1 **import** java.awt.\*;

2

3 **public** **class** ParentSnowmanV2

4 {

5 **private** **static** **int** snowmanCount = 0;

6 **private** **static** **int** w = 40;

7 **private** **static** **int** h = 77;

8 **private** **int** x = 8;

9 **private** **int** y = 30;

10 **private** String name;

11 **private** Color hatColor= Color.BLACK;

12 **private** **int** xSpeed = 2;

13 **private** **int** ySpeed = 2;

14 **private** boolean visible = **true**;

15

16 **public** ParentSnowmanV2()

17 {

18 snowmanCount++;

19 }

20 **public** ParentSnowmanV2(**int** intialX, **int** intialY, String name,

21 Color hatColor)

22 {

23 x = intialX;

24 y = intialY;

25 **this**.name = name;

26 **this**.hatColor = hatColor;

27 snowmanCount++;

28 }

29 **private** **void** copy(ParentSnowmanV2 ps) **//copies 4 data members**

30 {

31 ps.setX(x);

32 ps.setY(y);

33 ps.setName(name);

34 ps.setHatColor(hatColor);

35 }

36 **public** ParentSnowmanV2 partialClone()

37 {

38 ParentSnowmanV2 theClone = **new** ParentSnowmanV2();

39 **this**.copy(theClone);

40 **return** theClone;

41 }

42 **public** **boolean** shallowEquals(ParentSnowmanV2 ps)

43 {

44 **if**(**this** == ps)

45 {

46 **return** **true**;

47 }

48 **else**

49 {

50 **return** **false**;

51 }

52 }

53 **public** **boolean** equals(ParentSnowmanV2 ps)

54 {

55 **if**(hatColor.equals(ps.getHatColor())) **//same hat color**

56 {

57 **return** **true**;

58 }

59 **else**

60 {

61 **return** **false**;

62 }

63 }

64 **public** **boolean** collidedWith(ParentSnowmanV2 ps)

65 {

66 **if**( !(x > ps.getX( ) + w || x + w < ps.getX( ) ||

67 y > ps.getY( ) + h || y + h < ps.getY( )))

68 {

69 **return** **true**;

70 }

71 **else**

72 {

73 **return** **false**;

74 }

75 }

76 **public** **void** show(Graphics g) **// g is the game board object**

77 {

78 i**n**t[] xPoly = {x + 20, x + 15, x + 25};

79 **int**[] yPoly = {y + 25, y + 30, y + 30};

80

81 g.setColor(hatColor);

82 g.fillRect(x + 15, y, 10, 15); **// hat**

83 g.fillRect(x + 10, y + 15, 20, 2); **// brim**

84 g.setColor(Color.WHITE);

85 g.fillOval(x + 10, y + 17, 20, 20); **// head**

86 g.fillOval(x, y + 37, 40, 40); **// body**

87 g.setColor(Color.RED);

88 g.fillPolygon(xPoly, yPoly, 3); **// nose**

89 g.setColor(Color.BLACK);

90 g.setFont(new Font("Arial", Font.BOLD, 16));

91 g.drawString(name, x + 16, y + 62); **// name**

92 }

93 **public** **static** **int** getSnowmanCount()

94 {

95 **return** snowmanCount;

96 }

97 **public** **int** getXSpeed()

98 {

99 **return** xSpeed;

100 }

101 **public** **void** setXSpeed(int newXSpeed)

102 {

103 xSpeed = newXSpeed;

104 }

105 **public** **int** getYSpeed()

106 {

107 **return** ySpeed;

108 }

109 **public** **void** setYSpeed(**int** newYSpeed)

110 {

111 ySpeed = newYSpeed;

112 }

113 **public** **void** setHatColor(Color newHatColor)

114 {

115 hatColor = newHatColor;

116 }

117 **public** Color getHatColor()

118 {

119 **return** hatColor;

120 }

121 **public** **int** getX()

122 {

123 **return** x;

124 }

125 **public** **void** setX(**int** newX)

126 {

127 x = newX;

128 }

129 **public** **int** getY()

130 {

131 **return** y;

132 }

133 **public** **void** setY(**int** newY)

134 {

135 y = newY;

136 }

137 **public** String getName()

138 {

139 **return** name;

140 }

141 **public** **void** setName(String newName)

142 {

143 name = newName;

144 }

145 **public** **boolean** getVisible()

146 {

147 **return** visible;

148 }

149 }

**Figure 7.12 The class ParentSnowmanV2.**