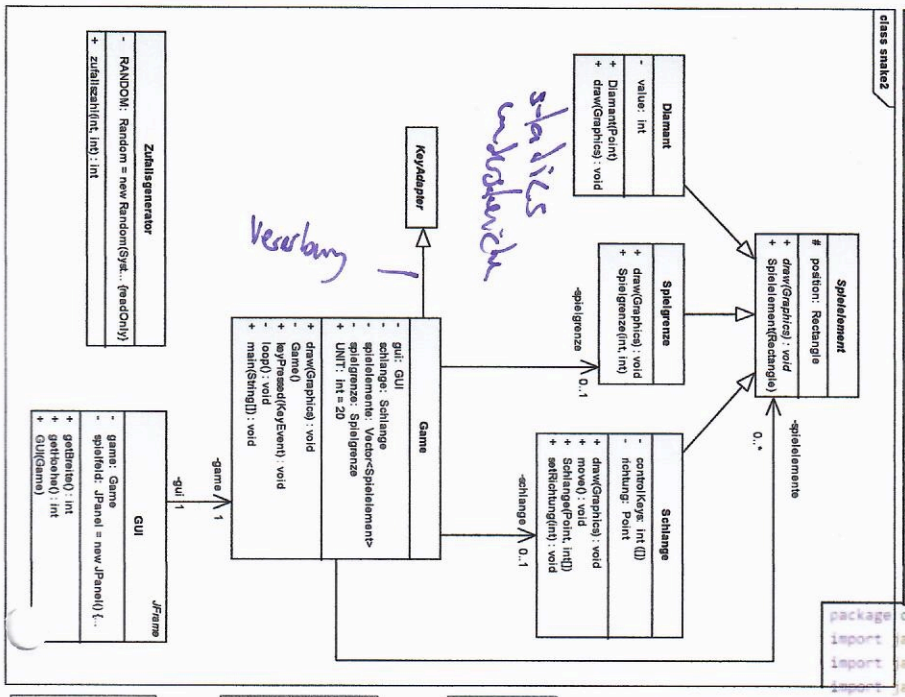
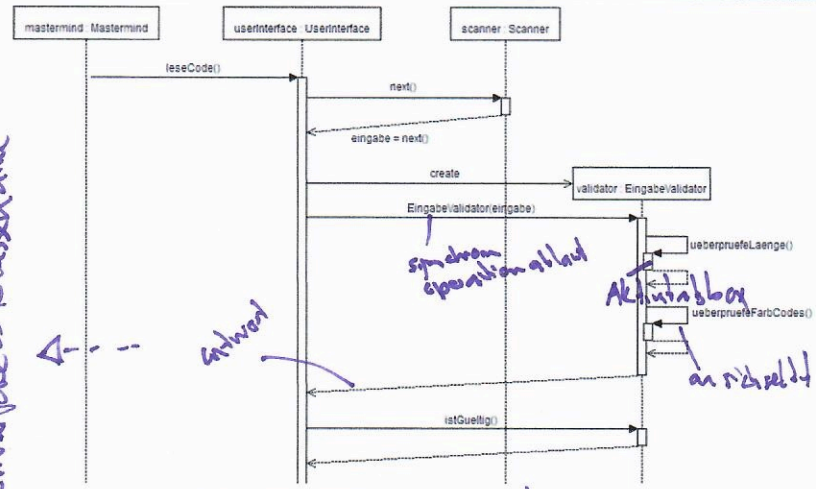
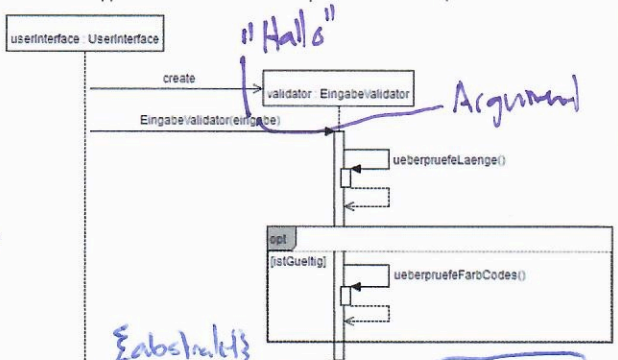
Mirio Eggmann

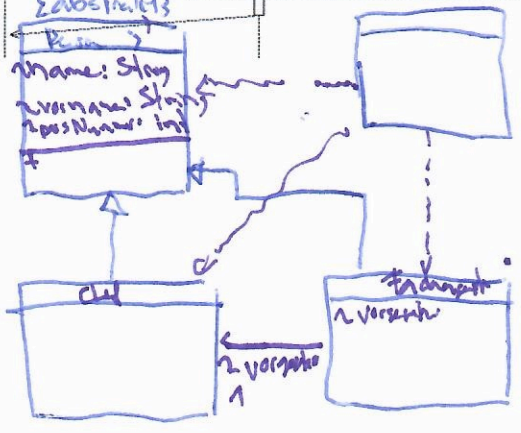



```
package ch.jmelab.tangerine.views;
import java.awt.Graphics;
import java.awt.Point;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.io.IOException;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import ch.jmelab.tangerine.controllers.EditorController;
import ch.jmelab.tangerine.libraries.FigureSaver;
import ch.jmelab.tangerine.libraries.adapters.EditorMouseAdapter;
@SuppressWarnings("serial")
final class EditorPanel extends JPanel {
  private EditorController editorController;
  protected EditorPanel(final EditorController editorController) {
    this.editorController = editorController;
    addMouseListener(new MouseAdapter() {
      public void mousePressed(MouseEvent e) {
        editorController.setStartPoint(new Point(e.getX(), e.getY()));
      public void mouseReleased(MouseEvent e) {
        editorController.createFigureWithEndPoint(new Point(e.getX(), e
            .getY()));
        repaint();    } }); }
    @Override
    protected void paintComponent(Graphics g) {
      super.paintComponent(g);
      editorController.redrawAll(g); }}
```

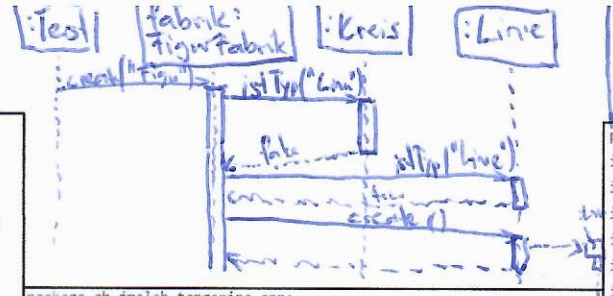
```
package ch.jmelab.tangerine.views;
import java.awt.Dimension;
import java.awt.Point;
import java.awt.Rectangle;
import java.awt.Toolkit;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import ch.jmelab.tangerine.controllers.EditorController;
@SuppressWarnings("serial")
public class EditorFrame extends JFrame {
  private EditorController editorController = new EditorController();
  public EditorFrame(int width, int height) {
    createAndSetEditorPanel();
    centerWindow(width, height);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setVisible(true);
    addKeyListener(new KeyAdapter() {
      public void keyPressed(KeyEvent key) {
        if (key.getKeyCode() == KeyEvent.VK_Q) {
          int answer = JOptionPane.showConfirmDialog(getRootPane(),
              "Möchten Sie die Applikation beenden?",
              "Bestätigung", JOptionPane.YES_NO_OPTION);
          if (answer == 0) {
            JOptionPane.showMessageDialog(getRootPane(),
                "Applikation wird beendet.");
            System.exit(0);          }
          return;        }
        if (key.getKeyCode() == KeyEvent.VK_K) {
          editorController.setFigureType('k');
          return;
        } else if (key.getKeyCode() == KeyEvent.VK_L) {
          editorController.setFigureType('l');
          return;
        } else if (key.getKeyCode() == KeyEvent.VK_R) {
          editorController.setFigureType('r');
          return;        }      }); }
  private void createAndSetEditorPanel() {
    JPanel panel = new EditorPanel(editorController);
    setContentPane(panel);  }
  private void centerWindow(int width, int height) {
    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
    Rectangle windowSize = new Rectangle();
    windowSize.width = width;
    windowSize.height = height;
    windowSize.x = (screenSize.width - windowSize.width) / 2;
    windowSize.y = (screenSize.height - windowSize.height) / 2;
    setBounds(windowSize);  }}
```







Mario Eggmann



- mouseEntered
- mouseClicked
- mouseReleased
- mouseExited
- mousePressed

```
package ai;
abstract class Person {
  protected String name;
  protected String vorname;
  protected int personalNummer;
  public Person(String n, String vn, int personNumb) {
    this.name = n;
    this.vorname = vn;
    this.personalNummer = personNumb; }
  public void print() {
    System.out.println("Name: "+name);
    System.out.println("Vorname: "+vorname);
    System.out.println("Personal Nummer: "+personalNummer); }
  public abstract int berechneFerien(int alter);}
```

```
package ch.jmelab.tangerine.app;
import java.io.IOException;
import ch.jmelab.tangerine.views.Display;
import ch.jmelab.tangerine.views.EditorFrame;
public class Tangerine {
  public static void main(String[] args) throws InterruptedException,
      IOException {
    new Tangerine();
  }
  private Tangerine() {
    @SuppressWarnings("unused")
    EditorFrame frame = new EditorFrame(800, 600);
  }
}
```

```
package ch.jmelab.tangerine.controllers;
import java.awt.Graphics;
import java.awt.Point;
import sun.awt.RepaintArea;
import com.sun.prism.paint.Color;
import ch.jmelab.tangerine.models.Drawing;
import ch.jmelab.tangerine.models.Figure;
import ch.jmelab.tangerine.models.figures.Circle;
import ch.jmelab.tangerine.models.figures.Line;
import ch.jmelab.tangerine.models.figures.Rectangle;
public class EditorController {
  public static Color LINECOLOR = Color.BLACK;
  public static int LINETHICKNESS = 1;
  private Drawing drawing = new Drawing();
  private char figureType;
  private Point startPoint;
  private Point endPoint;
  public void redrawAll(Graphics g) {
    drawing.drawFigures(g);}
  public void setFigureType(char figureType) {
    this.figureType = figureType;}
  public void setStartPoint(Point startPoint) {
    this.startPoint = startPoint;}
  public void createFigureWithEndPoint(Point endPoint) {
    Figure figure = null;
    if (figureType != 0) {
      if (figureType == 'k') {
        figure = new Circle();
      } else if (figureType == 'l') {
        figure = new Line();
      } else if (figureType == 'r') {
        figure = new Rectangle();}
      drawing.addFigure(figure.zeichneMitMaus(
          (int)startPoint.getX(), (int)startPoint.getY(),
          (int)endPoint.getX(), (int)endPoint.getY()));}}}
```

```
package ch.jmelab.tangerine.models;
import java.awt.Graphics;
import java.util.List;
public class Group extends Figure {
  public List<Figure> FigurenGruppe;
  public Group(List<Figure> figurenGruppe) {
    this.setFigurenGruppe(figurenGruppe); }
  public List<Figure> getFigurenGruppe() {
    return FigurenGruppe; }
  public void setFigurenGruppe(List<Figure> figurenGruppe) {
    FigurenGruppe = figurenGruppe;  }
  public void printGruppe() {
    for (Figure figur : this.FigurenGruppe) {
      System.out.println(figur.getClass().toString());    } }
  @Override
  public String[] getAllInformations() {
    // TODO Auto-generated method stub
    return null;  }
  @Override
  public void zeichneFigur(Graphics g) {  }
  @Override
  public Figure setAllInformations(String[] informationen) {
    return this;  }
  @Override
  public Figure zeichneMitMaus(int startX, int startY, int endX, int endY) {
    return this;  }}
```

```
package ch.jmelab.tangerine.libraries.adapters;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
public class EditorMouseAdapter extends MouseAdapter {
  public void mousePressed(MouseEvent event) {
    System.out.println(event.getPoint());}
  public void mouseReleased(MouseEvent event) {
    System.out.println(event.getPoint());}}
```

```
package
public  Personen {
  public static void main(String args[]) {
    Person[] angestellte = new Person[4];
    angestellte[0] = new Chef("Hallo", "Welt", 3333, "Informatik");
    angestellte[1] = new Fachangestellter("Mario", "Eggmann", 1234, (Chef) angestellte[0]);
    angestellte[2] = new Lernender("Brachi", "Brechbühler", 9876, 9999);
    for (int i = 0; i < angestellte.length; i++) {
      angestellte[i].print(); } }}
```

```
package ch.jmelab.tangerine.models;
import java.awt.Color;
import java.awt.Graphics;
public abstract class Figure {
  protected int x, y;
  protected Color linienFarbe;
  protected int linienDicke;
  public Figure(int x, int y, Color linienFarbe, int linienDicke) {
    this.x = x;
    this.y = y;
    this.linienFarbe = linienFarbe;
    this.linienDicke = linienDicke;}
  public Figure() {}
  public int getX() {
    return x;}
  public void setX(int x) {
    this.x = x;}
  public int getY() {
    return y;}
  public void setY(int y) {
    this.y = y;}
  public int getLinienDicke() {
    return linienDicke;}
  public void setLinienDicke(int linienDicke) {
    this.linienDicke = linienDicke;}
  public Color getLinienFarbe() {
    return linienFarbe;}
  public void setLinienFarbe(Color linienFarbe) {
    this.linienFarbe = linienFarbe;}
  public abstract String[] getAllInformations();
  public abstract Figure setAllInformations(String[] informationen);
  public void move(int deltaX, int deltaY) {
    this.setX(this.getX() + deltaX);
    this.setY(this.getY() + deltaY);}
  public abstract Figure zeichneMitMaus(int startX, int startY, int endX,
      int endY);
  public abstract void zeichneFigur(Graphics g);
```

```
package ch.jmelab.tangerine.models;
import java.awt.BasicStroke;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.util.ArrayList;
import java.util.List;
public class Drawing {
  private List<Figure> figures = new ArrayList<Figure>();
  public Drawing() {
    //this.figuren = figuren;
  }
  public void drawFigures(Graphics g) {
    Graphics2D g2d = (Graphics2D) g;
    for (Figure f : figures) {
      g2d.setColor(f.getLinienFarbe());
      g2d.setStroke(new BasicStroke(f.getLinienDicke()));
      f.zeichneFigur(g);
    }
  }
  public void addFigure(Figure figur) {
    figures.add(figur);
    //repaint();
  }
  public void removeFigure(Figure figur) {
    figures.remove(figur);
    //repaint();
  }
  public void removeAllFigures() {
    figures.clear();
    //repaint();
  }
}
```

Entered / Exited (event.getX() + event.getY())

this.add(g)

implements

```
package ai;
class Fachangestellter extends Person {
  private Chef vorgesetzter;
  public Fachangestellter(String n, String vn, int personNumb, Chef vorge) {
    super(n, vn, personNumb);
    this.vorgesetzter = vorge;}
  @Override
  public int berechneFerien(int alter) {
    int ferien = 4;
    if (alter >= 60) {
      ferien = 6;
    } else if(alter >= 50) {
      ferien = 5;  }
    return ferien;  }}
```

Label

Text → Textfield

- key Typed
- key Pressed
- key Released

Person angestellte = new Person();
angestellte = (Chef) angestellte;
Person angestellte = new Chef();