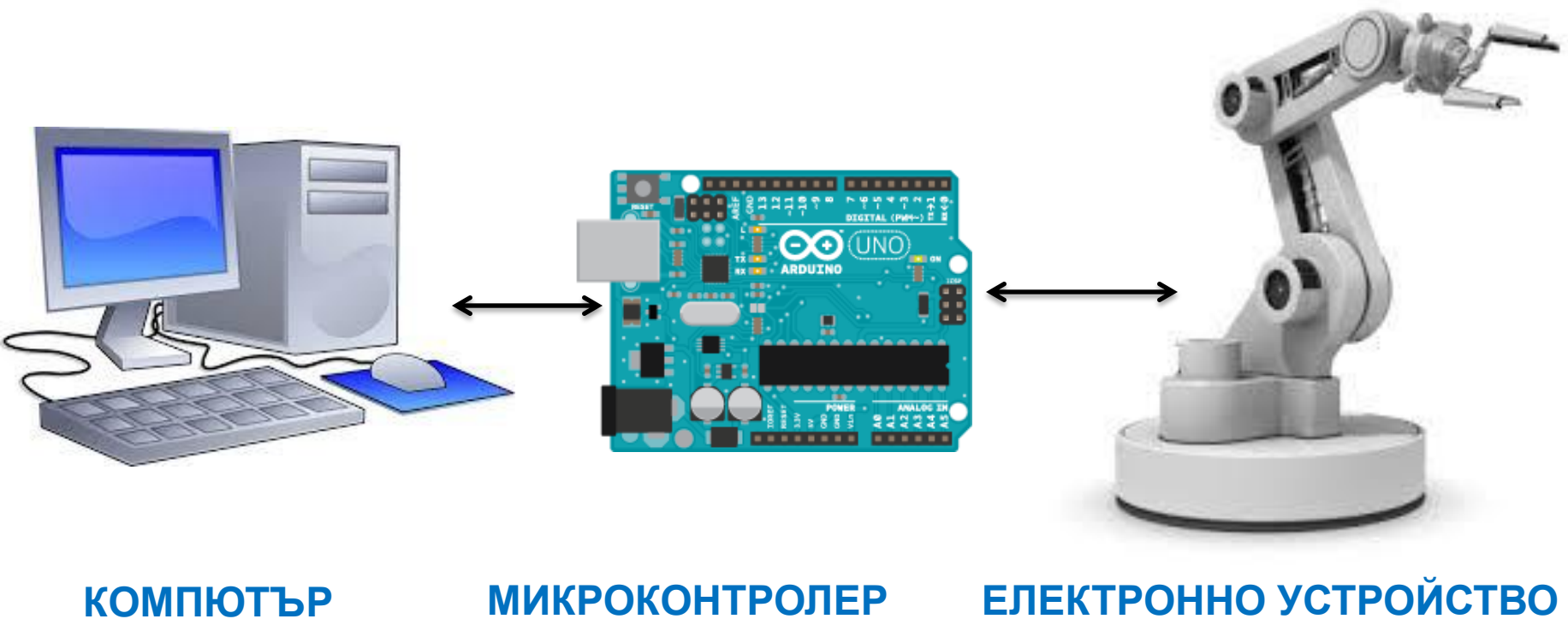
```
int x, y;

void setup(){
  size(300, 200);
  x = 300;
  y = 80;
}

void draw(){
  background(0);
  x--;
  fill(#44D8D0);
  rectMode(CENTER);
  rect(x, y, 50, 50);
  fill(#72CE7E);
  ellipse(x, y, 50, 50);
}
```



ВРЪЗКА НА PROCESSING С МИКРОКОНТРОЛЕР



PROCESSING

- ❖ Клас за серийна комуникация **Serial**.
- ❖ Метод за четене: **read()**.
- ❖ Метод за писане: **write()**.



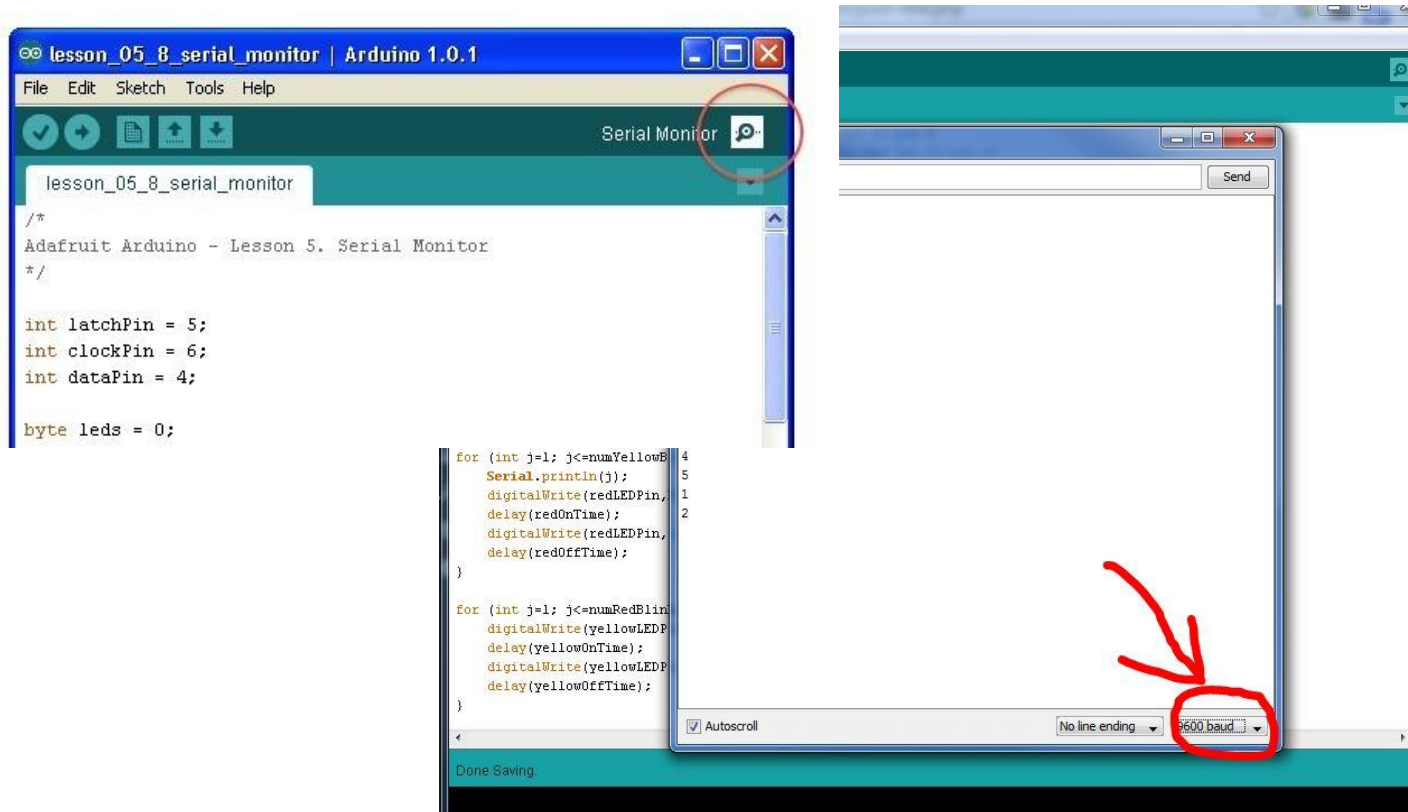
ARDUINO

- ❖ Клас за серийна комуникация **Serial**.
- ❖ Метод за четене: **read()**.
- ❖ Метод за писане: **write()**.

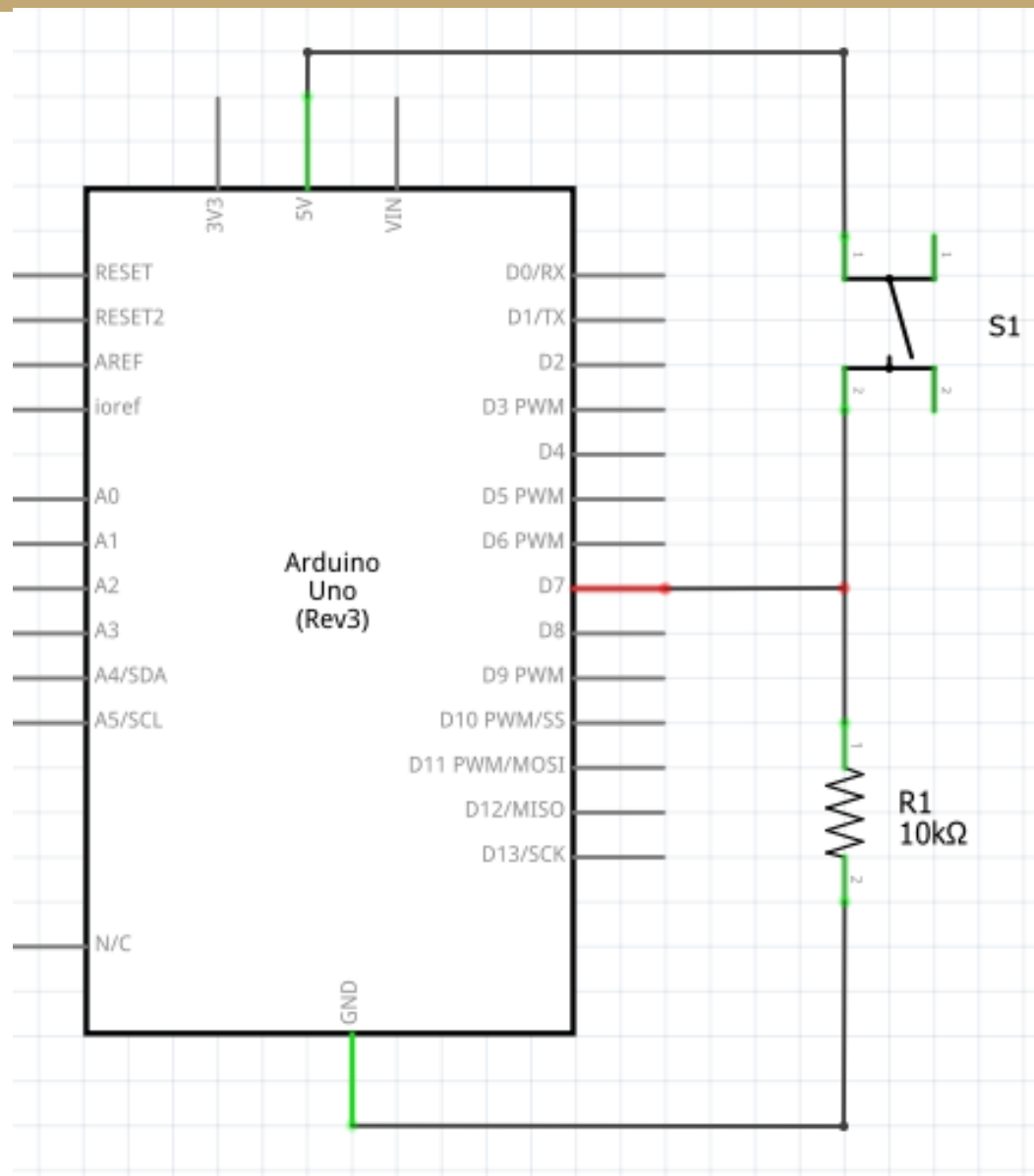


ЗАДАЧА 6.3

❖ Серийна комуникация: Ардуино -> Компютър



ЕЛЕКТРИЧЕСКА СХЕМА



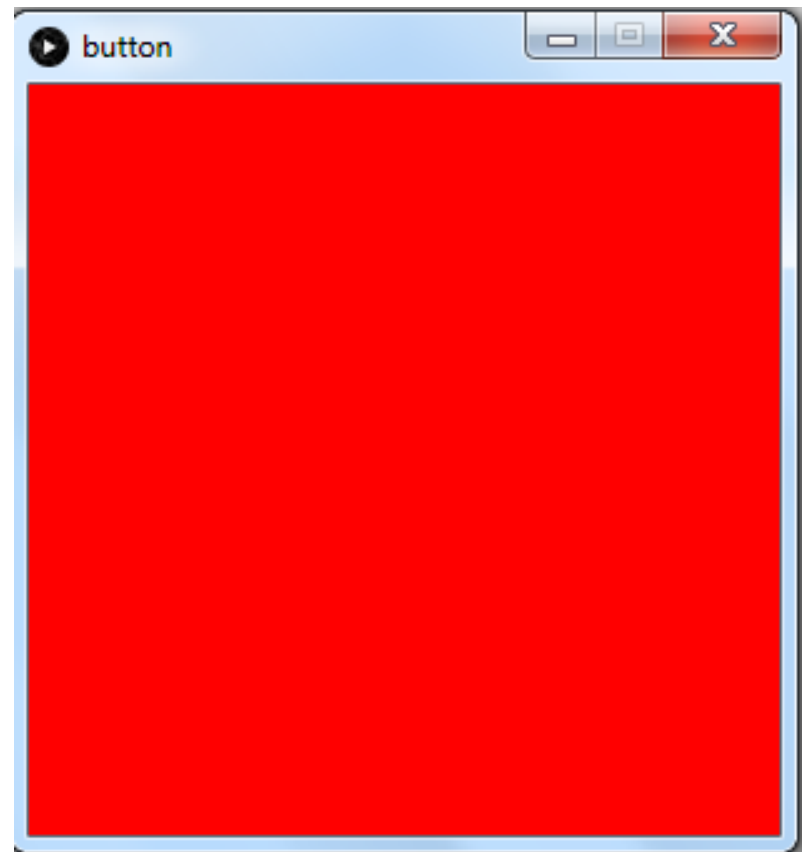
ПРОГРАМА

```
const int BUTTON = 7;
int b;

void setup() {
    Serial.begin(9600);
    pinMode(BUTTON, INPUT);
}

void loop() {
    b = digitalRead(BUTTON);
    if(b == 1){
        Serial.println("BUTTON IS PRESSED");
    } else {
        Serial.println("BUTTON IS NOT PRESSED");
    }
    delay(200);
}
```

ПРОГРАМА НА PROCESSING



COPC КОД – PROCESSING / ARDUINO

```
import processing.serial.*;
Serial arduino;
int s;

void setup(){
    size(300, 300);
    arduino = new Serial(this, "COM12", 9600);
}

void draw(){
    if(arduino.available() > 0){
        s = arduino.read();
    }

    if(s == 'a'){
        background(255, 0, 0);
    } else if (s == 'b'){
        background(0, 0, 255);
    }
}
```

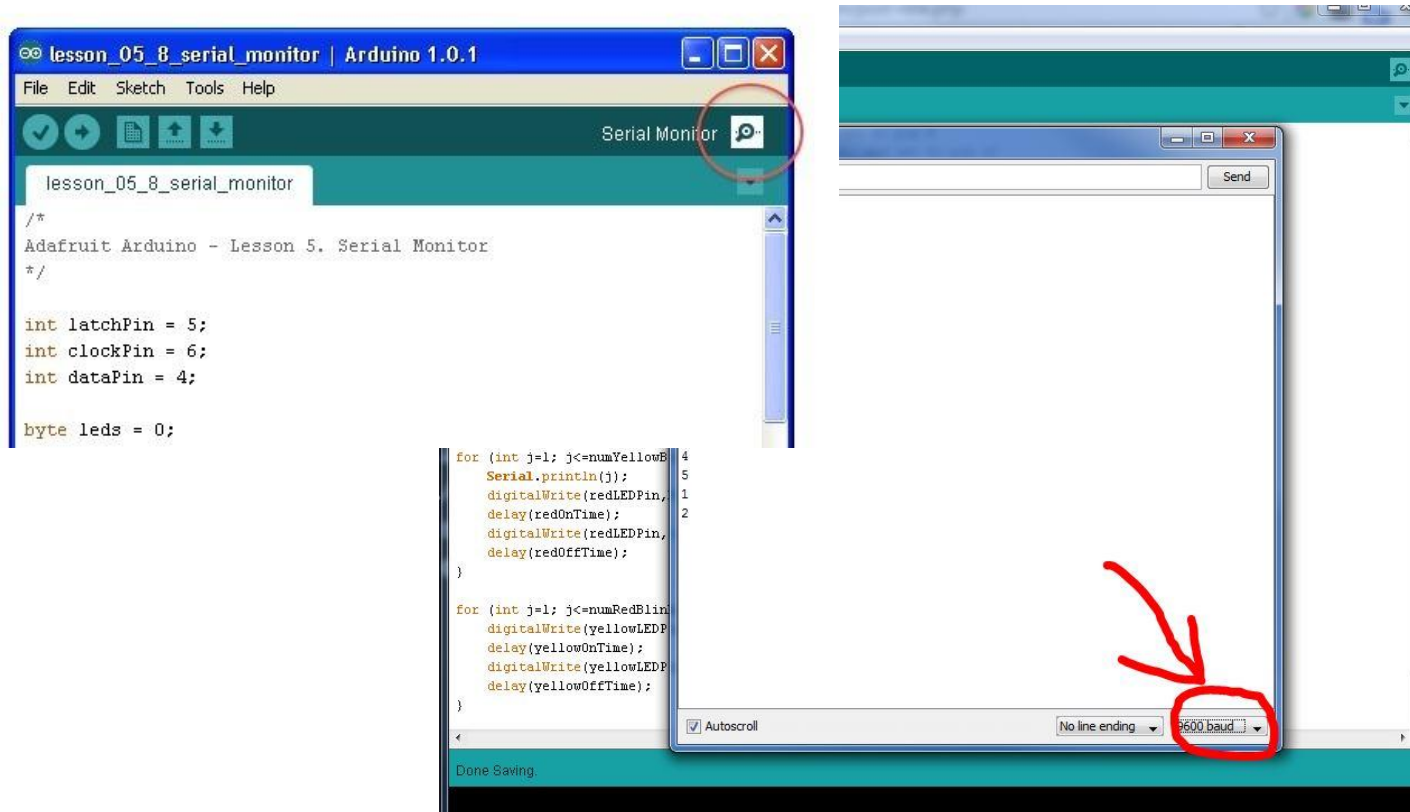
```
const int BUTTON = 7;
int b;

void setup() {
    Serial.begin(9600);
    pinMode(BUTTON, INPUT);
}

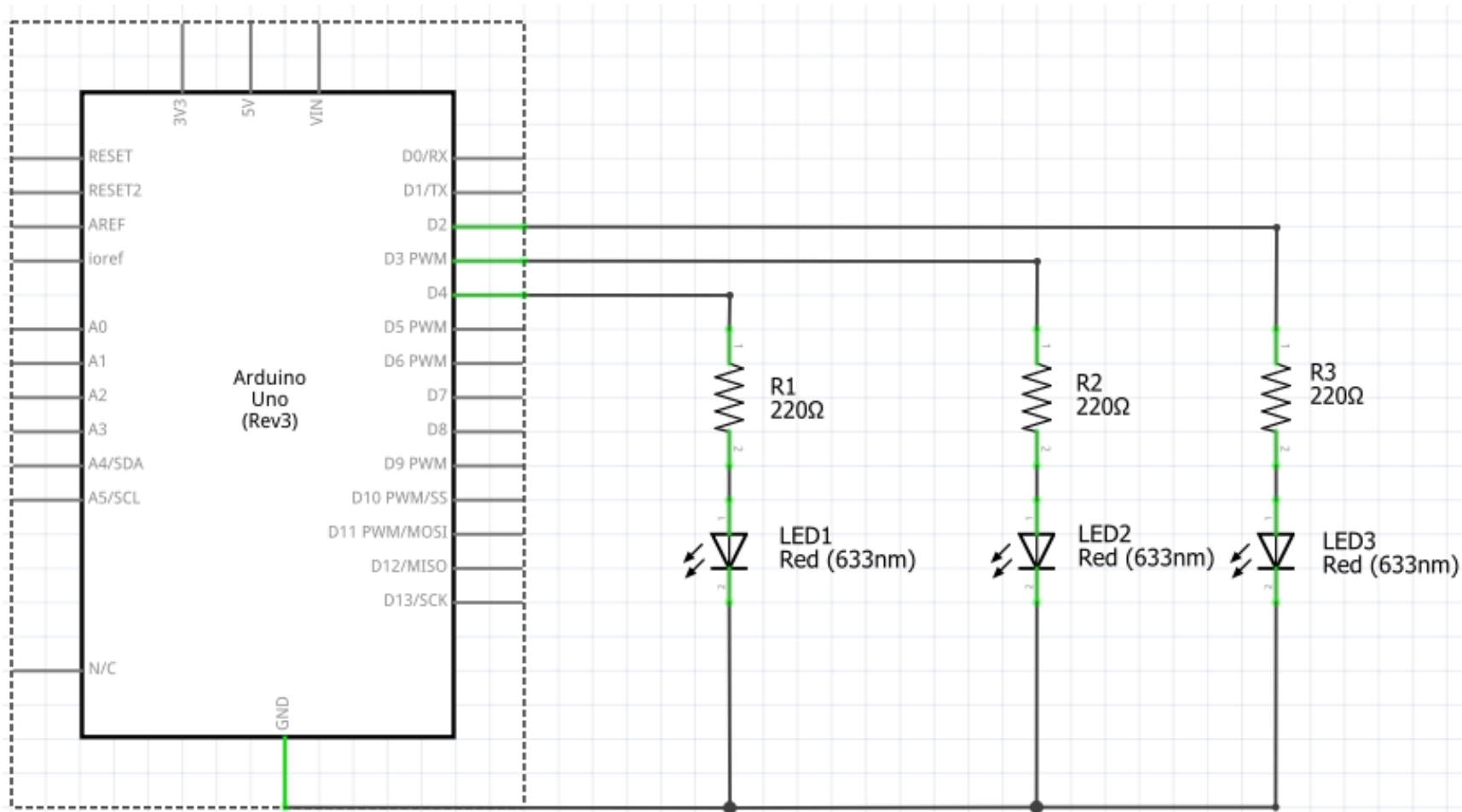
void loop() {
    b = digitalRead(BUTTON);
    if(b == 1){
        Serial.write('a');
    } else {
        Serial.write('b');
    }
}
```

ЗАДАЧА 6.3

❖ Серийна комуникация: Компютър -> Ардуино



ЕЛЕКТРИЧЕСКА СХЕМА



ПРОГРАМА

```
const int LED1 = 2;
const int LED2 = 3;
const int LED3 = 4;
char s;

void setup() {
    Serial.begin(9600);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
}
```

```
void loop() {
    if(Serial.available() > 0){
        s = Serial.read();
    }
    if(s == 'a'){
        digitalWrite(LED1, HIGH);
        digitalWrite(LED2, LOW);
        digitalWrite(LED3, LOW);
    } else if(s == 'b'){
        digitalWrite(LED1, LOW);
        digitalWrite(LED2, HIGH);
        digitalWrite(LED3, LOW);
    } else if(s == 'c'){
        digitalWrite(LED1, LOW);
        digitalWrite(LED2, LOW);
        digitalWrite(LED3, HIGH);
    } else{
        digitalWrite(LED1, LOW);
        digitalWrite(LED2, LOW);
        digitalWrite(LED3, LOW);
    }
}
```

ПРОГРАМА НА PROCESSING



ΠΡΟΓΡΑΜΑ - PROCESSING

```
import processing.serial.*;
Serial arduino;

void setup(){
  size(800, 600);
  arduino = new Serial(this, "COM12", 9600);
}

void draw(){
  fill(255, 255, 255);
  rect(0, 0, 800, 200);
  fill(0, 255, 0);
  rect(0, 200, 800, 400);
  fill(255, 0, 0);
  rect(0, 400, 800, 600);

  if(mouseY < 200){
    arduino.write('a');
  } else if(mouseY > 200 && mouseY < 400){
    arduino.write('b');
  } else if(mouseY > 400){
    arduino.write('c');
  }
}
```



БЛАГОДАРЯ ВИ ЗА ВНИМАНИЕТО!