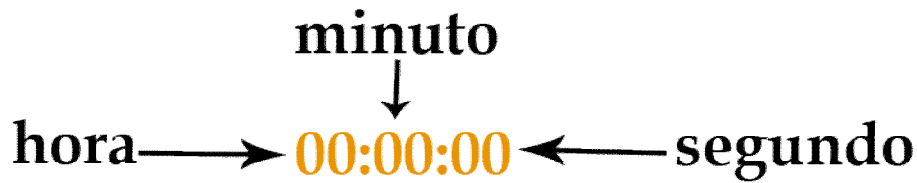


13 - Tempo

No pacote do JPlay temos um contador de tempo no seguinte formato:



13.1 – Cronômetro regressivo ou cronômetro progressivo

Para a classe `time` funcionar como um cronômetro regressivo na opção, `boolean` `crescentTime`, passe o valor `false`. Para funcionar como um cronômetro progressivo passe o valor `true`.

Essa opção existe em todos os construtores na classe `Time`.

13.2 - Construtores da classe `Time`

13.2.1 - `public Time(int x, int y, boolean crescentTime)`

Para esse construtor devem ser passadas as posições (x,y) de onde o tempo será desenhado e se o `Time` será usado como um cronômetro regressivo ou progressivo.

Usando esse construtor o tempo é iniciado com o valor zero, ou seja, hora, minutos e segundos são iniciados com o valor zero.

Exemplo:

```
Time tempo = new Time(100,100, true);  
X = 100, Y = 100, Cronômetro progressivo.
```

```
Time tempo = new Time(100,100, false);  
X = 100, Y = 100, Cronômetro regressivo.
```

13.2.2 - `public Time(int hour, int minute, int second, int x, int y)`

Nesse construtor temos que passar o valor da hora, minuto e segundo que serão usados além da posição onde o tempo será desenhado.

Exemplo: Tempo igual a 5 minutos

```
Time tempo = new Time(0,5,0, 456, 329, false);  
hora = 0, minuto = 5, segundo = 0, x = 456, y = 329, tempo decrescente ou cronômetro regressivo.
```

13.2.3 – `public Time(int hour, int minute, int second, int x, int y, Font font, Color color, boolean crescentTime)`

Nesse construtor temos a liberdade de escolher a cor e a fonte que serão usadas para exibir o tempo.

Exemplo:

```
//public Time(int hour, int minute, int second, int x, int y, Font font, Color color, boolean crescentTime )  
Font fonte = new Font("Comic Sans MS", Font. TRUEETYPE_FONT, 40);  
Time tempo = new Time( 1, 23, 34, 100, 100, fonte , Color.Yellow, false);
```

Obs.: Nos outros construtores a fonte e a cor padrões são as seguintes:
Font("Arial",Font.TRUETYPE_FONT, 20) e Color.YELLOW .

13.3 – Métodos da classe Time

public String toString() – retorna uma *string* com o valor do tempo, a string retornada é no formato 00:00:00.

public void draw(String string) – desenha uma mensagem que é uma escolha do usuário e em seguida desenha o valor do tempo.

public void draw() – desenha o valor do tempo na tela.

public void setColor(Color color) – seta a cor a ser usada na fonte que será usada para desenhar o tempo.

public void setFont(Font font) – seta a fonte que será usada para desenhar o tempo.

public boolean timeEnded() – retorna se o tempo terminou, esse método só serve se o tempo escolhido é do tipo cronômetro regressivo.

public void setHour(int hour) – seta o valor da hora.

public void setMinute(int minute) – seta o valor dos minutos.

public void setSecond(int second) – seta o valor dos segundos.

public long getHour() – retorna o valor da hora.

public long getMinute() – retorna o valor do minuto.

public long getSecond() – retorna o valor do segundo.

public long getTotalSecond() – converte o tempo em segundos e retorna o valor total.

public void setTime(int hour, int minute, int seconds) – seta o valor do tempo.

Exemplo 17: Mostra na tela um contador de tempo regressivo e outro progressivo.

```
public class Exemplo17
{
    public static void main(String[] args)
    {
        Window janela = new Window(800,600);
        Keyboard keyboard = janela.getKeyboard();
        GamelImage backGround = new GamelImage("fundo.png");

        Time tempo1 = new Time(100, 100,true);
        tempo1.setColor(Color.yellow);

        Time tempo2 = new Time(1, 39, 56, 100, 200,false);
        tempo2.setColor(Color.cyan);

        boolean executando = true;

        while(executando)
        {
            backGround.draw();
            tempo1.draw("Tempo 1: ");
            tempo2.draw("Tempo 2: ");
        }
    }
}
```

```
janela.display();  
  
    if ( keyboard.keyDown(Keyboard.ESCAPE_KEY) == true )  
        executando = false;  
    }  
janela.exit();  
}  
}
```