

---

## Thema: Barrierefreiheit mit Java

### Vorkenntnisse

Für diese Lehreinheit benötigen die Lernenden bereits fundierte Kenntnisse in der Programmierung mit Java. Das Verständnis von objektorientierter Programmierung, sowie das Handling von Klassen und Objekten sind unbedingt vorauszusetzen. Zudem wäre es sehr vorteilhaft, wenn die Schüler bereits den Umgang mit der Java-Dokumentation geübt haben, sich darin zurechtfinden und Informationen zu unbekannten Klassen, Methoden und Funktionen nachlesen können. Eine weitere Voraussetzung ist, dass die Lernenden bereits Erfahrung mit Swing-Komponenten und Event-Handling gemacht haben. Im Zuge der folgenden Lehreinheit müssen die Lernenden selbstständig graphische Oberflächen entwickeln und Schaltflächen unterschiedliche Funktionen zuweisen. Je nach verwendeter Entwicklungsumgebung werden auch Kenntnisse über den Umgang mit Terminals oder Kommando-Fenstern benötigt um die erstellten Dateien kompilieren bzw. ausführen zu können.

Grundsätzlich wäre diese Lehreinheit auch ohne Vorkenntnisse in vielen dieser Bereichen umsetzbar, allerdings müsste dafür der Zeitrahmen massiv erweitert werden und auch die Anordnung der Lehrinhalte etwas überarbeitet werden. Hinsichtlich der Befähigung von Schülern Software barrierefrei gestalten zu können wäre es sogar sinnvoll diese Fertigkeiten von Beginn an in den Entwicklungsprozess mit einzubeziehen. Somit können diese als „normales“ Standardverfahren in den Wissensstandard der Lernenden übergehen und werden nicht als „zusätzliches“ Feature von Software betrachtet.

Da das erste Kapitel dieser Arbeit allerdings gezeigt hat, dass dieser Weg an österreichischen Schulen nicht wirklich umgesetzt wird, soll die folgende Lehreinheit eine Möglichkeit bieten diese Informationen dennoch vermitteln zu können.

## Stundenaufbau

Zeit (min)	Thema / Inhalte	Lernziele/ Kompetenzerwerb	Medien	Methoden
5	Präsentieren des Themas; Stundenablauf beschreiben		Tafel, Beamer, Papier, Internet, Lernplattform, Computer	Präsentation/Moderation
20	Arbeitsauftrag: Grundinterface (Auswahlfenster) erstellen	Festigung bisher gelernter Inhalte  Eigenständiges Arbeiten	Computer, Beamer, Papier, Tafel, Lernplattform	Präsentation/Moderation, Aufgabenstellung, praktisches Arbeiten,
15	Diskussion über Barrierefreiheit und Interfacedesign  Worauf ist zu achten?  Welche Barrieren hat das vorliegende Beispiel?	Verstehen was Barrierefreiheit bedeutet und warum sie wichtig ist  Barrieren identifizieren und Lösungswege finden können	Tafel, Beamer, Papier, Internet, Lernplattform, Computer	Präsentation/Moderation, Frage/Antwort, Diskussion
40	Code schrittweise barrierefrei machen: - Accessible- Informationen - Mnemonics - Tooltips	Mittel um Barrierefreiheit zu schaffen kennen und umsetzen lernen.  Interface selbstständig barrierefrei machen können	Tafel, Beamer, Papier, Internet, Lernplattform, Computer	Präsentation/Moderation, praktisches Arbeiten, Frage/Antwort, selbstgesteuertes Lernen, Aufgabenstellung
15	Simulator-Tool erstellen um Accessible-Daten auszulesen	Wissen wie Barrierefreiheit überprüft werden kann Verstehen warum Accessible-Daten benötigt werden	Tafel, Beamer, Papier, Internet, Lernplattform, Computer	Präsentation/Moderation, Aufgabenstellung, praktisches Arbeiten, selbstgesteuertes Lernen,
5	Analyse und Abschlussdiskussion	Festigung der Informationen, Besprechen der Erfahrungen und Erkenntnisse	Tafel, Papier	Moderation, Diskussion

## Beschreibung und Analyse

Diese Lehreinheit baut stark auf bereits erworbenen Kompetenzen der Schüler auf und versucht diese durch Informationen zur Barrierefreiheit zu ergänzen. Hierzu wird von den Lernenden auch sehr viel eigenständige Arbeit verlangt, was sich bereits nach einer kurzen Einleitung bzw. Einstiegspräsentation des Lehrenden zeigt.

Es soll ein Auswahlfenster, welches aus drei Buttons sowie einer Combo-Box und einem Textfeld, letztere mit jeweils zugehörigem Label, erstellt werden. Das Gesamtlayout sowie die Funktionen und Inhalte der verschiedenen Elemente wird zuvor genau definiert und den Lernenden über ein Handout, eine Lernplattform, die Tafel oder einen Beamer kommuniziert. Je nach Wissensstand der Gruppe können zuvor bereits Anhaltspunkte gegeben, mögliche Fehlerquellen besprochen oder sogar bestimmte Codeabschnitte zur Verfügung gestellt werden. Hierbei ist es wichtig darauf zu achten, dass die Lernenden von dieser Aufgabe nicht zu sehr überfordert werden, da sonst der eigentliche Fokus, die Barrierefreiheit, zu sehr aus dem Mittelpunkt gedrängt wird.

Nachdem diese erste Aufgabe erledigt wurde, folgt ein kurzer theoretischer Informationsblock. Dieser kann wieder kreativ gestaltet werden und viele Hilfsmittel miteinbeziehen. Neben der allgemeinen Besprechung des Begriffes, des Einsatzgebietes und der betroffenen Zielgruppe ist es besonders wichtig auf die barrierefreie Gestaltung von Java-Applikationen und den diesbezüglichen Handlungsbedarf beim vorliegenden Beispiel einzugehen. Die Lernenden sollen hierbei befähigt werden die Hürden selbst zu erkennen und die richtigen Lösungswege einzuschlagen.

Der größte Block dieser Lehreinheit beschäftigt sich mit der anschließenden Umgestaltung des Programmcodes. Hierbei sollte schrittweise vorgegangen werden und immer genügend Zeit zur Vorführung und Beschreibung der neuen Inhalte eingeplant werden. In einem ersten Schritt könnten beispielsweise Accessible-Informationen zu den unterschiedlichen Komponenten definiert werden. Diese sind wichtig, da sie die Navigation mit assistiven Technologien, wie etwa Screenreadern, erleichtern. Anschließend wäre es sinnvoll Mnemonics, welche die Steuerbarkeit mittels Tastatur erheblich verbessern, zu beschreiben und diese für bestimmte Programmfunktionen zu setzen. Abschließend können noch Tooltips definiert werden, welche die Programmoberfläche beschreiben und somit den Lernerfolg des Anwenders beschleunigen. Während die Veränderungen, die Tooltips und Mnemonics mit sich bringen, sehr leicht und offensichtlich nachzuvollziehen sind, können die Vorteile von Accessible-Informationen nicht so einfach erfasst werden, da sie im Hintergrund arbeiten und keine visuelle Bestätigung liefern.

Aus diesem Grund wird der folgende Teil der Unterrichtseinheit dafür verwendet einen Simulator zu programmieren, welcher die Funktionsweise einer assistiven Technologie simuliert und versucht Informationen zu verschiedenen Komponenten abzufragen. Je nach verbleibender Zeit können auch hier wieder Hilfestellungen oder Codeabschnitte bereitgestellt werden. Dieses zusätzliche Programm sollte den Lernenden die Funktionsweise und Bedeutung von Accessible-Informationen veranschaulichen und somit näher bringen.

Zum Abschluss dieser Lehreinheit werden die gelernten und umgesetzten Inhalte nochmals besprochen und etwaige Fragen diesbezüglich geklärt.

Im Zuge dieser Unterrichtseinheit wird von den Schülern eine Java-Applikation entwickelt, welche anschließend hinsichtlich ihrer Barrierefreiheit überprüft und erweitert wird. Nachfolgend soll dieser Prozess schrittweise anhand von Codebeispielen beschrieben werden.

### Auswahlfenster: nicht-barrierefreier Code

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class Auswahlfenster extends JFrame implements ActionListener
{
    JButton exitButton;
    JButton bigButton;
    JButton littleButton;

    public Auswahlfenster()
    {
        super("Auswahlfenster");
        setSize(420, 180);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container pane = getContentPane();

        //Die Combo-Box ###
        JComboBox noten = new JComboBox();
        noten.addItem("Sehr Gut");
        noten.addItem("Gut");
        noten.addItem("Befriedigend");
```

---

```
noten.addItem("Genuegend");
noten.addItem("Nicht Genuegend");
noten.setEditable(false);

// Label für die Combo-Box ###
JLabel combolabel = new JLabel();
combolabel.setText("Schulnoten");

//Textfeld ###
JTextArea feedback = new JTextArea(5, 20);

//Label für das Textfeld ###
JLabel feedbacklabel = new JLabel();
feedbacklabel.setText("Feedback");

//Der "Fenster vergroessern"-Button ###
bigButton = new JButton("G");
bigButton.addActionListener(this);

//Der "Fenster verkleinern"-Button ###
littleButton = new JButton("K");
littleButton.addActionListener(this);

//Der "Fenster schließen"-Button ###
exitButton = new JButton("X");
exitButton.addActionListener(this);

//Bildschirmbereiche aufteilen
JPanel top = new JPanel();
top.add(bigButton);
top.add(littleButton);
top.add(exitButton);
JPanel center = new JPanel();
```

---

```

        center.add(combolabel);
        center.add(noten);
        JPanel bottom = new JPanel();
        bottom.add(feedbacklabel);
        bottom.add(feedback);

        //Alles richtig verpacken
        pane.setLayout(new BoxLayout(pane, BoxLayout.PAGE_AXIS));
        pane.add("North", top);
        pane.add("Center", center);
        pane.add("South", bottom);

        setVisible(true);
        setContentPane(pane);
    }

    public void actionPerformed(ActionEvent e)
    {

        Object quelle = e.getSource();

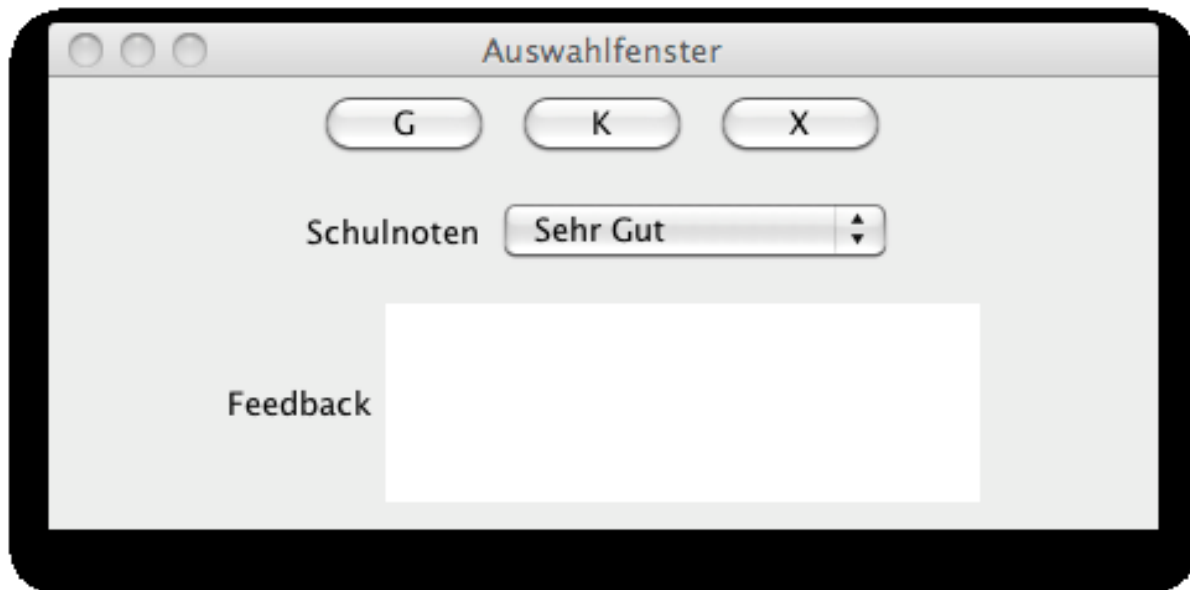
        if (quelle == exitButton) System.exit(0);
        if (quelle == bigButton) setSize(800,800);
        if (quelle == littleButton) setSize(400,60);

    }

    public static void main(String[] arguments)
    {
        Auswahlfenster app = new Auswahlfenster();
    }
}

```

Aus dem angeführten Code resultiert folgendes Programmfenster:



Java I

Die Programmoberfläche besteht aus drei Buttons im oberen Bereich. Diese beeinflussen bei Betätigung die optische Form des Programmfensters. Deren Funktionen sind (in Reihenfolge von links nach rechts): vergrößern, verkleinern und schließen. Im mittleren Bereich befinden sich eine Combo-Box sowie ein zugehöriges Label und am unteren Fensterrand ein Textfeld, ebenfalls mit zugehörigem Label. Die Combo-Box und das Textfeld bieten zwar die Möglichkeit zur Auswahl eines Elements bzw. zur Texteingabe, haben darüber hinaus allerdings keine weiterführende Funktion. Insgesamt dienen alle Elemente nur dazu die Umsetzung von Maßnahmen zur Barrierefreiheit zu üben.

### Auswahlfenster: Accessible-Informationen

Im ersten Schritt sollen allen relevanten Komponenten Accessible-Informationen hinzugefügt werden um assistiven Technologien die Kommunikation damit zu erleichtern. Die hierfür benötigten zusätzlichen Codeabschnitte werden im folgenden Listing durch die Färbung des Texthintergrundes, sowie Fettschrift hervorgehoben.

```
import java.awt.*;  
import javax.swing.*;  
import java.awt.event.*;  
import javax.accessibility.*;
```

```
public class Auswahlfenster extends JFrame implements ActionListener
{
    JButton exitButton;
    JButton bigButton;
    JButton littleButton;

    public Auswahlfenster()
    {
        super("Auswahlfenster");
        setSize(420, 180);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container pane = getContentPane();

        //Die Combo-Box ###
        JComboBox noten = new JComboBox();
        noten.addItem("Sehr Gut");
        noten.addItem("Gut");
        noten.addItem("Befriedigend");
        noten.addItem("Genuegend");
        noten.addItem("Nicht Genuegend");
        noten.setEditable(false);
        //Accessible-Informationen setzen
        AccessibleContext ac = noten.getAccessibleContext();
        ac.setAccessibleDescription("Waehle eine Schulnote aus der Liste aus");
        ac.setAccessibleName("Schulnoten");

        // Label für die Combo-Box ###
        JLabel combolabel = new JLabel();
        combolabel.setText(ac.getAccessibleName());

        //Textfeld ###
        JTextArea feedback = new JTextArea(5, 20);
        //Accessible-Informationen setzen
        AccessibleContext ac_2 = feedback.getAccessibleContext();
        ac_2.setAccessibleDescription("Bitte gib hier dein Feedback ein");
        ac_2.setAccessibleName("Feedback");
    }
}
```



```
//Label für das Textfeld ###
JLabel feedbacklabel = new JLabel();
feedbacklabel.setText(ac_2.getAccessibleName());

//Der "Fenster vergroessern"-Button ###
bigButton = new JButton("G");
bigButton.addActionListener(this);
//Accessible-Informationen setzen
bigButton.getAccessibleContext().setAccessibleDescription("Das Fenster vergroessern");
bigButton.getAccessibleContext().setAccessibleName("vergroessern");

//Der "Fenster verkleinern"-Button ###
littleButton = new JButton("K");
littleButton.addActionListener(this);
//Accessible-Informationen setzen
littleButton.getAccessibleContext().setAccessibleDescription("Das Fenster verkleinern");
littleButton.getAccessibleContext().setAccessibleName("verkleinern");

//Der "Fenster schließen"-Button ###
exitButton = new JButton("X");
exitButton.addActionListener(this);
//Accessible-Informationen setzen
exitButton.getAccessibleContext().setAccessibleDescription("Das Fenster schliessen");
exitButton.getAccessibleContext().setAccessibleName("schliessen");

(...)
```

Zunächst ist es wichtig die benötigten Ressourcen mittels `import javax.accessibility.*;` zu importieren. Anschließend müssen die Komponenten mit den Accessible-Informationen versehen werden. Am Beispiel der Combo-Box sieht das wie folgt aus:

---

```
//Accessible-Informationen setzen
AccessibleContext ac = noten.getAccessibleContext();
ac.setAccessibleDescription("Waehle eine Schulnote aus der Liste aus");
ac.setAccessibleName("Schulnoten");
```

Nachdem ein `AccessibleContext`-Objekt erzeugt wurde, welches in der Folge die `Accessible-Informationen` verwaltet, werden diesem durch die Funktionen `.setAccessibleDescription()` und `.ac.setAccessibleName()` eine Beschreibung der Komponente bzw. dessen Name übergeben.

Eine andere Möglichkeit diese Informationen zu setzen, wird bei den Buttons gezeigt. Hier wird das `AccessibleContext`-Objekt nicht extra erzeugt, sondern direkt mit Informationen beschrieben:

```
//Accessible-Informationen setzen
exitButton.getAccessibleContext().setAccessibleDescription("Das Fenster
schliessen");
exitButton.getAccessibleContext().setAccessibleName("schliessen");
```

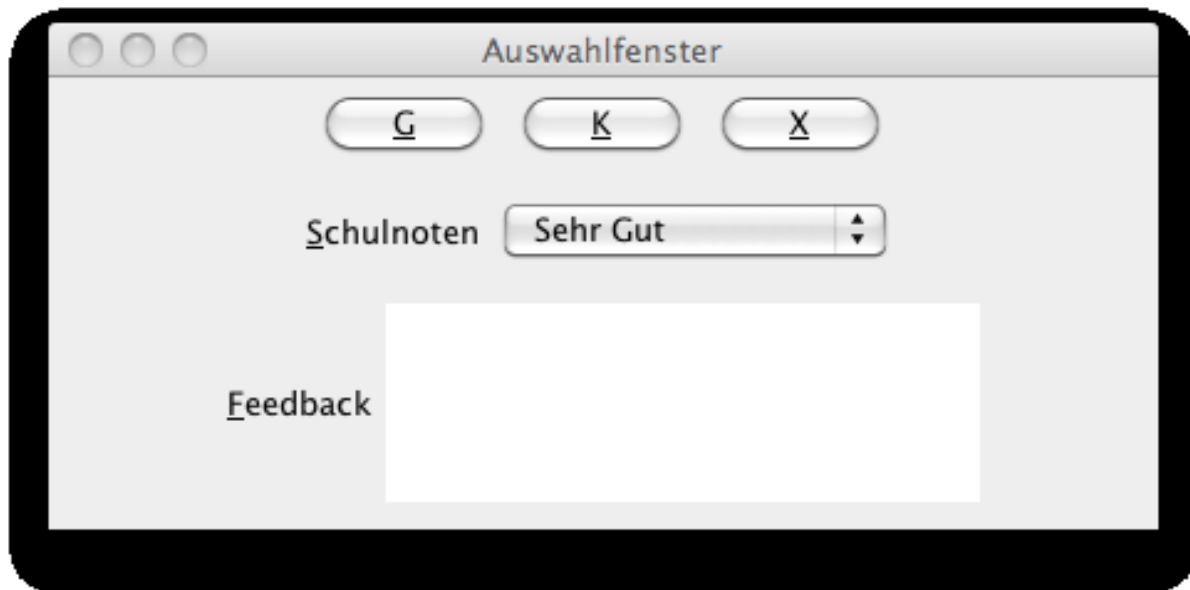
Eine zusätzliche Veränderung gab es auch noch bei den Labels. Anstatt den Namen des Labels direkt einzugeben, wird nun auf die bereits gesetzten `Accessible-Informationen` der zugehörigen Komponente zurückgegriffen:

```
//Label für das Textfeld ###
JLabel feedbacklabel = new JLabel();
feedbacklabel.setText(ac_2.getAccessibleName());
```

Dies zeigt einerseits, wie `Accessible-Informationen` ausgelesen werden können und bestätigt andererseits die Zusammengehörigkeit der beiden Elemente durch den nun sicher immer gleichen Namen.

## Auswahlfenster: Mnemonics

Nachdem die `Accessible-Informationen` gesetzt wurden, sollen nun `Mnemonics` erstellt werden um die Navigation mittels Tastatur zu erleichtern. Es werden also für jede relevante Komponente, mit Ausnahme der Labels, `Shortcuts` angelegt um sie direkt ansteuern zu können. Das Ausgabefenster sollte anschließend folgendermaßen aussehen:



## Java II

(...)

```
//Die Combo-Box ###
JComboBox noten = new JComboBox();
noten.addItem("Sehr Gut");
noten.addItem("Gut");
noten.addItem("Befriedigend");
noten.addItem("Genuiegend");
noten.addItem("Nicht Genuiegend");
noten.setEditable(false);
//Accessible-Informationen setzen
AccessibleContext ac = noten.getAccessibleContext();
ac.setAccessibleDescription("Waehle eine Schulnote aus der Liste aus");
ac.setAccessibleName("Schulnoten");

// Label für die Combo-Box ###
JLabel combolabel = new JLabel();
combolabel.setText(ac.getAccessibleName());
//Mnemonic setzen
combolabel.setDisplayedMnemonic(KeyEvent.VK_S);
combolabel.setLabelFor(noten);
```

```
//Textfeld ###
JTextArea feedback = new JTextArea(5, 20);
//Accessible-Informationen setzen
AccessibleContext ac_2 = feedback.getAccessibleContext();
ac_2.setAccessibleDescription("Bitte gib hier dein Feedback ein");
ac_2.setAccessibleName("Feedback");

//Label für das Textfeld ###
JLabel feedbacklabel = new JLabel();
feedbacklabel.setText(ac_2.getAccessibleName());
//Mnemonic setzen
feedbacklabel.setDisplayedMnemonic(KeyEvent.VK_F);
feedbacklabel.setLabelFor(feedback);

//Der "Fenster vergrößern"-Button ###
bigButton = new JButton("G");
bigButton.addActionListener(this);
//Accessible-Informationen setzen
bigButton.getAccessibleContext().setAccessibleDescription("Das Fenster vergrößern");
bigButton.getAccessibleContext().setAccessibleName("vergrößern");
//Mnemonic setzen
bigButton.setMnemonic('g');

//Der "Fenster verkleinern"-Button ###
littleButton = new JButton("K");
littleButton.addActionListener(this);
//Accessible-Informationen setzen
littleButton.getAccessibleContext().setAccessibleDescription("Das Fenster verkleinern");
littleButton.getAccessibleContext().setAccessibleName("verkleinern");
//Mnemonic setzen
littleButton.setMnemonic('k');

//Der "Fenster schließen"-Button ###
exitButton = new JButton("X");
```

```
exitButton.addActionListener(this);  
//Accessible-Informationen setzen  
exitButton.getAccessibleContext().setAccessibleDescription("Das Fenster  
schliessen");  
exitButton.getAccessibleContext().setAccessibleName("schliessen");  
//Mnemonic setzen  
exitButton.setMnemonic('x');  
  
(...)
```

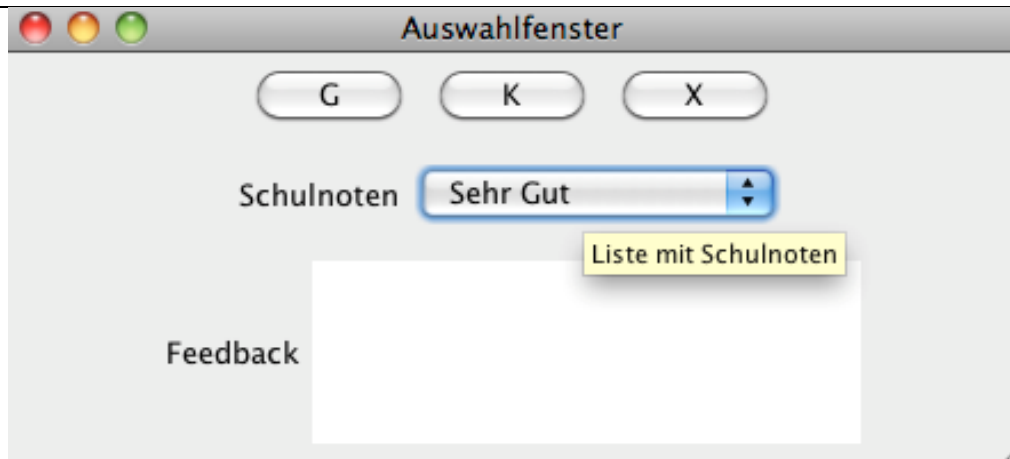
Der angeführte Programmcode zeigt unterschiedliche Methoden Mnemonics zu setzen. Bei den Buttons wurden sie mit der Funktion `.setMnemonic()` gesetzt und als Parameter eine Charakter-Variable übergeben. Dies hat zur Folge, dass Java automatisch nach der ersten Instanz dieser Variable im Namen der Komponente sucht und diese anschließend unterstrichen darstellt. Beim Textfeld und der Combo-Box werden die Mnemonics durch die Funktion `.setDisplayMnemonic()` angelegt. Außerdem wird hier keine Charakter-Variable, sondern ein Platzhalter für ein Key-Event (z.B.: `KeyEvent.VK_F`) übergeben. Zusätzlich zu den Mnemonics wird dem Programmcode an zwei Stellen auch noch die Verknüpfung der Labels mit der zugehörigen Komponente beigelegt:

```
combolabel.setLabelFor(noten); und feedbacklabel.setLabelFor(feedback);
```

Dies ermöglicht es bei Aktivierung der Mnemonics der Labels direkt die zugehörige Komponente in den Programmfokus zu bringen.

#### 7.5.4. Auswahlfenster: Tooltips + Simulator

Im letzten Schritt sollen dem Auswahlfenster noch Tooltips hinzugefügt werden:



Java III

(...)

```
//Die Combo-Box ###
JComboBox noten = new JComboBox();
noten.addItem("Sehr Gut");
noten.addItem("Gut");
noten.addItem("Befriedigend");
noten.addItem("Genuiegend");
noten.addItem("Nicht Genuiegend");
noten.setEditable(false);
//Accessible-Informationen setzen
AccessibleContext ac = noten.getAccessibleContext();
ac.setAccessibleDescription("Waehle eine Schulnote aus der Liste aus");
ac.setAccessibleName("Schulnoten");
//ToolTip setzen
noten.setToolTipText("Liste mit Schulnoten");

// Label für die Combo-Box ###
JLabel combolabel = new JLabel();
combolabel.setText(ac.getAccessibleName());
//Mnemonic setzen
combolabel.setDisplayedMnemonic(KeyEvent.VK_S);
combolabel.setLabelFor(noten);

//Textfeld ###
```

---

```

        JTextArea feedback = new JTextArea(5, 20);
        //Accessible-Informationen setzen
        AccessibleContext ac_2 = feedback.getAccessibleContext();
        ac_2.setAccessibleDescription("Bitte gib hier dein Feedback ein");
        ac_2.setAccessibleName("Feedback");
        //ToolTip setzen
        feedback.setToolTipText("Hier koennen Sie Feedback eintragen");

        //Label für das Textfeld ###
        JLabel feedbacklabel = new JLabel();
        feedbacklabel.setText(ac_2.getAccessibleName());
        //Mnemonic setzen
        feedbacklabel.setDisplayedMnemonic(KeyEvent.VK_F);
        feedbacklabel.setLabelFor(feedback);

        //Der "Fenster vergroessern"-Button ###
        JButton bigButton = new JButton("G");
        bigButton.addActionListener(this);
        //Accessible-Informationen setzen
        bigButton.getAccessibleContext().setAccessibleDescription("Das Fenster
vergroessern");
        bigButton.getAccessibleContext().setAccessibleName("vergroessern");
        //Mnemonic setzen
        bigButton.setMnemonic('g');
        //ToolTip setzen
        bigButton.setToolTipText("Das Fenster vergroessern");

        //Der "Fenster verkleinern"-Button ###
        JButton littleButton = new JButton("K");
        littleButton.addActionListener(this);
        //Accessible-Informationen setzen
        littleButton.getAccessibleContext().setAccessibleDescription("Das
Fenster verkleinern");
        littleButton.getAccessibleContext().setAccessibleName("verkleinern");
        //Mnemonic setzen
        littleButton.setMnemonic('k');
        //ToolTip setzen
    
```

```
littleButton.setToolTipText("Das Fenster verkleinern");

//Der "Fenster schließen"-Button ###
exitButton = new JButton("X");
exitButton.addActionListener(this);
//Accessible-Informationen setzen
exitButton.getAccessibleContext().setAccessibleDescription("Das Fenster
schliessen");
exitButton.getAccessibleContext().setAccessibleName("schliessen");
//Mnemonic setzen
exitButton.setMnemonic('x');
//ToolTip setzen
exitButton.setToolTipText("Das Fenster schliessen");

(...)

pane.add("South", bottom);

setVisible(true);
setContentPane(pane);

//Simulator
Simulator sim = new Simulator(noten, exitButton, bigButton, littleButton,
combolabel, feedbacklabel, feedback);
}

(...)
```

Das Setzen der Tooltips erfolgt durch den Einsatz der Funktion `.setToolTipText()`, welche jeder Komponente hinzugefügt werden kann und den später am Bildschirm enthaltenen Text enthält. In diesem Listing wurde zusätzlich noch die abschließende Erzeugung des Simulator-Objektes angefügt, welchem alle verwendeten Komponenten als Parameter übergeben werden. Die Funktionsweise dieses Simulators wird im folgenden Listing beschrieben.



## Gesamter Code: Simulator

```
import java.awt.*;
import javax.swing.*;
import javax.accessibility.*;

public class Simulator implements Runnable
{
    Thread runner;
    JComboBox schulnoten;
    JButton exitButton;
    JButton bigButton;
    JButton littleButton;
    JLabel combolabel;
    JLabel feedbacklabel;
    JTextArea feedback;

    Simulator(JComboBox not, JButton ex, JButton big, JButton lit, JLabel com,
    JLabel fbl, JTextArea fb)
    {
        schulnoten = not;
        exitButton = ex;
        bigButton = big;
        littleButton = lit;
        combolabel = com;
        feedbacklabel = fbl;
        feedback = fb;
        if (runner == null)
        {
            runner = new Thread(this);
            runner.start();
        }
    }

    public void run()
    {
        AccessibleContext ac_1 = schulnoten.getAccessibleContext();
```

---

```

AccessibleContext ac_2 = exitButton.getAccessibleContext();
AccessibleContext ac_3 = bigButton.getAccessibleContext();
AccessibleContext ac_4 = littleButton.getAccessibleContext();
AccessibleContext ac_5 = combolabel.getAccessibleContext();
AccessibleContext ac_6 = feedbacklabel.getAccessibleContext();
AccessibleContext ac_7 = feedback.getAccessibleContext();

System.out.println("\nSchulnoten:");
System.out.println("Name: " + ac_1.getAccessibleName());
System.out.println("Beschreibung: " + ac_1.getAccessibleDescription());
System.out.println("Funktion: " + ac_1.getAccessibleRole());

System.out.println("\nexitButton:");
System.out.println("Name: " + ac_2.getAccessibleName());
System.out.println("Beschreibung: " + ac_2.getAccessibleDescription());
System.out.println("Funktion: " + ac_2.getAccessibleRole());

System.out.println("\nbigButton:");
System.out.println("Name: " + ac_3.getAccessibleName());
System.out.println("Beschreibung: " + ac_3.getAccessibleDescription());
System.out.println("Funktion: " + ac_3.getAccessibleRole());

System.out.println("\nlittleButton:");
System.out.println("Name: " + ac_4.getAccessibleName());
System.out.println("Beschreibung: " + ac_4.getAccessibleDescription());
System.out.println("Funktion: " + ac_4.getAccessibleRole());

System.out.println("\nComboLabel:");
System.out.println("Name: " + ac_5.getAccessibleName());
System.out.println("Beschreibung: " + ac_5.getAccessibleDescription());
System.out.println("Funktion: " + ac_5.getAccessibleRole());

System.out.println("\nFeedbackFeld:");
System.out.println("Name: " + ac_7.getAccessibleName());
System.out.println("Beschreibung: " + ac_7.getAccessibleDescription());
System.out.println("Funktion: " + ac_7.getAccessibleRole());

System.out.println("\nFeedbackLabel:");
System.out.println("Name: " + ac_6.getAccessibleName());
System.out.println("Beschreibung: " + ac_6.getAccessibleDescription());

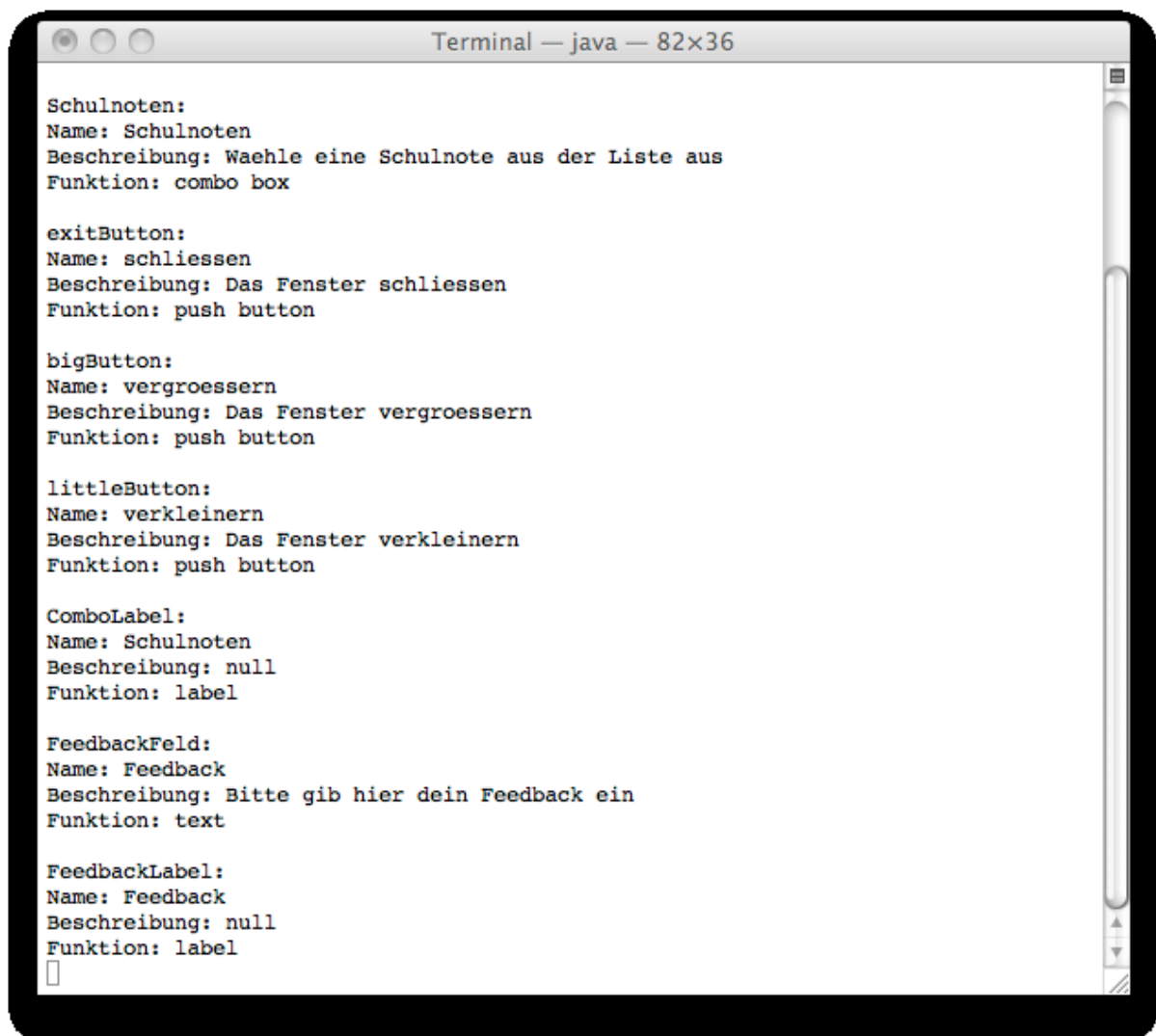
```

```
System.out.println("Funktion: " + ac_6.getAccessibleRole());
```

```
}
```

```
}
```

Die Aufgabe dieses Programmes ist es die Accessible-Informationen des Auswahlfensters abzufragen. Es simuliert also eine assistive Technologie, wie beispielsweise einen Screenreader, welche Zugriff auf diese Komponenten benötigt. Hierzu werden alle Komponenten unter Verwendung der Funktionen `.getAccessibleName()`, `.getAccessibleDescription()` und `.getAccessibleRole()` untersucht und die gefundenen Informationen in der Konsole ausgegeben, wie folgendes Bild zeigt:



```
Terminal — java — 82x36

Schulnoten:
Name: Schulnoten
Beschreibung: Waehle eine Schulnote aus der Liste aus
Funktion: combo box

exitButton:
Name: schliessen
Beschreibung: Das Fenster schliessen
Funktion: push button

bigButton:
Name: vergroessern
Beschreibung: Das Fenster vergroessern
Funktion: push button

littleButton:
Name: verkleinern
Beschreibung: Das Fenster verkleinern
Funktion: push button

ComboLabel:
Name: Schulnoten
Beschreibung: null
Funktion: label

FeedbackFeld:
Name: Feedback
Beschreibung: Bitte gib hier dein Feedback ein
Funktion: text

FeedbackLabel:
Name: Feedback
Beschreibung: null
Funktion: label
□
```

Java IV

## Gesamter Code: Auswahlfenster

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import javax.accessibility.*;

public class Auswahlfenster extends JFrame implements ActionListener
{
    JButton exitButton;
    JButton bigButton;
    JButton littleButton;

    public Auswahlfenster()
    {
        super("Auswahlfenster");
        setSize(420, 180);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container pane = getContentPane();

        //Die Combo-Box ###
        JComboBox noten = new JComboBox();
        noten.addItem("Sehr Gut");
        noten.addItem("Gut");
        noten.addItem("Befriedigend");
        noten.addItem("Genuegend");
        noten.addItem("Nicht Genuegend");
        noten.setEditable(false);
        //Accessible-Informationen setzen
        AccessibleContext ac = noten.getAccessibleContext();
        ac.setAccessibleDescription("Waehle eine Schulnote aus der Liste aus");
        ac.setAccessibleName("Schulnoten");
        //ToolTip setzen
        noten.setToolTipText("Liste mit Schulnoten");

        // Label für die Combo-Box ###
        JLabel combolabel = new JLabel();
```

---

```

        combolabel.setText(ac.getAccessibleName());
//Mnemonic setzen
        combolabel.setDisplayedMnemonic(KeyEvent.VK_S);
        combolabel.setLabelFor(noten);

//Textfeld ###
        JTextArea feedback = new JTextArea(5, 20);
//Accessible-Informationen setzen
        AccessibleContext ac_2 = feedback.getAccessibleContext();
        ac_2.setAccessibleDescription("Bitte gib hier dein Feedback ein");
        ac_2.setAccessibleName("Feedback");
//ToolTip setzen
        feedback.setToolTipText("Hier koennen Sie Feedback eintragen");

//Label für das Textfeld ###
        JLabel feedbacklabel = new JLabel();
        feedbacklabel.setText(ac_2.getAccessibleName());
//Mnemonic setzen
        feedbacklabel.setDisplayedMnemonic(KeyEvent.VK_F);
        feedbacklabel.setLabelFor(feedback);

//Der "Fenster vergroessern"-Button ###
        bigButton = new JButton("G");
        bigButton.addActionListener(this);
//Accessible-Informationen setzen
        bigButton.getAccessibleContext().setAccessibleDescription("Das Fenster vergroessern");
        bigButton.getAccessibleContext().setAccessibleName("vergroessern");
//Mnemonic setzen
        bigButton.setMnemonic('g');
//ToolTip setzen
        bigButton.setToolTipText("Das Fenster vergroessern");

//Der "Fenster verkleinern"-Button ###
        littleButton = new JButton("K");
        littleButton.addActionListener(this);

```

---

```
//Accessible-Informationen setzen
littleButton.getAccessibleContext().setAccessibleDescription("Das
Fenster verkleinern");
littleButton.getAccessibleContext().setAccessibleName("verkleinern");
//Mnemonic setzen
littleButton.setMnemonic('k');
//ToolTip setzen
littleButton.setToolTipText("Das Fenster verkleinern");

//Der "Fenster schließen"-Button ###
exitButton = new JButton("X");
exitButton.addActionListener(this);
//Accessible-Informationen setzen
exitButton.getAccessibleContext().setAccessibleDescription("Das Fenster
schliessen");
exitButton.getAccessibleContext().setAccessibleName("schliessen");
//Mnemonic setzen
exitButton.setMnemonic('x');
//ToolTip setzen
exitButton.setToolTipText("Das Fenster schliessen");

//Bildschirmbereiche aufteilen
JPanel top = new JPanel();
top.add(bigButton);
top.add(littleButton);
top.add(exitButton);
JPanel center = new JPanel();
center.add(combolabel);
center.add(noten);
JPanel bottom = new JPanel();
bottom.add(feedbacklabel);
bottom.add(feedback);

//Alles richtig verpacken
pane.setLayout(new BoxLayout(pane, BoxLayout.PAGE_AXIS));
pane.add("North", top);
pane.add("Center", center);
pane.add("South", bottom);

setVisible(true);
```

---

```
        setContentPane (pane);

        //Simulator
        Simulator sim = new Simulator(noten, exitButton, bigButton, littleButton,
        combolabel, feedbacklabel, feedback);

    }

    public void actionPerformed(ActionEvent e)
    {

        Object quelle = e.getSource();

        if (quelle == exitButton) System.exit(0);
        if (quelle == bigButton) setSize(800,800);
        if (quelle == littleButton) setSize(400,60);

    }

    public static void main(String[] arguments)
    {
        Auswahlfenster app = new Auswahlfenster();
    }
}
```