Line «interface» «enumerati... isMarked :Boolean BoxListener POSITION\_LEFT :String = "left" {readOnly} Gamemode POSITION\_UPPER :String = "upper" {readOnly + boxEvent(Boolean) :void Local + getIsMarked() :Boolean Network + Line() + setIsMarked(Boolean) :void -bot-upperLine JPanel LinePicture color :Color colorBlank :Color = Color.YELLOW {readOnly} colorHighlight :Color = Color.RED {readOnly} colorPainted :Color = Color.BLACK {readOnly} HEIGHT :int = 10 {readOnly} Player **JFrame** highlight :Boolean **DotAndBoxes** + HORIZONTAL :String = "horizontal" {readOnly} name :String score :int dialog :CreateGameDialog + listeners :ArrayList<PictureListener> = new ArrayList<>() + getName() :String VERTICAL :String = "vertical" {readOnly} createGame() :void + getScore() :int WIDTH :int = 50 {readOnly} DotAndBoxes() + increaseScore() :void x :int GUI() :void ~player, \* + Player() y :int main(String[]) :void + Player(int) -markedBy addListener(PictureListener) :void + setName(String) :void lineHighlight():void +currer+++playerW\ lineNormal() :void - linePaint():void + LinePicture(Line, String) + paintComponent(Graphics) :void pictureEvent() :void setColor() :void Box bottomLine :Line leftLine :Line marked :Boolean

markedBy :Player

### javax.swing.JDialog

## Create Game Dialog

- buttonCreateGame :JButton {readOnly}
- buttonGroup :ButtonGroup {readOnly}
- labelGameboardSize :JLabel {readOnly}
- labelGameMode :JLabel {readOnly}
- labellpEnemy :JLabel {readOnly}
- labelSpielerName :JLabel {readOnly}
- radiobuttonLocal :JRadioButton {readOnly}
- radiobuttonNetwork :JRadioButton {readOnly}
- spinnerGameboardSize :JSpinner {readOnly}
- textFieldIpEnemy :JTextField {readOnly}
- textFieldPlayerName :JTextField {readOnly}
- + CreateGameDialog(Frame)
- + getGameboardSize() :int
- + getGamemode() :Gamemode
- + getIpAdress() :String
- + getPlayerName() :String
- initComponents() :void

# dialog 1

# **JPanel**

### **DotPicture**

- color :Color = Color.BLACK {readOnly}
- DIMENSION :int = 10 {readOnly}
- + DotPicture()
- + paintComponent(Graphics) :void



