

Dokumentation

Modul226B

Von
Endrit Lena

Inhaltsverzeichnis

Funktionale Anforderungen	3
Nicht funktionale Anforderungen	3
Use-Case Diagramm	4
Klassendiagramm	5
Sequenzdiagramm.....	Fehler! Textmarke nicht definiert.
Testprotokoll	6
Testfälle	7
Code Erläuterung.....	10
KalenderAusgabe.....	10
Kalender	13

Funktionale Anforderungen

Muss:

- Bilder hochladen
- Monat auswählen
- Jahr auswählen
- Template auswählen (min. 2)
- Kalender erstellen, d.h. Bilder sind im gewählten Template am richtigen Ort
- Kalender mit Bilder (crud)
- Ausgabe von Kalender Tabellarisch
- Kalender als PDF exportieren

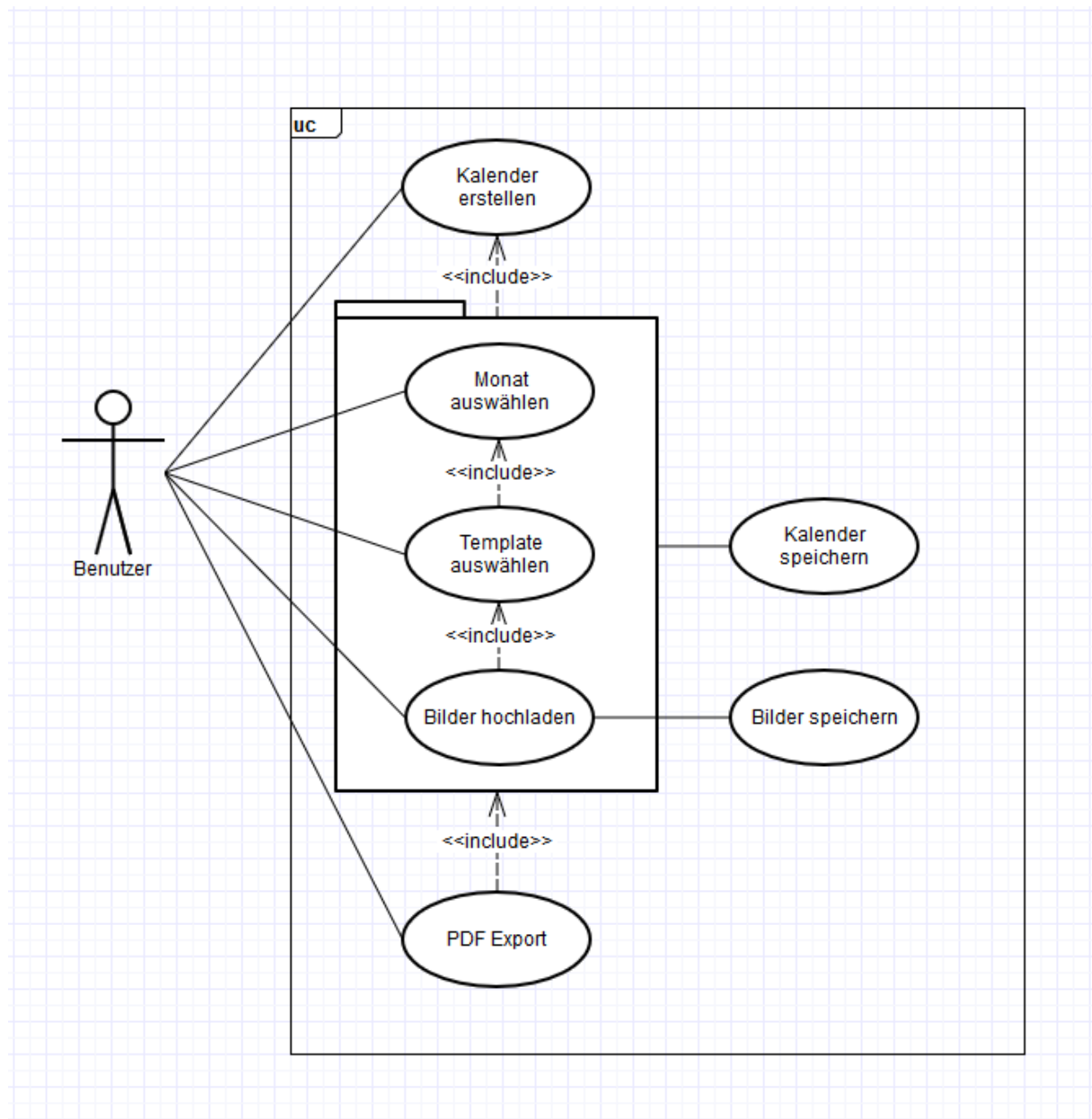
Kann:

- Template erstellen
- Kalender dynamische Texte erfassen / editieren
- Bilder werden automatisch skaliert

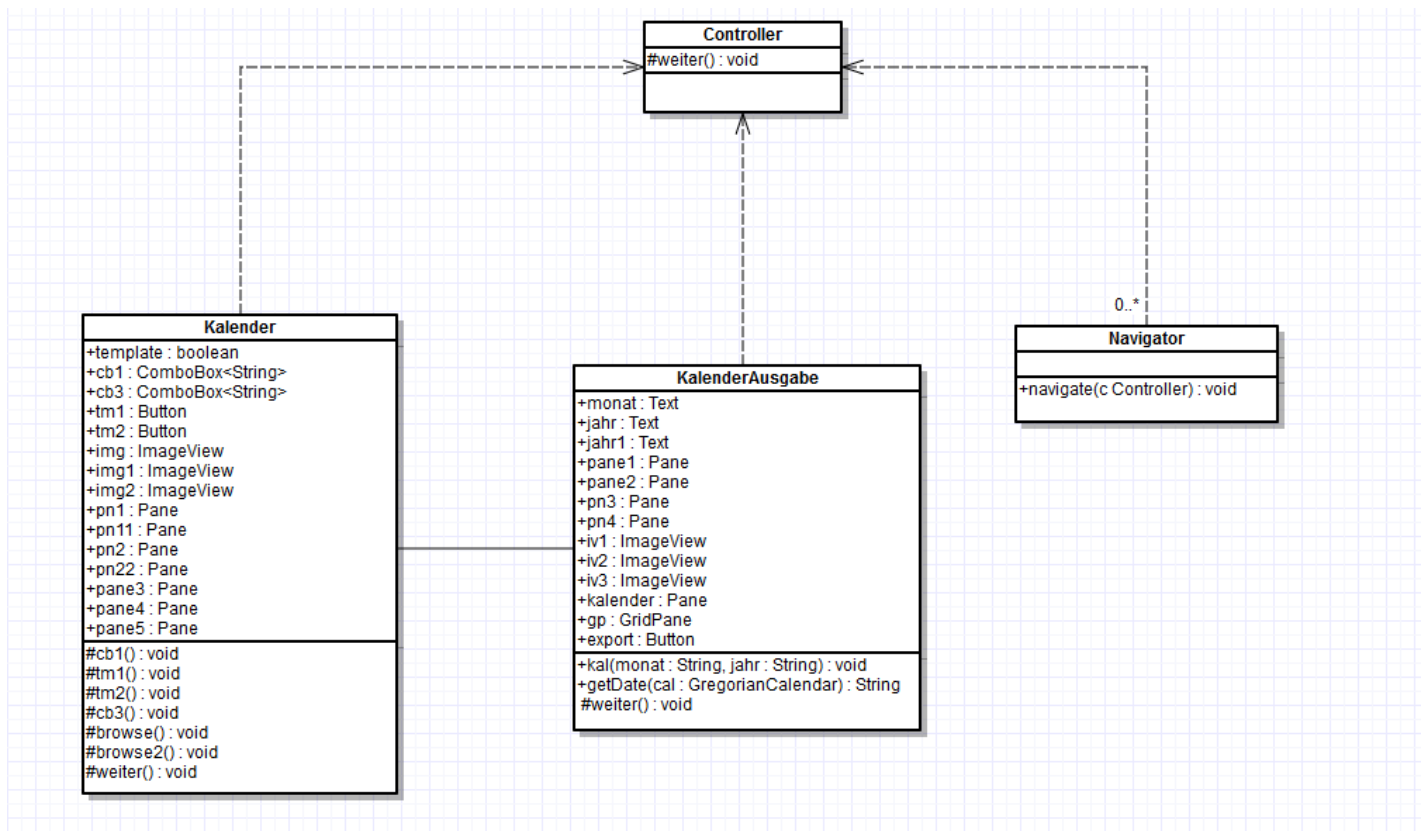
Nicht funktionale Anforderungen

- Java als Programmiersprache
- OpenSource Bibliothek für PDF export

Use-Case Diagramm



Klassendiagramm



Testprotokoll

Funktionale Anforderungen			
Muss			
NR.	Beschreibung	Getestet von	Resultat
1	Template auswählen (min. 2)	Endrit Lena	Funktioniert
2	Monat auswählen	Endrit Lena	Funktioniert
3	Jahr auswählen	Endrit Lena	Funktioniert
4	Kalender erstellen, d.h. Bilder sind im gewählten Template am richtigen Ort	Endrit Lena	Funktioniert
5	Kalender mit Bilder (crud)	Endrit Lena	Fehlgeschlagen
6	Ausgabe von Kalender Tabellarisch	Endrit Lena	Funktioniert
7	Bilder hochladen	Endrit Lena	Funktioniert
8	Kalender als PDF exportieren	Endrit Lena	Fehlgeschlagen
Kann			
9	Template erstellen	Endrit Lena	Fehlgeschlagen
10	Kalender dynamische Texte erfassen / editieren	Endrit Lena	Fehlgeschlagen
11	Bilder werden automatisch skaliert	Endrit Lena	Funktioniert
Nicht funktionale Anforderungen			
12	Java als Programmier- sprache	Endrit Lena	Funktioniert
13	OpenSource Bibliothek für PDF export	Endrit Lena	Fehlgeschlagen

Testfälle

Testfall: 1	Titel: Template auswählen
Datum: 23.12.2016	Art: Funktionale Anforderung
Autor: Endrit Lena	Bezug: Kalender
Vorbedienung:	Es wurden mind. 2 Templates erstellt
Beschreibung:	Es sollten min. 2 Templates ausgewählt werden können
Nachbedienung:	Es können min. 2 Templates ausgewählt werden.

Testfall: 2	Titel: Monat auswählen
Datum: 23.12.2016	Art: Funktionale Anforderung
Autor: Endrit Lena	Bezug: Kalender
Vorbedienung:	Es wurden Monate um auszuwählen zur Verfügung gestellt
Beschreibung:	Ein Monat sollte ausgewählt werden können
Nachbedienung:	Ein Monat kann ausgewählt werden

Testfall: 3	Titel: Jahr auswählen
Datum: 23.12.2016	Art: Funktionale Anforderung
Autor: Endrit Lena	Bezug: Kalender
Vorbedienung:	Es wurden Jahre um auszuwählen zur Verfügung gestellt
Beschreibung:	Ein Jahr sollte ausgewählt werden können
Nachbedienung:	Ein Jahr kann ausgewählt werden

Testfall: 4	Titel: Kalender erstellen
Datum: 23.12.2016	Art: Funktionale Anforderung
Autor: Endrit Lena	Bezug: Kalender
Vorbedienung:	Monat, Jahr und Templates müssen ausgewählt werden können und Bilder müssen Hochgeladen werden
Beschreibung:	Ein Kalender sollte erstellt werden, inkl. Bilder
Nachbedienung:	Ein Kalender mit Bilder wird erstellt.

Testfall: 5	Titel: Kalender mit Bilder
Datum: 23.12.2016	Art: Funktionale Anforderung
Autor: Endrit Lena	Bezug: Kalender
Vorbedienung:	Kalender kann erstellt werden und Bilder können hochgeladen werden
Beschreibung:	Bilder im Kalender sollte gespeichert, geladen, bearbeitet und gelöscht werden können
Nachbedienung:	Es funktioniert nicht

Testfall: 6	Titel: Kalender tabellarisch ausgeben
Datum: 23.12.2016	Art: Funktionale Anforderung
Autor: Endrit Lena	Bezug: Kalender
Vorbedienung:	Monat und Jahr kann ausgewählt werden
Beschreibung:	Kalender sollte tabellarisch ausgegeben werden
Nachbedienung:	Kalender wird tabellarisch ausgegeben

Testfall: 7	Titel: Bilder hochladen
Datum: 23.12.2016	Art: Funktionale Anforderung
Autor: Endrit Lena	Bezug: Kalender
Vorbedienung:	Templates sind erstellt und können ausgewählt werden
Beschreibung:	Bilder sollten hochgeladen werden und in Templates ausgegeben werden
Nachbedienung:	Bilder werden hochgeladen und in Templates ausgegeben

Testfall: 8	Titel: Kalender exportieren
Datum: 23.12.2016	Art: Funktionale Anforderung
Autor: Endrit Lena	Bezug: Kalender
Vorbedienung:	Kalender wird erstellt
Beschreibung:	Kalender sollte als PDF exportiert werden können
Nachbedienung:	Funktioniert nicht

Testfall: 9	Titel: Template erstellen
Datum: 23.12.2016	Art: Funktionale Anforderung
Autor: Endrit Lena	Bezug: Kalender
Vorbedienung:	
Beschreibung:	Templates sollten selber vom Benutzer erstellt werden können
Nachbedienung:	Funktioniert nicht

Testfall: 10	Titel: Texte erfassen
Datum: 23.12.2016	Art: Funktionale Anforderung
Autor: Endrit Lena	Bezug: Kalender
Vorbedienung:	Kalender wurde erstellt
Beschreibung:	Texte sollten im Kalender erfasst werden können
Nachbedienung:	Funktioniert nicht

Testfall: 11	Titel: Bilder werden skaliert
Datum: 23.12.2016	Art: Funktionale Anforderung
Autor: Endrit Lena	Bezug: Kalender
Vorbedienung:	Bilder könne hochgeladen werden
Beschreibung:	Bilder sollten sich automatisch skalieren
Nachbedienung:	Bilder skalieren sich automatisch

Testfall: 12		Titel: Java als Programmiersprache	
Datum: 23.12.2016		Art: Nicht Funktionale Anforderung	
Autor: Endrit Lena		Bezug: Kalender	
Vorbedienung:			
Beschreibung:		Java sollte als Programmiersprache verwendet werden	
Nachbedienung:		Java wird als Programmiersprache verwendet	

Testfall: 13		Titel: OpenSource Bibliothek	
Datum: 23.12.2016		Art: Nicht Funktionale Anforderung	
Autor: Endrit Lena		Bezug: Kalender	
Vorbedienung:			
Beschreibung:		OpenSource Bibliothek sollte für PDF export verwendet werden	
Nachbedienung:		OpenSource Bibliothek wird für PDF export verwendet	

Code Erläuterung

KalenderAusgabe

```
/*
 *
 * Dies ist die KalenderAusgabe Klasse. Hier wird der Kalender ausgegeben
 * inkl. Bilder.
 */
public class KalenderAusgabe extends Controller {

    @FXML
    public Text monat;
    @FXML
    public Text jahr;
    @FXML
    public Text jahr1;
    @FXML
    public Pane pane1;
    @FXML
    public Pane pane2;
    @FXML
    public Pane pn3;
    @FXML
    public Pane pn4;
    @FXML
    public ImageView iv1;
    @FXML
    public ImageView iv2;
    @FXML
    public ImageView iv3;
    @FXML
    public Pane kalender;
    @FXML
    public GridPane gp;
    @FXML
    public Button export;

    /*
     * in kal() wir der monat und das jahr von der Kalenderklasse übernommen
     * und hier ausgegeben. Ausserdem werden noch die gespeichertent Bilder
     * hier ausgelesen und ausgegeben. Dann wird noch der Kalender ausgegeben.
     */
    public void kal(String monat, String jahr) {
        this.monat.setText(monat);           //hier wird der Monat ausgegeben
        this.jahr.setText(jahr);             //hier wird das Jahr ausgegeben
        this.monat.setTextAlignment(TextAlignment.CENTER);

        //Hier wird kontrolliert ob man einen oder zwei Templates verwendet
        if (Kalender.template == true) {
            pn3.setVisible(true);
            Image i = new Image("/ch/csbe/bilder/bilder.png");
            iv1.setImage(i);
        }
    }
}
```

```

    } else if (Kalender.template == false) {
        pn4.setVisible(true);
        Image i = new Image("/ch/csbe/bilder/bild1.png");
        Image im = new Image("/ch/csbe/bilder/bild2.png");
        iv2.setImage(i);
        iv3.setImage(im);
    }

    //Hier wird kontrolliert welchen Monat man auswählt
    int mon = 0;
    switch(monat){
        case"Januar": mon = 0; break;
        case"Februar": mon = 1; break;
        case"März": mon = 2; break;
        case"April": mon = 3; break;
        case"Mai": mon = 4; break;
        case"Juni": mon = 5; break;
        case"Juli": mon = 6; break;
        case"August": mon = 7; break;
        case"September": mon = 8; break;
        case"Oktober": mon = 9; break;
        case"November": mon = 10; break;
        case"Dezember": mon = 11; break;
        default : mon = 0;
    }

    //hier wird der Kalender erstellt und ausgegeben
    GregorianCalendar cal = new GregorianCalendar(Integer.parseInt(jahr),mon,1);

    int day = cal.get(Calendar.DAY_OF_WEEK);
    cal.set(Calendar.DAY_OF_MONTH, -(7-day));
    for (int i = 0; i < 6; i++) {
        for (int j = 0; j < 7; j++) {
            Text t = new Text(getDate(cal));

            GridPane.setRowIndex(t, i);
            GridPane.setColumnIndex(t, j);
            gp.getChildren().addAll(t);
            cal.add(Calendar.DATE, 1);
        }
    }
}

//hier wird das datum ausgerechnet
public String getDate(GregorianCalendar cal) {
    SimpleDateFormat fmt = new SimpleDateFormat("dd-MMM-yyyy");
    fmt.setCalendar(cal);
    return fmt.format(cal.getTime());
}

```

```
}
```

```
//weiter() wechselt zur Kalender Klasse
```

```
@FXML
```

```
protected void weiter() {
```

```
    Kalender k = new Kalender();
```

```
    new Navigator().navigate(k);
```

```
}
```

```
}
```

Kalender

```
/**
 *
 * Dies ist die Kalender Klasse. Hier werden Monat, Jahr und
 * Templates zur Verfügung gestellt und Bilder können noch
 * ausgewählt werden
 *
 */
public class Kalender extends Controller {

    public static boolean template = true;

    @FXML
    public ComboBox<String> cb1;
    @FXML
    public ComboBox<String> cb3;
    @FXML
    public Button tm1;
    @FXML
    public Button tm2;
    @FXML
    public ImageView img;
    @FXML
    public ImageView img1;
    @FXML
    public ImageView img2;
    @FXML
    public Pane pn1;
    @FXML
    public Pane pn11;
    @FXML
    public Pane pn2;
    @FXML
    public Pane pn22;
    @FXML
    public Pane pane3;
    @FXML
    public Pane pane4;
    @FXML
    public Pane pane5;

    final FileChooser fileChooser = new FileChooser();

    //Hier kann man einen Monat auswählen
    @FXML
    protected void cb1() {
        cb1.setItems(FXCollections.observableArrayList("Januar", "Februar", "März", "April",
"Mai", "Juni", "Juli",
"August", "September", "Oktober", "November", "Dezember"));
    }

    //durch benutzung von tm1() wird nur ein Template ausgewählt
```

```

@FXML
protected void tm1() {
    template = true;
    pn1.setVisible(true);
    pn11.setVisible(true);
    if (pn2.isVisible() == true) {
        pn2.setVisible(false);
        pn22.setVisible(false);
    }
}

//durch benutzung von tm2() werden zwei Templates ausgewählt
@FXML
protected void tm2() {
    template = false;
    pn11.setVisible(true);
    pn2.setVisible(true);
    pn22.setVisible(true);
    if (pn1.isVisible() == true) {
        pn1.setVisible(false);
    }
}

//Hier kann man ein Jahr auswählen
@FXML
protected void cb3() {
    cb3.setItems(FXCollections.observableArrayList("2016", "2017", "2018", "2019",
"2020"));
}

//mit browse() kann man für tm1() ein bild auswählen
@FXML
protected void browse() {

    if (pn1.isVisible() == true) {
        FileChooser fileChooser = new FileChooser();

        FileChooser.ExtensionFilter extFilterJPG = new
FileChooser.ExtensionFilter("JPG files (*.jpg)", "*.JPG");
        FileChooser.ExtensionFilter extFilterPNG = new
FileChooser.ExtensionFilter("PNG files (*.png)", "*.PNG");
        fileChooser.getExtensionFilters().addAll(extFilterJPG, extFilterPNG);

        File file = fileChooser.showOpenDialog(null);

        try {
            BufferedImage bufferedImage = ImageIO.read(file);
            Image image = SwingFXUtils.toFXImage(bufferedImage, null);
            img.setImage(image);
            img.setFitWidth(200);
            img.setFitHeight(200);

            File f = new File("src/ch/csbe/bilder/bilder.png");

```

```

        if (!f.exists()) {
            f.mkdir();
        }
        if (f != null) {
            try {
                ImageIO.write(bufferedImage, "png", f);
            } catch (IOException ex) {

            }
        }
    } catch (IOException ex) {
        Logger.getLogger(Kalender.class.getName()).log(Level.SEVERE, null,
ex);
    }
} else if (pn2.isVisible() == true) {
    FileChooser fileChooser = new FileChooser();

    FileChooser.ExtensionFilter extFilterJPG = new
FileChooser.ExtensionFilter("JPG files (*.jpg)", "*.JPG");
    FileChooser.ExtensionFilter extFilterPNG = new
FileChooser.ExtensionFilter("PNG files (*.png)", "*.PNG");
    fileChooser.getExtensionFilters().addAll(extFilterJPG, extFilterPNG);

    File file = fileChooser.showOpenDialog(null);

    try {
        BufferedImage bufferedImage = ImageIO.read(file);
        Image image = SwingFXUtils.toFXImage(bufferedImage, null);
        img1.setImage(image);
        img1.setFitWidth(200);
        img1.setFitHeight(200);

        File f = new File("src/ch/csbe/bilder/bild1.png");
        if (!f.exists()) {
            f.mkdir();
        }
        if (f != null) {
            try {
                ImageIO.write(bufferedImage, "png", f);
            } catch (IOException ex) {

            }
        }
    } catch (IOException ex) {
        Logger.getLogger(Kalender.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
}
}

```

```

//mit browse2() kann man für tm2() zwei Bilder auswählen
@FXML
protected void browse2() {
    FileChooser fileChooser = new FileChooser();

    FileChooser.ExtensionFilter extFilterJPG = new FileChooser.ExtensionFilter("JPG files (*.jpg)", "*.JPG");
    FileChooser.ExtensionFilter extFilterPNG = new FileChooser.ExtensionFilter("PNG files (*.png)", "*.PNG");
    fileChooser.getExtensionFilters().addAll(extFilterJPG, extFilterPNG);

    File file = fileChooser.showOpenDialog(null);

    try {
        BufferedImage bufferedImage = ImageIO.read(file);
        Image image = SwingFXUtils.toFXImage(bufferedImage, null);
        img2.setImage(image);
        img2.setFitWidth(200);
        img2.setFitHeight(200);

        File f = new File("src/ch/csbe/bilder/bild2.png");
        if (!f.exists()) {
            f.mkdir();
        }
        if (f != null) {
            try {
                ImageIO.write(bufferedImage, "png", f);
            } catch (IOException ex) {
            }
        }
    } catch (IOException ex) {
        Logger.getLogger(Kalender.class.getName()).log(Level.SEVERE, null, ex);
    }
}

//Durch weiter() wird zu KalenderAusgabe gewechselt
protected void weiter() {

    KalenderAusgabe ka = new KalenderAusgabe();
    new Navigator().navigate(ka);

    String monat = cb1.getSelectionModel().getSelectedItem().toString();
    String jahr = cb3.getSelectionModel().getSelectedItem().toString();
    ka.kal(monat, jahr);
}
}

```


Fazit

Dieses Projekt gefiel mir gut. Es hat mir Spass gemacht den Kalender zu programmieren. Dennoch finde ich schade, dass das Exportieren nicht funktioniert und dass es Probleme mit den Bildern speichern und ausgeben hat. Sehr froh war ich als ich geschafft habe aus der Combobox auszulesen und diese dann auf der anderen Scene auszugeben.