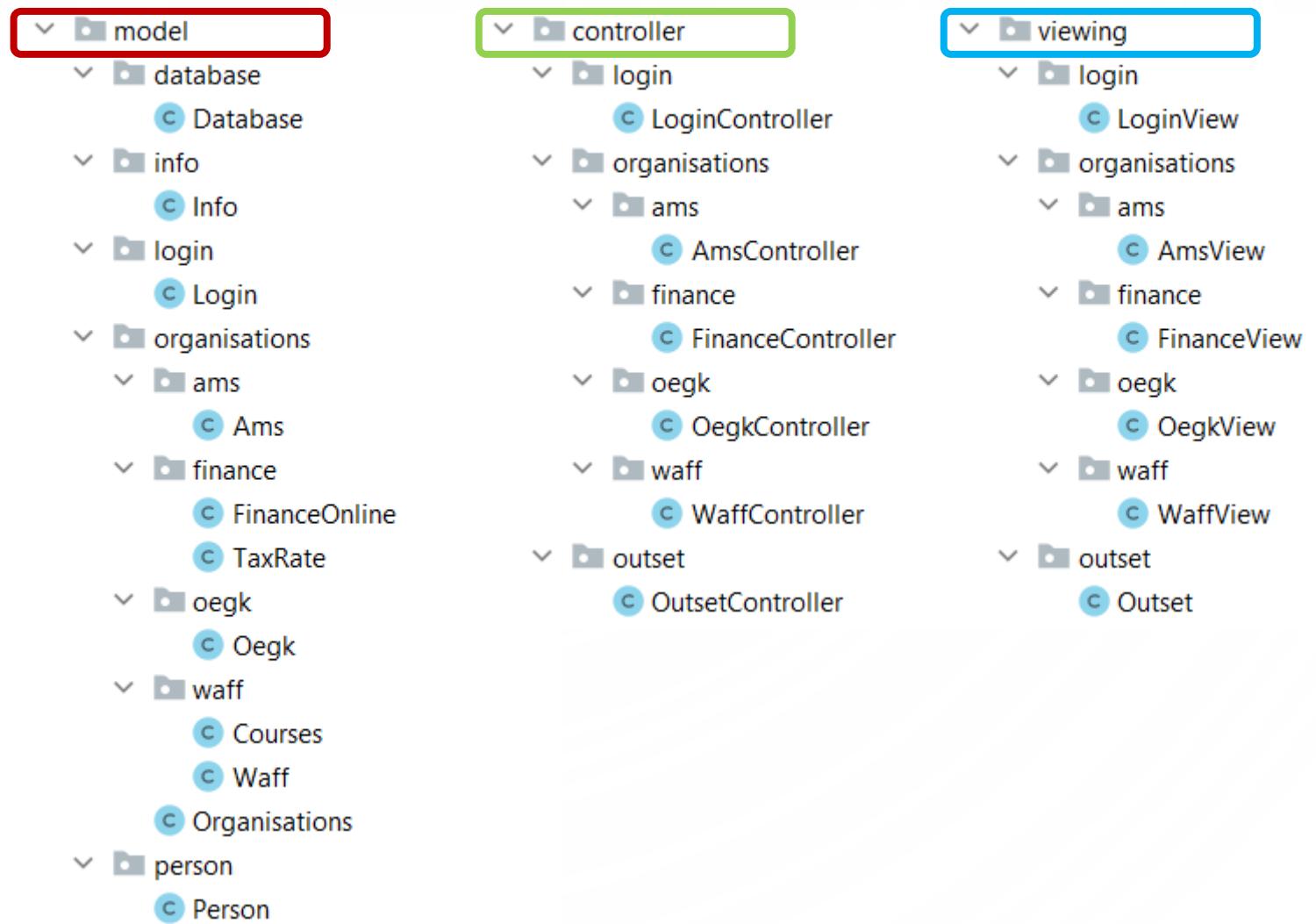
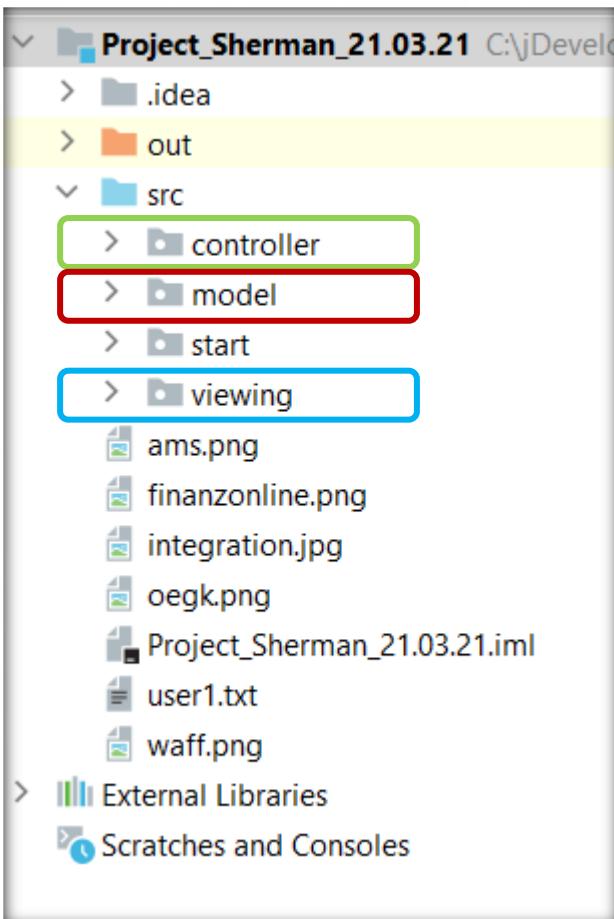
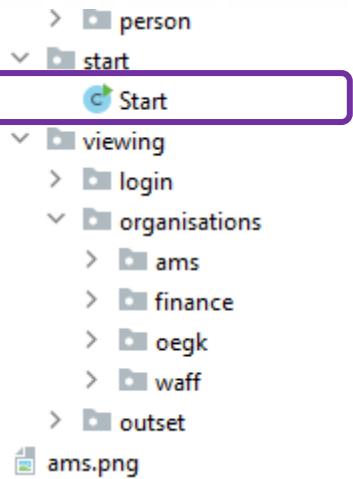


InfoAustria

Project structure | MCV



1st Step | Login



```
11  public class Start {  
12    public static void main(String[] args){  
13      //=====GUI=====  
14      new LoginView();  
15      new Outset();  
16      new AmsView();  
17      new OegkView();  
18      new WaffView();  
19      //  
20    }  
21  }
```

The screenshot shows a Java Swing application window titled "Login to InfoAustria 23.03.2021". The window contains the following fields:

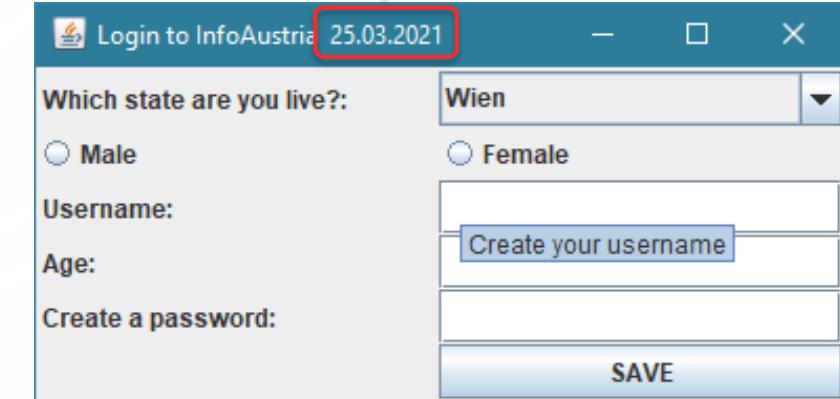
- Which state are you live?: Wien (selected)
- Male (radio button)
- Female (radio button)
- Username: (empty text field)
- Age: (empty text field)
- Create a password: (empty text field)
- SAVE (button at the bottom right)

Login | View

```
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setTitle("Login to InfoAustria " + new Info().getDate());
this.setSize( width: 400, height: 200);
this.setLocationRelativeTo(null);
this.setVisible(true);
```

```
//DateGetter-----
public String getDate() {
    return new SimpleDateFormat( pattern: "dd.MM.yyyy").format(new Date());
}
```

In title it will be always the **current date**.



```
//CSS-
ageText.addKeyListener(new KeyAdapter() {
    @Override
    public void keyPressed(KeyEvent e) {
        info.numbersValidation(e,ageText);
    }
});
```

```
//Tool Tip
nameText.setToolTipText("Create your userna
passwordField.setToolTipText("Create your I
save.setToolTipText("Save for creating a ne
```

```
//validation Method-----
public void numbersValidation (KeyEvent e, JTextField jTextField){
    jTextField.setEditable((e.getKeyChar() >= '0' && e.getKeyChar() <= '9')
                           || (e.getKeyCode() == KeyEvent.VK_BACK_SPACE));
```

As „Age“ you can write only numbers.

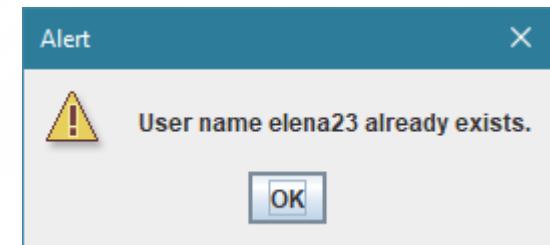
Login | Controller

```
//function Controller---  
} else if (e.getSource() == save) {  
    if (nameText.getText().equals("") || ageText.getText().equals("") || passwordField.getText().equals("")) {  
        JFrame alert = new JFrame();  
        JOptionPane.showMessageDialog(alert, message: "Please fill in all fields", title: "Alert",  
        JOptionPane.WARNING_MESSAGE);  
    }  
    else if (!male.isSelected() && !female.isSelected()) {  
        JFrame alert = new JFrame();  
        JOptionPane.showMessageDialog(alert, message: "Please choose your gender", title: "Alert",  
        JOptionPane.WARNING_MESSAGE);  
    }  
    else {  
        //saving a User---  
        state = Objects.requireNonNull(stateBox.getSelectedItem()).toString();  
        name = nameText.getText();  
        age = Integer.parseInt(ageText.getText());  
        password = passwordField.getText().toCharArray();  
  
        if (login.getName().equals(name.toLowerCase(Locale.ROOT))) {  
            JFrame alert = new JFrame();  
            JOptionPane.showMessageDialog(alert, message: "User name " + name + " already exists.", title: "Alert",  
            JOptionPane.WARNING_MESSAGE);  
        } else {  
            person.setId(id++);  
            login.userSave(id, name.toLowerCase(Locale.ROOT), age, gender, state, password);  
  
            person = new Person(id, name, age, gender, state, password);  
            System.out.println(person.toString());  
  
            //go to OutsetMenu---  
            new Outset();  
            loginView.setVisible(false);  
        }  
    }  
}
```

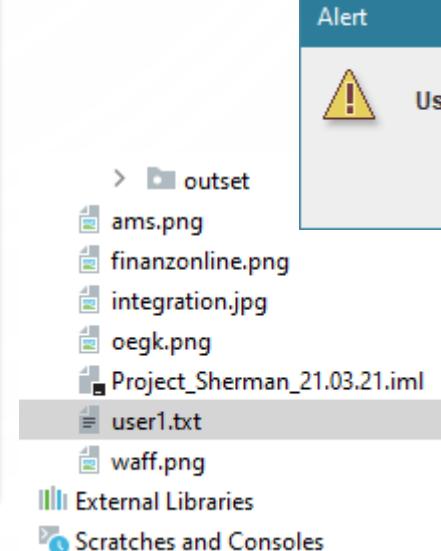
User must fill in all fields or he will **get a alert msg.**



If „user name“ already exists in **user1.txt**,
User will get alert msg and be **not saved**.



Program saves User data and go
to the **next Menu**.



Login | code Logic

```
//FileWriter-----  
public void userSave(long id, String name, int age, char gender, String state, char[] password) {  
    try {  
        FileWriter user = new FileWriter(fileName: "user1.txt");  
        user.write(str: id + " " + name + " " + age + " " + gender + " " + state + " " + password);  
        user.close();  
    } catch (Exception e) {  
        System.out.println(e);  
    }  
    System.out.println("User " + name + " is saved");  
}  
  
//FileReader-----  
public void userRead() throws Exception {  
    FileReader fr = new FileReader(fileName: "user1.txt");  
    int i;  
    while ((i = fr.read()) != -1)  
        System.out.print((char) i);  
    fr.close();  
}
```

With **Login** program can:

- **save** and **read** a new user;
- **get** main information, which were saved.

```
//InfoGetter Methods  
public int getAge() {  
    String[] userInfo = {};  
    try (FileReader fileReader = new FileReader(fileName: "user1.txt")) {  
        try (BufferedReader bufferedReader = new BufferedReader(fileReader)) {  
            String user = bufferedReader.readLine();  
            userInfo = user.split(regex: "[ ]+");  
        }  
    } catch (Exception e1) {  
        e1.printStackTrace();  
    }  
    int age = Integer.parseInt(userInfo [2]);  
    return age;  
}  
  
public String getGender() {  
    String[] userInfo = {};  
    try (FileReader fileReader = new FileReader(fileName: "user1.txt")) {  
        try (BufferedReader bufferedReader = new BufferedReader(fileReader)) {  
            String user = bufferedReader.readLine();  
            userInfo = user.split(regex: "[ ]+");  
        }  
    } catch (Exception e1) {  
        e1.printStackTrace();  
    }  
    String gender = userInfo [3];  
    return gender;  
}
```

Login | Two ways to save User data

```
public void userSave(long id, String name, int age, char gender, String state, char[] password) {  
    try {  
        FileWriter user = new FileWriter( fileName: "user1.txt");  
        user.write( str: id + " " + name + " " + age + " " + gender + " " + state + " " + password);  
        user.close();  
    } catch (Exception e) {  
        System.out.println(e);  
    }  
    System.out.println("User " + name + " is saved");  
}
```

```
=====TXT Writer-Reader=====  
login.userSave(id, name.toLowerCase(Locale.ROOT), age, gender, state, password);  
person = new Person(id, name, age, gender, state, password);  
System.out.println(person.toString());
```

Program saves User data in **.txt file** with **FileWriter Class**.

```
=====SERIALIZATION=====  
login.userCreate(id, name.toLowerCase(Locale.ROOT), age, gender, state, password);
```

```
//go to OutsetMenu--  
new Outset();  
loginView.setVisible(false);
```

Program saves the **Object Person** by **Serialization**.

```
=====SERIALIZATION=====  
//Serialization-----  
public void userCreate(long id, String name, int age, char gender, String state, char[] password) {  
    try {  
        Person person = new Person(id, name, age, gender, state, password);  
        ObjectOutputStream fileInOut = new ObjectOutputStream(new FileOutputStream( name: "user2.ser"));  
        fileInOut.writeObject(person);  
        fileInOut.flush();  
        fileInOut.close();  
        System.out.println("\nSerialized data are saved in user2.ser");  
    } catch (Exception ex) {  
        ex.printStackTrace();  
    }  
}
```

Login | Two ways to get User data

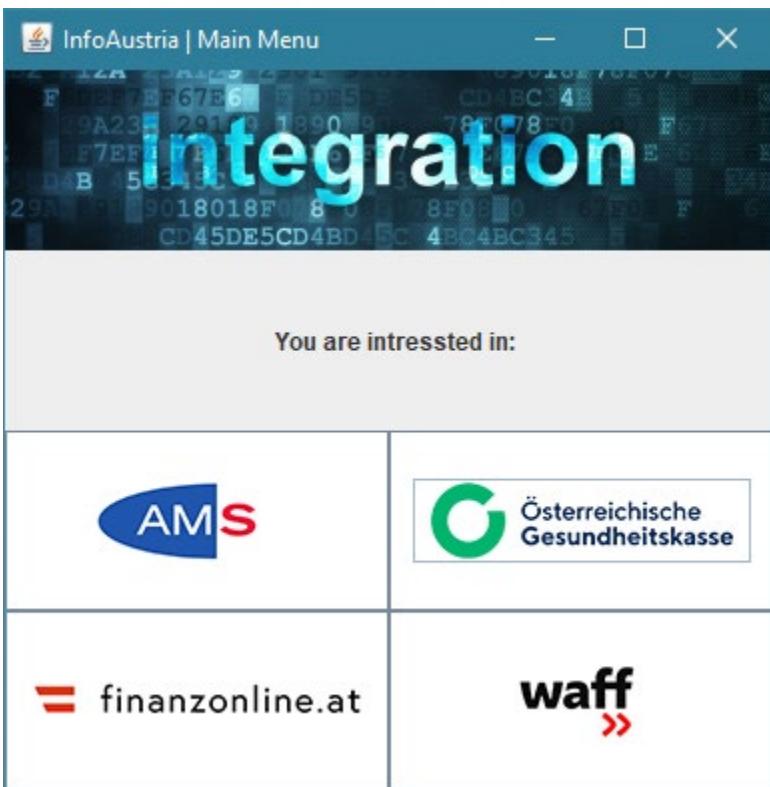
```
//FileReader-----  
public void userRead() throws Exception {  
    FileReader fr = new FileReader( fileName: "user1.txt");  
    int i;  
    while ((i = fr.read()) != -1)  
        System.out.print((char) i);  
    fr.close();  
}  
  
//InfoGetter Methods-----  
public String getName() {  
    String[] userInfo = {};  
    try (FileReader fileReader = new FileReader( fileName: "user1.txt")) {  
        try (BufferedReader bufferedReader = new BufferedReader(fileReader)) {  
            String user = bufferedReader.readLine();  
            userInfo = user.split( regex: "[ ]+");  
        }  
    } catch (Exception e1) {  
        e1.printStackTrace();  
    }  
    String name = userInfo[1];  
    return name;  
}
```

FileReader Class: splitting the data and read as Array.

```
//InfoGetter Methods-----  
public String getName2() {  
    Person person;  
    try {  
        ObjectInputStream fileIn = new ObjectInputStream  
            (new FileInputStream( name: "user2.ser"));  
        person = (Person) fileIn.readObject();  
        fileIn.close();  
        return person.getName();  
    } catch (IOException ex) {  
        ex.printStackTrace();  
    } catch (ClassNotFoundException c) {  
        c.printStackTrace();  
    }  
    return "";  
}
```

Deserialization: input the whole object with its properties.

Main Menu | Oufset



JButtons are the way to **other Menus.**

```
public Outset() {  
    //JLabels  
    JLabel wellcomeLabel;  
    JLabel moreAboutLabel = new JLabel( text: " You are interested in: ", SwingConstants.CENTER);  
  
    //CSS--  
    ImageIcon imageIcon = new ImageIcon( filename: "integration.jpg");  
    Image image = imageIcon.getImage();  
    Image newimg = image.getScaledInstance( width: 400, height: 300, java.awt.Image.SCALE_SMOOTH);  
    imageIcon = new ImageIcon(newimg);  
  
    wellcomeLabel = new JLabel(imageIcon);
```

scaled image for JLabel

```
private JButton ams = new JButton (new ImageIcon( filename: "ams.png"));  
private JButton finanzonline = new JButton (new ImageIcon( filename: "finanzonline.png"));  
private JButton oegk = new JButton(new ImageIcon( filename: "oegk.png"));  
private JButton waff = new JButton(new ImageIcon( filename: "waff.png"));
```

JButtons as Icons

```
//ActionPerformed Method-----  
@Override  
public void actionPerformed(ActionEvent e) {  
    if (e.getSource() == ams) { new AmsView(); }  
  
    else if (e.getSource() == finanzonline) { new FinanceView(); }  
  
    else if (e.getSource() == oegk) { new OegkView(); }  
  
    else if (e.getSource() == waff) { new WaffView(); }  
}
```

Menu | AMS

InfoAustria | AMS - 23.03.2021

In order to receive unemployment benefits, you must register as unemployed in right time.

Are you registered in Austria? Yes No

Did you have unemployment insurance in Austria?

Did you work in Austria?

Do you want to apply to unemployment benefits for the first time? Yes No

Continue

Calculator

You started your last job:

You ended your last job:

Still working

Back to Main Menu **Calculate**

```
if(firstTimeBox.isSelected()) {  
    infoAMS.setFont(new Font( name: "Abyssinica SIL",Font.ITALIC, size: 14));  
    infoAMS.setText(String.format("<html><body style=\"text-align: center \" "+  
        "\">%s</body></html>",ams.calculateWork1(workBegin, workEnd, login.getAge())));  
}  
else if(secondTimeBox.isSelected()){  
    infoAMS.setFont(new Font( name: "Abyssinica SIL",Font.ITALIC, size: 14));  
    infoAMS.setText(String.format("<html><body style=\"text-align: center \" "+  
        "\">%s</body></html>",ams.calculateWork2(workBegin, workEnd)));  
}
```

interactive JLabel

```
//ActionListener via Anonymous class-----  
backButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        new Outset();  
        setVisible(false);  
    }  
});
```

„BackButton“ way to Outset with **closing current Window**.

AMS | View

```
//ItemListener via Lambda-----  
stillWorking.addItemListener(e -> {  
    if (stillWorking.isSelected()) {  
        infoAMS.setFont(new Font( name: "Abyssinica SIL",Font.ITALIC, size: 14));  
        infoAMS.setForeground(Color.red);  
        infoAMS.setText(String.format("<html><body style=\"text-align: center " +  
            "\">\">%s</body></html>","You must register as unemployed on the first day of your unemployment " +  
            "at the latest. You can register earliest 3 weeks before dismissal."));  
    }else if (stillWorking.isEnabled()){  
        infoAMS.setFont(new Font( name: "Abyssinica SIL",Font.ITALIC, size: 12));  
        infoAMS.setForeground(Color.black);  
        infoAMS.setText("In order to receive unemployment benefits, you must register as unemployed in right time.");  
    }});
```

ItemListener via Lambda

```
//JLabels-----  
JLabel registeredLab = new JLabel( text: "Are you registered in Austria?");  
JLabel insuranceLab = new JLabel(String.format("<html><body style=\"text-align: left " +  
            "\">\">%s</body></html>","Did you have unemployment insurance in Austria?"));  
JLabel workedInATLab = new JLabel( text: "Did you work in Austria?");  
JLabel calculator = new JLabel ( text: "Calculator", SwingConstants.CENTER);  
JLabel workBegin = new JLabel( text: "You started your last job:");  
JLabel workEnd = new JLabel( text: "You ended your last job:");  
JLabel firstTimeLab= new JLabel(String.format("<html><body style=\"text-align: left " +  
            "\">\">%s</body></html>","Do you want to apply to unemployment benefits for the first time?"));
```

String.format for text editing

AMS | code Logic | Calendar

User can calculate the **duration** of his employment and check the **possibility** of receiving benefits using **CALENDAR**.

```
//calendar Logic - never apply for AMS-----
public String calculateWork1(LocalDate workBegin, LocalDate workEnd, int age) {
    Period p = Period.between(workBegin, workEnd);
    long p2 = ChronoUnit.DAYS.between(workBegin, workEnd);
    if (workBegin.isAfter(workEnd)) {
        return "Please correct the date of start.";
    }
    else if ((workBegin != null)&&(workEnd != null)&&(p2 > 364)){
        return "Your period of work is " + p.getYears() + " years, " + p.getMonths() +
               " months, and " + p.getDays() +
               " days. \nYou have worked enough to apply for unemployment benefits the first time.";
    }
    else if ((p2 < 364) && (age > 25)) {
        return "Your period of work is " + p.getYears() + " years, " + p.getMonths() +
               " months, and " + p.getDays() + " days. " +
               "\nIt's less than 52 Weeks." + "You must have worked for at least 52 weeks in " +
               "the last 2 years to be eligible for unemployment benefits.";
    }
    else if ((workBegin != null)&&(workEnd != null)&&(p2 < 364)&&(p2 >= 182)&&(age >= 18)&&(age <= 25)) {
        return "Your period of work is " + p.getYears() + " years, " + p.getMonths() +
               " months, and " + p.getDays() + " days. " +
               "\nYou have worked enough to apply for unemployment benefits the first time.";
    }
    return "Please provide correct dates.";
}
```

```
//calendarProperties-----
private final List<String> days= new ArrayList();
private final List<String> months= new ArrayList();
private final List<String> years= new ArrayList();

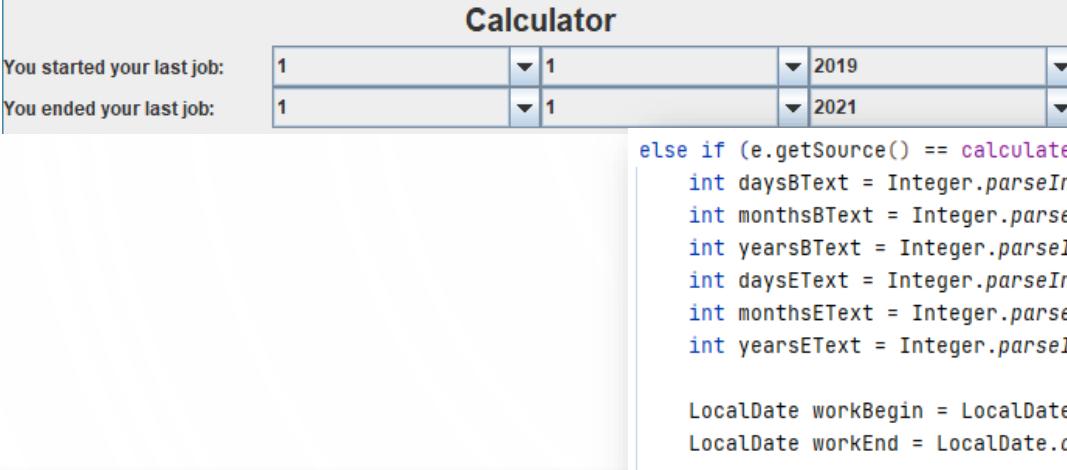
//calendarCreation Methods-
public List<String> calendarDays () {
    int currentDay =31;
    for (int i = 1; i <= currentDay; i++) {
        days.add(String.valueOf(i));
    }
    return days;
}

public List<String> calendarMonths () {
    int currentMonth = 12;
    for (int i = 1; i <= currentMonth; i++) {
        months.add(String.valueOf(i));
    }
    return months;
}

public List<String> calendarYears () {
    int currentYear = Calendar.getInstance().get(Calendar.YEAR);
    for (int i = currentYear; i > currentYear - 100; i--) {
        years.add(String.valueOf(i));
    }
    return years;
}
```

Calendar is created with **List** and **forLoops**.

AMS | Controller

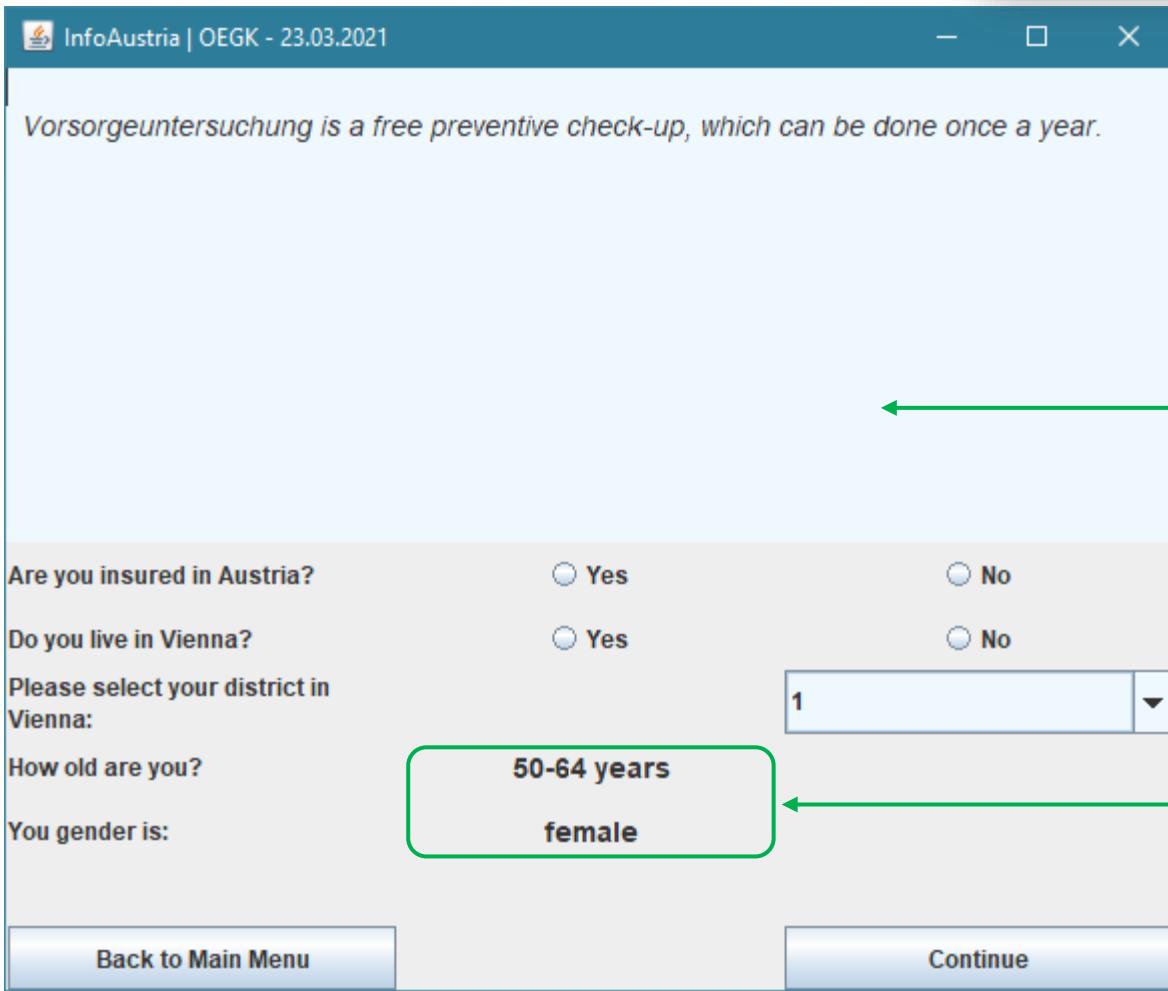


```
if (e.getSource() == yesButton) {
    amsView.setTitle("You are registered in Austria");
}
else if (e.getSource() == noButton) {
    amsView.setTitle("Sorry, you must be registered in Austria!");
    JFrame alert = new JFrame();
    JOptionPane.showMessageDialog(alert, message: "Please registered in Austria to be able to use " +
        "social system opportunities", title: "Alert", JOptionPane.WARNING_MESSAGE);
}
else if (e.getSource() == continueButton) {
    if ((insuranceBox.isSelected() && (workedInATBox.isSelected())) {
        JFrame msg = new JFrame();
        JOptionPane.showMessageDialog(msg, message: "You have opportunity to get unemployment benefits." +
            "\nPlease, use calculator to get more information.");
    } else if ((!insuranceBox.isSelected()) || (!workedInATBox.isSelected())) {
        JFrame msg = new JFrame();
        JOptionPane.showMessageDialog(msg, message: "Unfortunately, you have no opportunity " +
            "to get unemployment benefits.");
    }
}
```

Program get information from calendar via **parsing**.

Buttons' logic using **MessageDialog**

Menu | ÖGK



```
//functionProperties  
private JTextArea infoOEGK = new JTextArea("\n Vorsorgeuntersuchung is a free preventive  
done once a year.");  
private JButton continueButton = new JButton( text: "Continue");  
private JButton backButton = new JButton( text: "Back to Main Menu");
```

```
if (e.getSource() == yesIsured) {  
    infoOEGK.setText(" \nYou can get free preventive check-up with your e-card.");  
}  
else if (e.getSource() == noIsured) {  
    infoOEGK.setText (" \nYou need to get certificate marked \"Vorsorgeuntersuchung fu  
"(Medical examination for uninsured persons). You can get help by filling  
"at your city's social security customer service center.");  
}
```

interactive JTextArea

```
JLabel yourGenderLab = new JLabel(oegk.getGender());  
JLabel ageLab = new JLabel( text: "How old are you?");  
JLabel yourAgeLab = new JLabel(oegk.getAge());
```

User data was taken from user1.txt file.

ÖGK | code Logic | User Data

```
//InfoGetter Methods-----  
public int getAge() {  
    String[] userInfo = {};  
    try (FileReader fileReader = new FileReader( fileName: "user1.txt")) {  
        try (BufferedReader bufferedReader = new BufferedReader(fileReader)) {  
            String user = bufferedReader.readLine();  
            userInfo = user.split( regex: "[ ]+" );  
        }  
    } catch (Exception e1) {  
        e1.printStackTrace();  
    }  
    int age = Integer.parseInt(userInfo [2]);  
    return age;  
}  
  
public String getGender() {  
    String[] userInfo = {};  
    try (FileReader fileReader = new FileReader( fileName: "user1.txt")) {  
        try (BufferedReader bufferedReader = new BufferedReader(fileReader)) {  
            String user = bufferedReader.readLine();  
            userInfo = user.split( regex: "[ ]+" );  
        }  
    } catch (Exception e1) {  
        e1.printStackTrace();  
    }  
    String gender = userInfo [3];  
    return gender;  
}
```

```
//AgeGetter Logic Method-----  
public String getAge(){  
    if(login.getAge()<18) {  
        age = "< 18 years";  
        return age;  
    } else if((login.getAge()>=18)&&(login.getAge()<=49)) {  
        age = "18-49 years";  
        return age;  
    } else if(login.getAge()>=50) {  
        age = "50-64 years";  
        return age;  
    } else if (login.getAge()>=65){  
        age = "> 65 years";  
        return age;  
    }  
    return age;  
}
```

```
//GenderGetter Logic Method-----  
public String getGender(){  
    if(login.getGender().equals("f")) {  
        gender = "female";  
        return gender;  
    } else if (login.getGender().equals("m")){  
        gender = "male";  
        return gender;  
    }  
    return gender;  
}
```

Program uses information from Login step to **get user data**.

```
JLabel yourGenderLab = new JLabel(oegk.getGender());  
JLabel ageLab = new JLabel( text: "How old are you?");  
JLabel yourAgeLab = new JLabel(oegk.getAge());
```

ÖGK | code Logic | Offers

```
//Adding Offers to ArrayList-----  
public void addOffers() {  
    offer.add("interview to identify chronic non-communicable diseases and risk factors");  
    offer.add("measurement of height and body weight, calculation of body mass index (BMI)");  
    offer.add("heart ECG + transcript");  
    offer.add("blood pressure measurement");  
    offer.add("blood test: clinical blood test, determination of cholesterol and blood glucose levels");  
    offer.add("urine analysis");  
    offer.add("detection of gum disease");  
  
    //Offers Logic-----  
    if((login.getAge()>=65)&&(login.getGender().equals("f"))){  
        offer.add("smear for cancer");  
        offer.add("colon cancer screening, stool blood test, colonoscopy every 10 years");  
        offer.add("detection of hearing and vision impairments");  
    }  
    else if ((login.getAge()>=50)&&(login.getGender().equals("f"))){  
        offer.add("smear for cancer");  
        offer.add("colon cancer screening, stool blood test, colonoscopy every 10 years");  
    }  
    else if(login.getAge()>=65){  
        offer.add("colon cancer screening, stool blood test, colonoscopy every 10 years");  
        offer.add("detection of hearing and vision impairments");  
    }  
    else if(login.getGender().equals("f")) {  
        offer.add("smear for cancer");  
    }  
    else if(login.getAge()>=50){  
        offer.add("colon cancer screening, stool blood test, colonoscopy every 10 years");  
    }  
}
```

According to user data, program adds **special offers** from ÖGK.

```
//OfferGetter Method-----  
public String getOffer() {  
    String offerList = "";  
    for (String list : offer) {  
        offerList += "\n -> " + list;  
    }  
    return offerList; }
```

ÖGK | Controller

Program **proposes hospital**, if there is one in user's district.

```
else if (e.getSource() == districtBox) {
    if(yesLive.isSelected()) {
        String yourdistric = Objects.requireNonNull(districtBox.getSelectedItem()).toString();
        if (yourdistric.equals("3")) {
            infoOEGK.setText("\nYou have hospital in your district!\n" +
                "\nMein Gesundheitszentrum Landstrasse\n" +
                "1030 Wien, Strohgasse 28.");
        }
        else if (yourdistric.equals("6")) {
            infoOEGK.setText("\nYou have hospital in your district!\n" +
                "\nMein Gesundheitszentrum Mariahilf\n" +
                "1060 Wien, Mariahilfer Strasse 85-87.");
        }
        else if (yourdistric.equals("10")) {
            infoOEGK.setText("\nYou have hospital in your district!\n" +
                "\nMein Gesundheitszentrum Favoriten\n" +
                "1100 Wien, Wienerbergstrasse 13.");
        }
        else if (yourdistric.equals("21")) {
            infoOEGK.setText("\nYou have hospital in your district!\n" +
                "\nMein Gesundheitszentrum Floridsdorf\n" +
                "1210 Wien, Karl-Aschenbrenner-Gasse 3.");
        }
    }
}
```

```
//districtCreation Method-----
public List getDistrict () {
    int districtNo = 23;
    for (int i = 1; i <= districtNo; i++) {
        district.add(String.valueOf(i));
    }
    return district;
}
```

You have hospital in your district!
"Mein Gesundheitszentrum Landstrasse"
1030 Wien, Strohgasse 28.

Are you insured in Austria? Yes No
Do you live in Vienna? Yes No
Please select your district in Vienna:
How old are you? 50-64 years

ÖGK | View

```
//CSS  
Color myColor = new Color( r: 240, g: 248, b: 255);  
yesIsured.setHorizontalAlignment(SwingConstants.CENTER);  
noIsured.setHorizontalAlignment(SwingConstants.CENTER);  
yesLive.setHorizontalAlignment(SwingConstants.CENTER);  
noLive.setHorizontalAlignment(SwingConstants.CENTER);  
  
infoOEGK.setWrapStyleWord(true);  
infoOEGK.setLineWrap(true);  
infoOEGK.setBackground(myColor);  
infoOEGK.setFont(new Font("Abyssinica SIL", Font.ITALIC, siz
```

Using own color with RGB parameters.

You need to get certificate marked "Vorsorgeuntersuchung fuer Nicht Versicherte" (Medical examination for uninsured persons). You can get help by filling out the registration form at your city's social security customer service center.

Are you insured in Austria?

Do you live in Vienna?

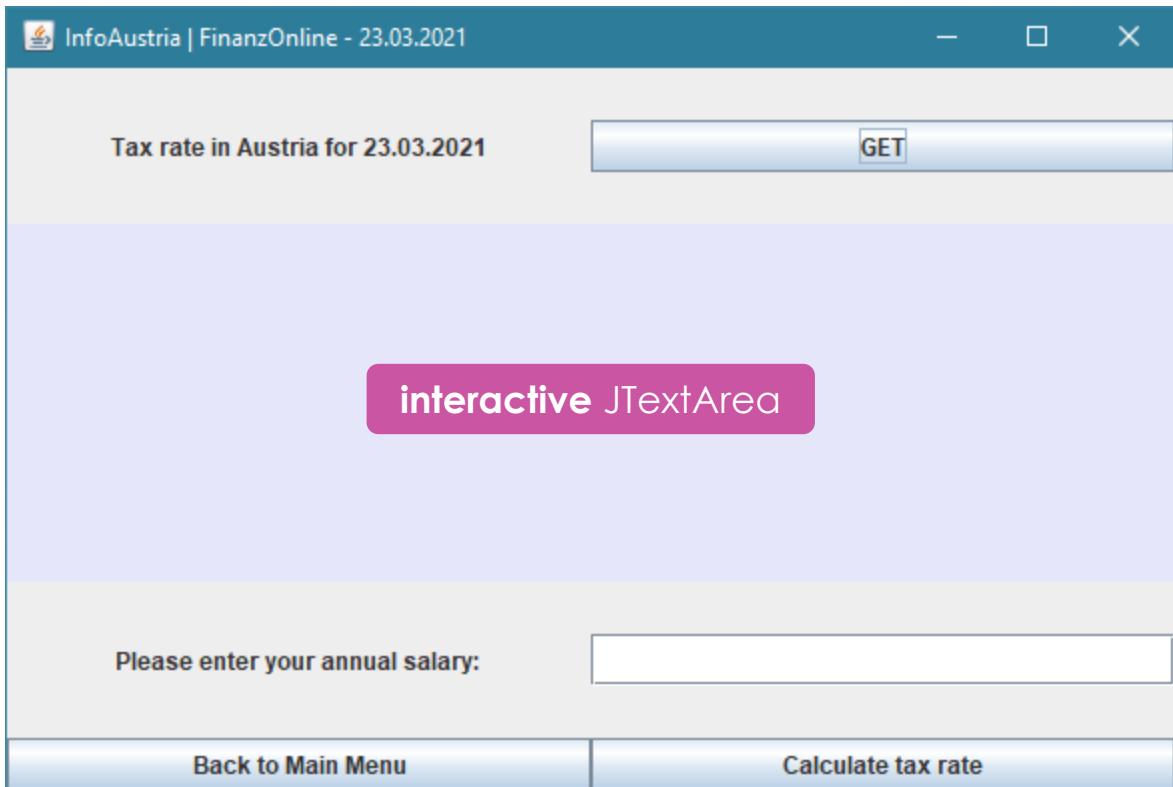
Please select your district in Vienna:

How old are you?

You gender is:

Back to Main Menu Continue

Menu | FinanzOnline

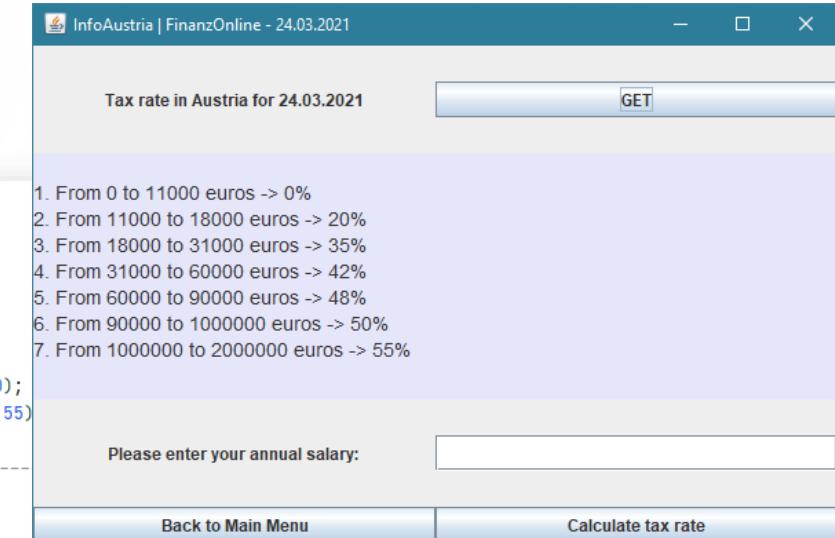


FinanzOnline | code Logic | TaxRate List

```
public class TaxRate {  
    //Properties  
    private int minIncome;  
    private int maxIncome;  
    private int rate;  
  
    //Handing over properties via Constructor  
    public TaxRate(int minIncome, int maxIncome, int rate) {...}  
  
    //Calculate Methods  
    public int betweenAmount() {...}  
  
    public int salaryTax(int pYourSalary) {...}  
  
    public int withoutTaxAmount(int maxIncome) {...}  
  
    public int percentAmount() {...}  
  
    public int percentAmount(int maxIncome) {...}  
  
    //Getter Methods  
    public int getMinIncome() { return minIncome; }  
  
    public int getMaxIncome() { return maxIncome; }  
  
    public int getRate() { return rate; }  
  
    //info Methods  
    @Override  
    public String toString() {  
        return "From " + minIncome + " to " + maxIncome  
    }  
}
```

Class **TaxRate** is using for creating the **ArrayList** of tax options.

```
//taxProperties  
private final List<TaxRate> taxRateList = new ArrayList<>();  
private TaxRate t1 = new TaxRate( minIncome: 0, maxIncome: 11000, rate: 0 );  
private TaxRate t2 = new TaxRate( minIncome: 11000, maxIncome: 18000, rate: 20 );  
private TaxRate t3 = new TaxRate( minIncome: 18000, maxIncome: 31000, rate: 35 );  
private TaxRate t4 = new TaxRate( minIncome: 31000, maxIncome: 60000, rate: 42 );  
private TaxRate t5 = new TaxRate( minIncome: 60000, maxIncome: 90000, rate: 48 );  
private TaxRate t6 = new TaxRate( minIncome: 90000, maxIncome: 1000000, rate: 50 );  
private TaxRate t7 = new TaxRate( minIncome: 1000000, maxIncome: 2000000, rate: 55 );  
  
//Adding tax rates to ArrayList  
public void addTaxRate() {  
    taxRateList.add(t1);  
    taxRateList.add(t2);  
    taxRateList.add(t3);  
    taxRateList.add(t4);  
    taxRateList.add(t5);  
    taxRateList.add(t6);  
    taxRateList.add(t7);  
}  
  
//TaxRateGetter Method  
public String getTaxRateList() {  
    int i = 1;  
    String rateList = "";  
    for (TaxRate list : taxRateList) {  
        rateList += "\n" + i++ + ". " + list;  
    }  
    return rateList;  
}
```

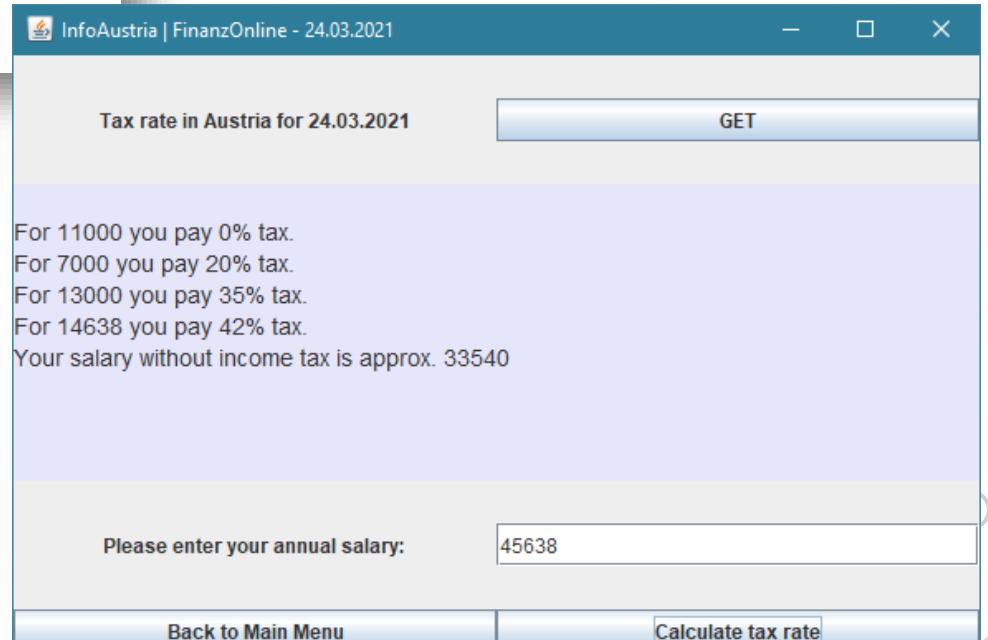


```
//ActionPerformed Method  
@Override  
public void actionPerformed(ActionEvent e) {  
    //taxRateList Controller  
    if (e.getSource() == getButton) {  
        financeOnline = new FinanceOnline();  
        financeOnline.addTaxRate();  
        infoFinance.setText(financeOnline.getTaxRateList());  
    }  
}
```

FinanzOnline | code Logic | your TaxRate

```
public String getTaxSum(int pYourSalary){  
    this.yourSalary=pYourSalary;  
  
    if(yourSalary<=11000){  
        return t1.info(t1.betweenAmount());  
    }  
    else if((11000<yourSalary)&&(yourSalary<=18000)){  
  
        return t1.info(t1.betweenAmount()) + //info about  
        "\nFor " + t2.salaryTax(yourSalary) + " you  
        "\nYour salary without income tax is approx. " + // = static sums without tax + rest of your salary with  
        (t1.betweenAmount() + t2.withoutTaxAmount(t2.salaryTax(yourSalary)));  
    }  
    else if((18000<yourSalary)&&(yourSalary<=31000)){  
        return t1.info(t1.betweenAmount()) + t2.info(t2.betweenAmount()) +  
        "\nFor " + t3.salaryTax(yourSalary) + " you pay " + t3.getRate() + "% tax." +  
        "\nYour salary without income tax is approx. " +  
        (t1.betweenAmount() + t2.withoutTaxAmount(t2.betweenAmount()) + t3.withoutTaxAmount(t3.salaryTax(yourSalary)));  
    }  
    else if((31000<yourSalary)&&(yourSalary<=60000)){  
        return t1.info(t1.betweenAmount()) + t2.info(t2.betweenAmount()) + t3.info(t3.betweenAmount()) +  
        "\nFor " + t4.salaryTax(yourSalary) + " you pay " + t4.getRate() + "% tax." +  
        "\nYour salary without income tax is approx. " +  
        (t1.betweenAmount() + t2.withoutTaxAmount(t2.betweenAmount()) + t3.withoutTaxAmount(t3.betweenAmount()) +  
        t4.withoutTaxAmount(t4.salaryTax(yourSalary)));  
    }  
    else if((60000<yourSalary)&&(yourSalary<=90000)){  
        return t1.info(t1.betweenAmount()) + t2.info(t2.betweenAmount()) + t3.info(t3.betweenAmount()) + t4.info(t4.betweenAmount()) + t5.info(t5.betweenAmount())  
        "\nFor " + t5.salaryTax(yourSalary) + " you pay " + t5.getRate() + "% tax." +  
        "\nYour salary without income tax is approx. " +  
        (t1.betweenAmount() + t2.withoutTaxAmount(t2.betweenAmount()) + t3.withoutTaxAmount(t3.betweenAmount()) +  
        t4.withoutTaxAmount(t4.betweenAmount()) + t5.withoutTaxAmount(t5.salaryTax(yourSalary)));  
    }  
}
```

User can **calculate the tax rate** of his salary, according to the logic in Class FinanceOnline.



FinanzOnline | code Logic | your TaxRate

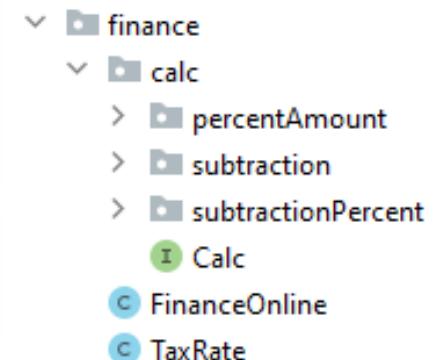
```
//Calculate Methods-----  
public int betweenAmount(){  
    Calc subtraction = new Subtraction();  
    int taxSum = subtraction.calc(maxIncome, minIncome);  
    return taxSum;  
}
```

```
public int salaryTax(int pYourSalary){  
    Calc subtraction = new Subtraction();  
    int taxSum = subtraction.calc(pYourSalary, minIncome);  
    return taxSum;  
}
```

```
public int withoutTaxAmount(int maxIncome){  
    Calc minusPercent = new SubtractionPercent();  
    int taxSum = minusPercent.calc(maxIncome, rate);  
    return taxSum;  
}
```

For the calculation of tax rate it was used **upcasting (polymorphism)** for calling the **interface Calc** which represents the **IS-A** relationship.

```
public interface Calc {  
    public abstract int calc(int x, int y);  
}
```



```
public class PercentAmount implements Calc {  
    @Override  
    public int calc(int max, int rate) {  
        return max*rate/100;  
    }  
}
```

```
public class SubtractionPercent implements Calc {  
    @Override  
    public int calc(int between, int rate) {  
        return between*(100-rate)/100;  
    }  
}
```

```
public class Subtraction implements Calc {  
    @Override  
    public int calc(int x,int y){  
        return x - y;  
    }  
}
```

Menu | waff

InfoAustria | waff - 23.03.2021

Do you live in Vienna?	<input checked="" type="radio"/> Yes	<input type="radio"/> No	
Are you currently employed in Vienna?	<input checked="" type="radio"/> Yes	<input type="radio"/> No	
Are you AK Member?	<input type="radio"/> Yes, in Vienna	<input type="radio"/> Yes, in other state	<input type="checkbox"/> No, I am not a member
Your salary per month:	<input type="radio"/> <1800 net	<input type="radio"/> <2500 net	<input type="radio"/> >2500 net

[Back to Main Menu](#) [Continue](#)

WAFF provide people in Vienna with information, advice and financial support for their professional development. In addition, they work in close cooperation with the Vienna Public Employment Service (AMS Vienna) to serve as a port of call for job-searcher and companies.

interactive List area

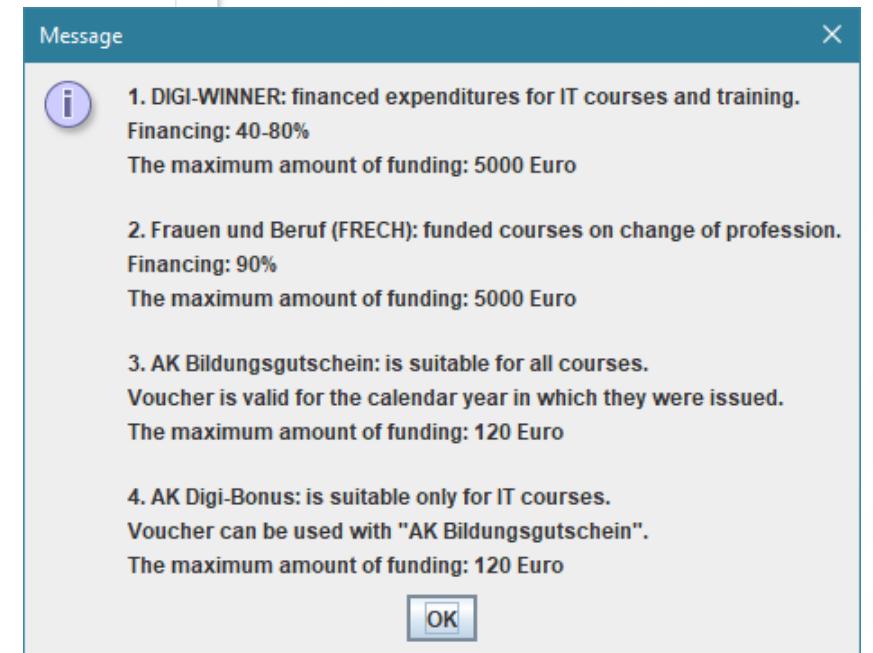
waff | code Logic | Courses

```
public class Courses {  
    //Properties--  
    private String name, financing, specialization;  
    private int maxSum;  
  
    //Handing over properties via Constructor--  
    public Courses(String name, String financing, String specialization, int maxSum) {  
        this.name = name;  
        this.financing = financing;  
        this.specialization = specialization;  
        this.maxSum = maxSum;  
    }  
  
    //toString Method--  
    public String info(){  
        return name + ": " + specialization  
            + "\n" + financing  
            + "\nThe maximum amount of funding: " + maxSum + " Euro";  
    }  
  
    //courseProperties  
    private final Map<Integer, Courses> courses = new HashMap<>();  
    private Courses c1=new Courses( name: "DIGI-WINNER", financing: "Financing: 40-80%",  
        specialization: "financed expenditures for IT courses and training.", maxSum: 5000);  
    private Courses c2=new Courses( name: "DIGI-WINNER", financing: "Financing: 40-80%",  
        specialization: "financed expenditures for IT courses and training.", maxSum: 2500);  
    private Courses c3=new Courses( name: "Frauen und Beruf (FRECH)", financing: "Financing: 90%",  
        specialization: "funded courses on change of profession.", maxSum: 5000);  
    private Courses c4=new Courses( name: "AK Bildungsgutschein", financing: "Voucher is valid for the calendar year in" +  
        " which they were issued.", specialization: "is suitable for all courses.", maxSum: 120);  
    private Courses c5=new Courses( name: "AK Digi-Bonus", financing: "Voucher can be used with \"AK Bildungsgutschein\".",  
        specialization: "is suitable only for IT courses.", maxSum: 120);  
  
    //Adding Course to Map-----  
    public void addCourse(){  
        courses.put(1,c1);  
        courses.put(2,c2);  
        courses.put(3,c3);  
        courses.put(4,c4);  
        courses.put(5,c5);  
    }  
}
```

Class Courses with structure of course is using for creating the **Map** of available courses.

waff | code Logic | Courses

```
//CourseGetter Logic Method-----  
public String getCourses () {  
    int i = 1;  
    if((login.getGender().equals("f"))&&(liveInVienna == true)&&(workInVienna == true)&& (akMemberVienna == true)  
    &&(salary1800==true)){  
        return i++ + ". " + c1.info() + "\n\n" +  
            i++ + ". " + c3.info() + "\n\n" +  
            i++ + ". " + c4.info() + "\n\n" +  
            i++ + ". " + c5.info();  
    }  
    else if((Login.getGender().equals("f"))&&(LiveInVienna == true)&&(workInVienna == true)&&  
        (akMemberVienna == false)&&(akMemberAT == true)&&(salary1800==true)){  
        return i++ + ". " + c1.info() + "\n\n" +  
            i++ + ". " + c3.info();  
    }  
    else if((LiveInVienna == true)&&(workInVienna == true)&&(akMemberVienna == true)  
        & ((salary1800==true)|| (salaryLess2500==true))){  
        return i++ + ". " + c1.info() + "\n\n" +  
            i++ + ". " + c4.info() + "\n\n" +  
            i++ + ". " + c5.info();  
    }  
    else if((liveInVienna == false)&&(workInVienna == true)&&(akMemberVienna == true)  
        && ((salary1800==true)|| (salaryLess2500==true))){  
        return i++ + ". " + c2.info() + "\n\n" +  
            i++ + ". " + c4.info() + "\n\n" +  
            i++ + ". " + c5.info();  
    }  
    else if((liveInVienna == true)&&(workInVienna == false)&&(akMemberVienna == true)  
        && ((salary1800==true)|| (salaryLess2500==true))){  
        return i++ + ". " + c2.info() + "\n\n" +  
            i++ + ". " + c4.info() + "\n\n" +  
            i++ + ". " + c5.info();  
    }  
}
```



waff | code Logic | Conditions

```
//Creating a conditionsList Methods-----  
public void addCondition(String pCondition){ conditions.add(pCondition); }  
  
public void deleteCondition(String pCondition){ conditions.remove(pCondition); }
```

```
//ConditionsGetter Method-----  
public String getConditions() {  
    String conditionsList = "";  
    for (String list : conditions) {  
        conditionsList += list;  
    }  
    return conditionsList;  
}
```

InfoAustria | waff - 23.03.2021

Do you live in Vienna?	<input checked="" type="radio"/> Yes	<input type="radio"/> No
Are you currently employed in Vienna?	<input checked="" type="radio"/> Yes	<input type="radio"/> No
Are you AK Member?	<input checked="" type="radio"/> Yes, in Vienna	<input type="radio"/> Yes, in other state
Your salary per month:	<input checked="" type="radio"/> <1800 net	<input type="radio"/> <2500 net
	<input type="radio"/> >2500 net	

[Back to Main Menu](#) [Continue](#)

Possibility to **add** and **delete** conditions, which influence the choice of the course.

WAFF provide people in Vienna with information, advice and financial support for their professional development. In addition, they work in close cooperation with the Vienna Public Employment Service (AMS Vienna) to serve as a port of call for job-searcher and companies.

You live in Vienna, You work in Vienna, You are a member of AK Vienna, Your salary is less then 1800 Eur net.

waff | Controller

```
//additionalProperties  
private boolean liveInVienna, workInVienna;  
private boolean akMemberVienna, akMemberAT;  
private boolean salary1800, salaryLess2500, salaryMore2500;  
  
//Handing over Properties-----  
public Waff(boolean liveInVienna, boolean workInVienna, boolean akMemberVienna,  
            boolean salary1800, boolean salaryLess2500, boolean salaryMore2500){  
    this.liveInVienna = liveInVienna;  
    this.workInVienna = workInVienna;  
    this.akMemberVienna = akMemberVienna;  
    this.akMemberAT = akMemberAT;  
    this.salary1800 = salary1800;  
    this.salaryLess2500 = salaryLess2500;  
    this.salaryMore2500 = salaryMore2500;  
}
```

Using **additional properties** program can get the courses logic.

```
//ActionPerformed Method-----  
@Override  
public void actionPerformed(ActionEvent e) {  
    //questions Controller-----  
    if (e.getSource() == yesLive) {  
        liveInViennaMap = true;  
        waff.addCondition( pCondition: "You live in Vienna, ");  
        haveList.add("You live in Vienna");  
    }  
    else if (e.getSource() == noLive) {  
        liveInViennaMap = false;  
        waff.deleteCondition( pCondition: "You live in Vienna, ");  
        try {  
            haveList.remove( item: "You live in Vienna");  
        }catch (Exception r){  
            System.out.println(r);  
        }  
    }  
    else if (e.getSource() == yesWork) {  
        workInViennaMap = true;  
        waff.addCondition( pCondition: "You work in Vienna, ");  
        haveList.add("You work in Vienna");  
    }  
    else if (e.getSource() == noWork) {  
        workInViennaMap = false;  
        waff.deleteCondition( pCondition: "You work in Vienna, ");  
        try {  
            haveList.remove( item: "You work in Vienna");  
        }catch (Exception r){  
            System.out.println(r);  
        }  
    }  
}
```

With **try/catch** (Exception) program catch bugs in **Runtime**.

