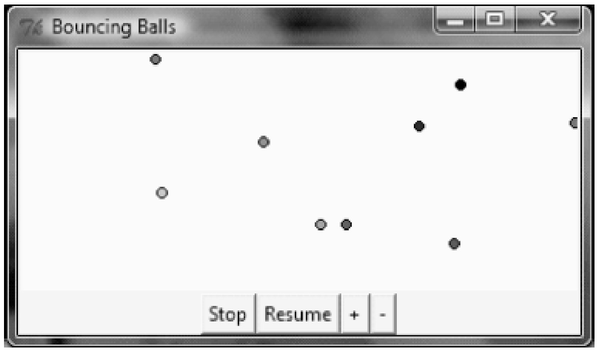
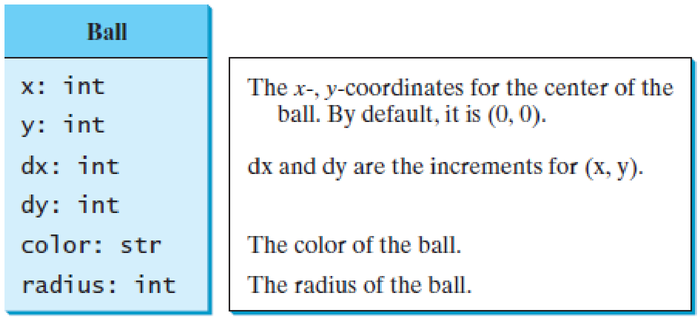
The program enables the user to click the and buttons to add a ball or remove a ball from the canvas, and click the *Stop* and *Resume* buttons to stop the ball movements or resume them.
Each ball has its own center location (**x**, **y**), **radius**, **color**, and next increment for its center position, **dx** and **dy**.
You can define a class to encapsulate all this information, as shown in Figure below b)



(a)



**Ball**

| | |
|---|---|
| x: int<br>y: int | The *x*-, *y*-coordinates for the center of the ball. By default, it is (0, 0). |
| dx: int<br>dy: int | dx and dy are the increments for (x, y). |
| color: str | The color of the ball. |
| radius: int | The radius of the ball. |

(b)

Initially, the ball is centered at (0, 0), and **dx** = **2** and **dy** = **2**.
In the animation, the ball is moved to (**x** + **dx**, **y** + **dy**). When the ball reaches the right boundary, change **dx** to **-2**. When the ball reaches the bottom boundary, change **dy** to **-2**. When the ball reaches the left boundary, change **dx** to **2**. When the ball reaches the top boundary, change **dy** to **2**. The program simulates a bouncing ball by changing the **dx** or **dy** values when the ball touches the boundary of the canvas.
When the + button is clicked, a new ball is created. How do you store the ball in the program? You can store the balls in a list. When the - button is clicked, the last ball in the list is removed.