- MẢNG NÀY DÙNG ĐỂ LƯU TRỮ CÁC KHUNG HÌNH ĐỂ TẠO ẢNH -

private BufferedImage[] idelAni;

- MỘT BIẾN ĐẾM TỰ ĐỘNG ĐÁNH DẤU THỜI GIAN TRÔI QUA GIỮA CÁC HOẠT ẢNH ,BIẾN LƯU TRỮ CHỈ SỐ CỦA KHUNG HÌNH, BIẾN LƯU TRỮ TỐC ĐỘ CỦA HOẠT ẢNH-

private int aniTick, aniIndex, aniSpeed = 30;

* PHƯƠNG THỨC NÀY DÙNG ĐỂ LOAT ANIMATION -

private void loadAnimation() {

* KHỞI TẠO MẢNG CÓ KÍCH THƯỚC LÀ 5 LƯU TRỮ HOẠT ẢNH -

idelAni = new BufferedImage[5];

* DUYỆT QUA TỪNG CHỈ SỐ ĐỂ TRÍCH XUẤT ẢNH -

for (int i=0; i < idelAni.length ; i++) {

* DÙNG HÀM getSubimage CỦA ĐỐI TƯỢNG BufferedImage để trích xuất được ra các cái hoạt ảnh và lưu trữ nó vào trong biến ảnh.

idelAni[i] = img.getSubimage(i\*64, 0,64,40 );

}

* SAU BƯỚC KHỞI TẠO RA THÌ CÁC HOẠT ẢNH ĐÃ NẰM GỌN TRONG MẢNG ẢNH -

private void updateAnimationTick() {

aniTick ++; 0 1 2 3 ….. 30

* Vượt qua giới hạn 30 thì aniTich = 0

if(aniTick >= aniSpeed) {

aniTick = 0;

aniIndex ++; 1 2,3 4 5 ….. 1 2 3 4 5 … 1 2 3 4 5

if(aniIndex >=idelAni.length) {

aniIndex = 0;

}

}

}

}

public void paintComponent(Graphics g) {

super.paintComponent(g);

updateAnimationTick();

g.drawImage(idelAni[aniIndex], (int)xDelta, (int)yDelta, null);

}