

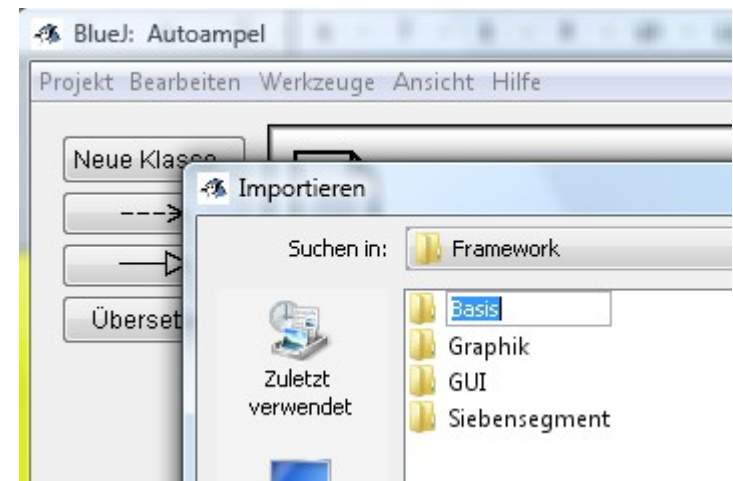
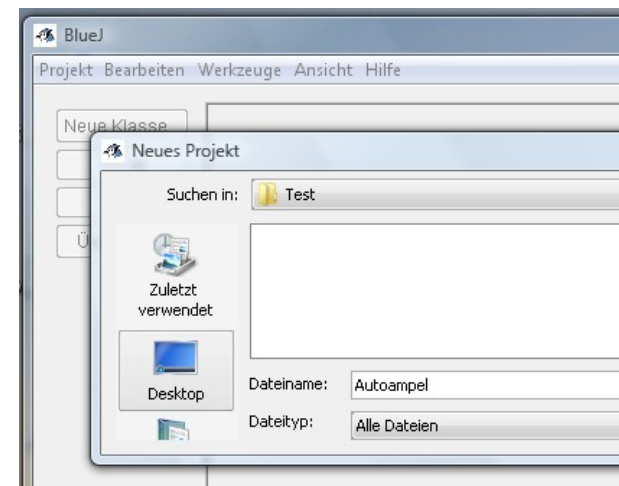
GUI mit BlueJ

Übung:

Autoampel

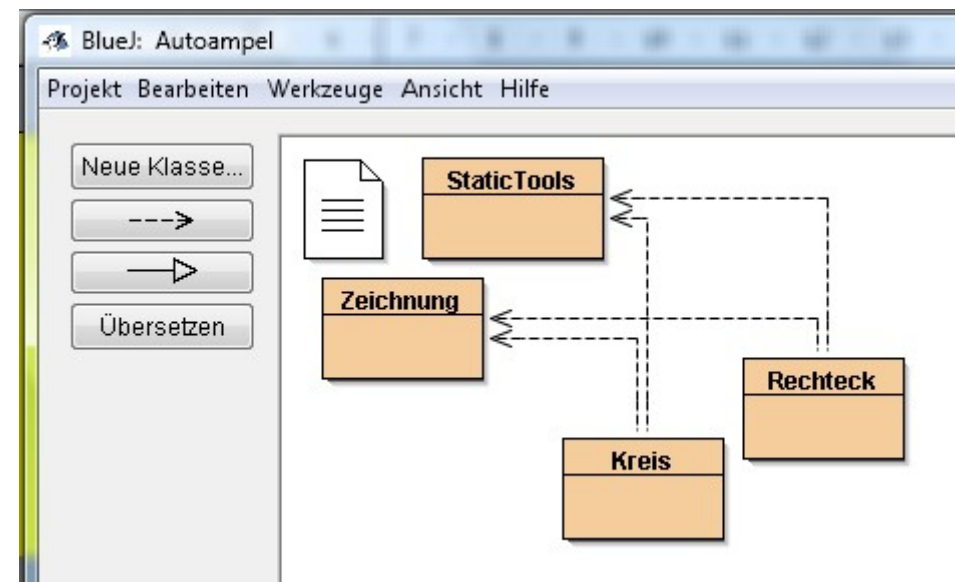
Die Autoampel

- Neues Projekt erstellen
- Benötigten GUI-Elemente importieren (Basis, Graphik)



Die Autoampel

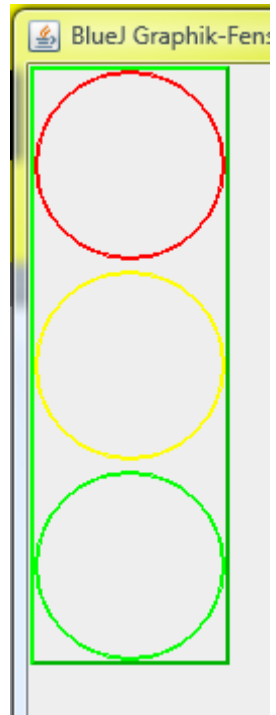
Klassen verwalten:
benötigte
Elemente:
→ Kreis
→ Rechteck
übrige Elemente
löschen



Die Autoampel

```
public class AutoAmpel
{
    private Kreis rot;
    private Kreis gelb;
    private Kreis gruen;
    private Rechteck rechteck ;
    // Konstruktor
    public AutoAmpel()
    {
        rechteck = new Rechteck();
    }
    ...
    rot = new Kreis(48);
    rot.setzeFarbe("rot");
    rot.setzePosition(2, 2);
    rot.sichtbarMachen();

    gelb = new Kreis(48);
    ...
}
```

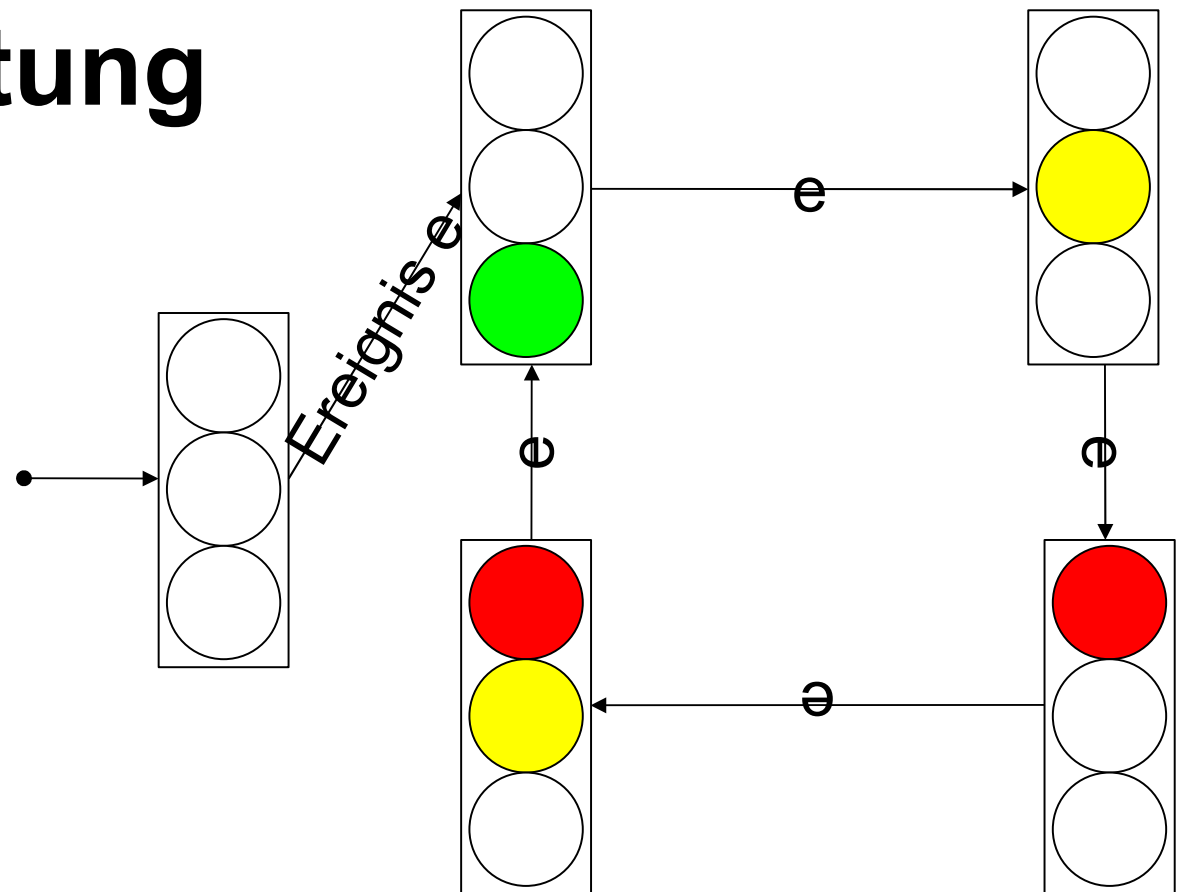


Die Autoampel

Zustandsänderung in der Klasse **Ampelschaltung**

Das Ereignis e :

→ Timer
oder
→ Taste



Die Klasse Ampelschaltung

Ereignis: Zeit (Ablauf einer Zeitspanne)

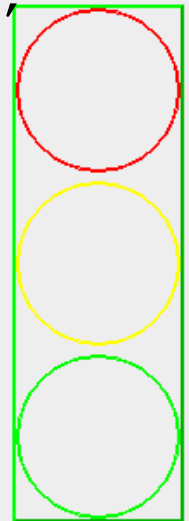
```
public class Ampelschaltung_Warten {
    AutoAmpel ampel;
    public Ampelschaltung_Warten() {
        ampel = new AutoAmpel();
        ampel.setzePosition(100, 100);
    }
    enum Zustaende { aus , gruen , gelb , rot , gelbgruen };
    public void aendereZustand() {
        Zustaende zustand = Zustaende.aus ;
        while (true) {
            switch (zustand) {
                case aus:
                    ampel.ampelAus();
                    zustand = Zustaende.gruen;
                    break;
                case gruen:
                    ampel.gruen();
                    ...
            }
        }
        StaticTools.warte(1000); // warte 1s
    }
}
```

Die Klasse Ampelschaltung

Ereignis: Taster - Polling

```
public class Ampelschaltung_Taste {
    AutoAmpel ampel;
    Taste taste;
    public Ampelschaltung_Taste() {
        ...
        taste = new Taste("Bitte drücken", 200, 50);
        taste.setzePosition(50, 450);
    }

    enum Zustaende {aus, gruen, gelb, rot, gelbgruen};
    public void aendereZustand() {
        Zustaende zustand = Zustaende.aus;
        while (true) {
            switch (zustand) {
                case aus:
                    ampel.ampelAus();
                    zustand = Zustaende.gruen;
                    break;
                ...
            }
            taste.warteBisGedrueckt();
        }
    }
    ...
}
```



Bitte drücken

Die Klasse Ampelschaltung

Ereignis: Taster - Callback

```
public class Ampelschaltung_Callback implements ITuWas {
    AutoAmpel ampel;
    Taste taste;
    public Ampelschaltung_Callback() {
        ...
        taste = new Taste("Bitte drücken", 200, 50);
        taste.setzePosition(50, 450);
        taste.setzeLink(this, 0);
    }

    public void tuWas(int ID) {
        aendereZustand();
    }

    enum Zustaende {aus, gruen, gelb, rot, gelbgruen};
    Zustaende zustand = Zustaende.aus;
    public void aendereZustand() { // Keine Wiederholung !!!
        switch (zustand) {
            case aus:
                ampel.ampelAus();
                ...
        }
    }
}
```