Laboratory 8: Fun with Calculators – Week 3

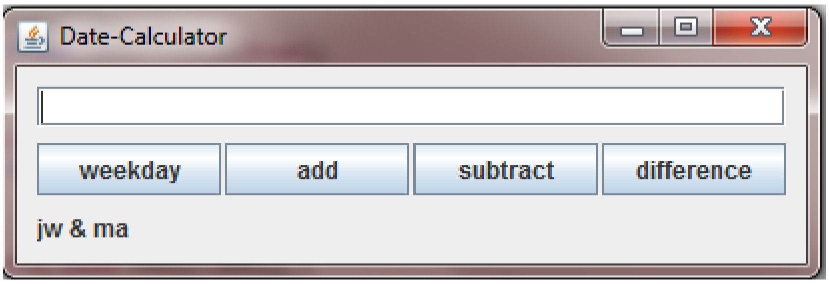
(http://people.f4.htw-berlin.de/~weberwu/info2/labs/Exer8.shtml)

1. Make another new copy of the Calculator (don't wreck your previous versions) before you start.

We took the calculator from the exercises before and removed every unnecessary functions and classes. For example the whole CalcEngine class is gone, since we are only going to work with Strings from the text field. We also removed all the buttons and added 4 new ones for the upcoming JDate tasks. And implemented the project from exercise 4 in order to use their useful methods on Julian dates.

Those are the buttons we implemented:

1. *Weekday-Button*
2. *Add-Button*
3. *Substract-Button*
4. *Difference-Button*



2. If you are doing the date calculator, implement the following functions:

1. input a date

We decided to simply write things with the hardware keyboard to the textfield of the user interface, to input dates and as soon as we press any of the buttons above the input gets evaluated. So here was not much to do.

2. get the calculator to display a String for the weekday in the window (hint: you will need a button to push)

At first we created a new Java class called “JDateFunctions”, where we put any of the methods we need to code, to make all the magic happen.

In the constructor of this class we simply initialize an instance of JDate (exercise 4) to be able to use its methods.

```
public JDateFunctions() {
        c = Calendar.getInstance();
        jd = new JDate(0);
}
```

Now we create a method

```
public String showWeekday(String s)
```

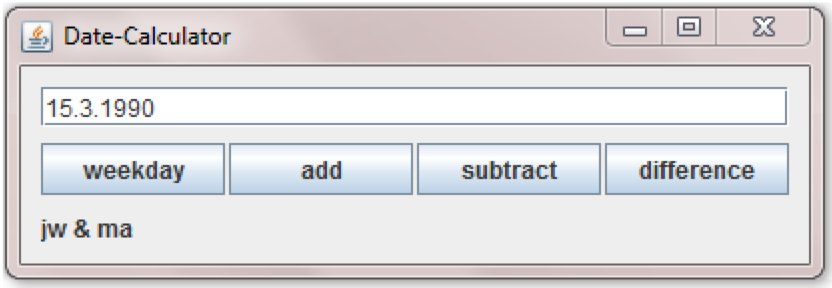
which takes a String and returns a String.

Now we get the String “15.3.1990” through the parameters. We split the String at its dots, to access the days, months and years of the date and then simply pass this data to the `getWeekday` method of the JDate class from exercise 4 and return the result.

```
  public String showWeekday(String s) {
          String[] a = s.split("\\.");
          try {
                  //Lese Datum
                  int day = Integer.parseInt(a[0]);
                  int month = Integer.parseInt(a[1]);
                  int year = Integer.parseInt(a[2]);

                  return jd.getWeekday(jd.convertDateToJulianDate(day, month,
                  year));
          }catch(Exception e) {
                  return "Error: Wrong formatted input. Please clear!";
          }
}
```

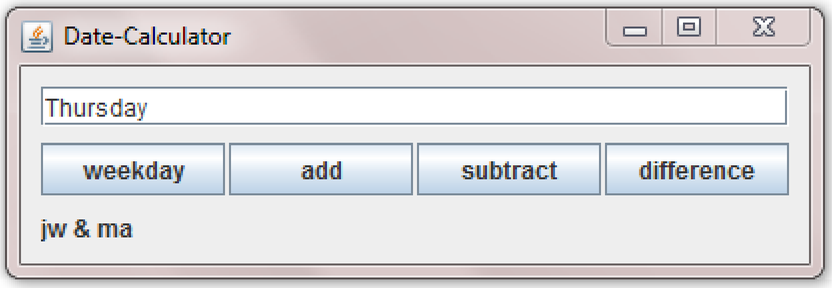
The method itself gets called in the actionPerformed() method oft he UserInterface class, when the *Weekday-Button* is pressed.

```
if (command.equals("weekday")) {
                displayContent = func.showWeekday(display.getText());
```



15.3.1990

Evaluates to



Thursday

When *Weekday-Button* is pressed.

3. add a number of days to a date, displaying the new date

A new method *addDays()* is created.

```
public String addDays(String s)
```

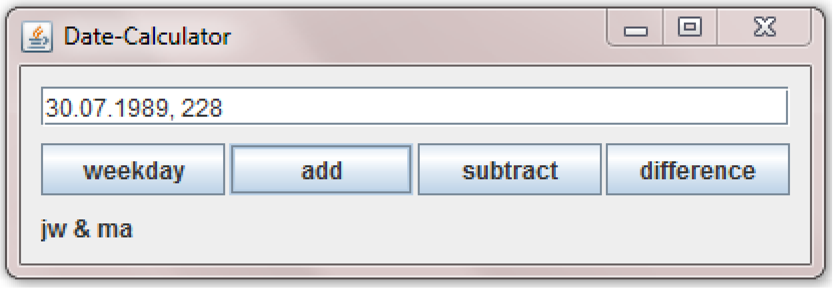
Here the inputted String may look like “30.7.1989, 228” which means we want to add 228 days to the given date.

At first we have to remove possible spaces, split the String at the comma, and treat the first String just as we did before, to access its day, month and year value. The second String is simply parsed into a Integer value and all the data is added and return with the help of the methods from JDate.

This method also gets called in the actionPerformed() method of the UserInterface, if the *subtract-Button* is pressed.

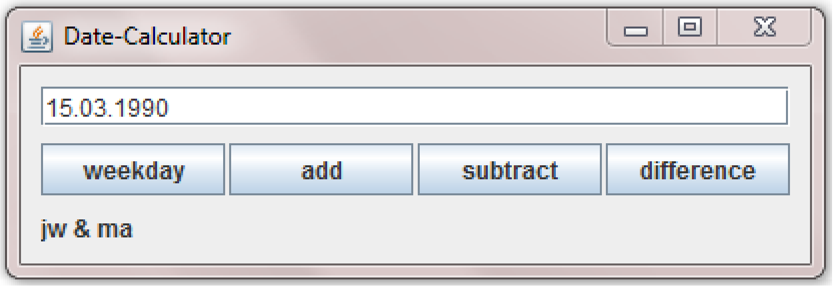
```
public String addDays(String s){
//Entferne evtl. vorhandene Leerzeichen aus dem String
s = s.replaceAll(" ", "");
//Teile String bei "-" in 2 Strings (Datum & Anzahl Tage)
String[] comma = s.split(",");
//Teile ersten String bei "." in 3 Strings (Tage, Monat, Jahr)
String[] date = comma[0].split("\\.");
int jDate = 0, numDays = 0;

try{
```

```
            //Lese Datum
            int day = Integer.parseInt(date[0]);
            int month = Integer.parseInt(date[1]);
            int year = Integer.parseInt(date[2]);
            jDate = jd.convertDateToJulianDate(day, month, year);

            //Lese Anzahl Tage
            numDays = Integer.parseInt(comma[1]);
            return jd.convertJulianDateToDate(jDate+numDays);
      }catch(Exception e){
            return "Error: Wrong formatted input. Please clear!";
      }
    }
```



30.07.1989, 228

Evaluates to



15.03.1990

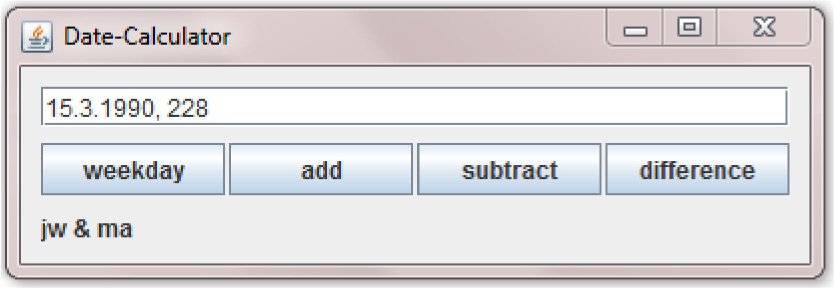
When *add-Button* is pressed.

4. subtract a number of days from a date, displaying the new date

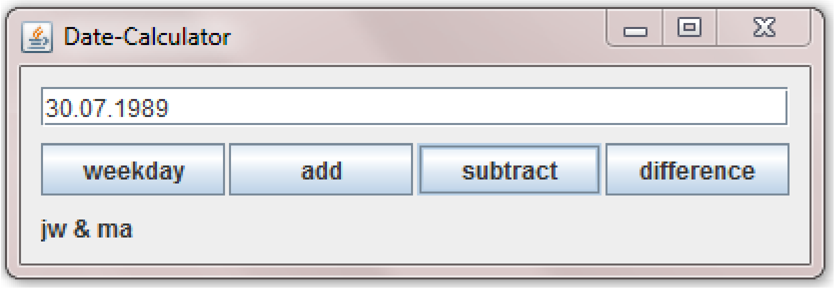
Here basically it's the same thing as before, but subtracting instead of adding.

```
public String subtractDays(String s) {
//Entferne evtl. vorhandene Leerzeichen aus dem String
s = s.replaceAll(" ", "");
//Teile String bei "-" in 2 Strings (Datum & Anzahl Tage)
String[] comma = s.split(",");
//Teile ersten String bei "." in 3 Strings (Tage, Monat, Jahr)
String[] date = comma[0].split("\\.");
int jDate = 0, numDays = 0;
```

```
        try{
            //Lese Datum
            int day = Integer.parseInt(date[0]);
            int month = Integer.parseInt(date[1]);
            int year = Integer.parseInt(date[2]);
            jDate = jd.convertDateToJulianDate(day, month, year);

            //Lese Anzahl Tage
            numDays = Integer.parseInt(comma[1]);
            return jd.convertJulianDateToDate(jDate-numDays);
        }catch(Exception e){
            return "Error: Wrong formatted input. Please clear!";
        }
    }
```



Evaluates to



When *subtract-Button* is pressed.

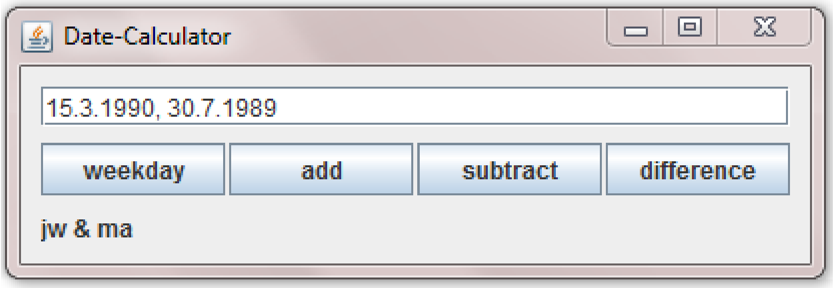
5. subtract two dates, giving the number of days between the dates

For the method *differenceDates()* to give the number of days between two dates, when the *difference-Button* is pressed the input String may look like “15.3.1990, 30.7.1989”.

So we again remove all the spaces, split the String at the comma and now split both Strings at its dots again, so we can access the day, month and year value of both dates.

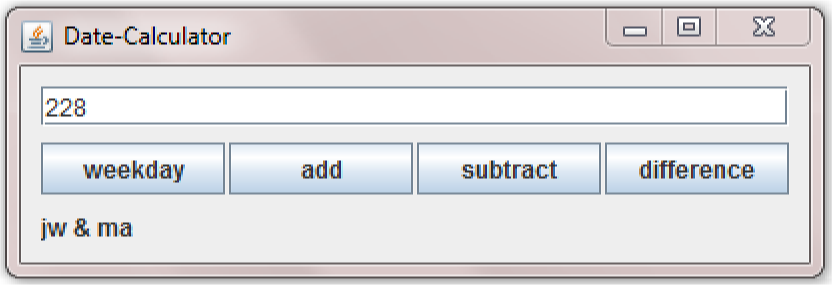
Those are passed to the *getDateDifference()* method of JDate and the result is getting returned.

```
        public String differenceDates(String s) {
        //Entferne evtl. vorhandene Leerzeichen aus dem String
        s = s.replaceAll(" ", "");
        //Teile String bei "-" in 2 Strings (Datum & Datum 2)
        String[] comma = s.split(",");
        //Teile beide Strings bei "."
        String[] date1 = comma[0].split("\\.");
        String[] date2 = comma[1].split("\\.");

        try {
                //Erstes Datum
                int day1 = Integer.parseInt(date1[0]);
                int month1 = Integer.parseInt(date1[1]);
                int year1 = Integer.parseInt(date1[2]);
                //Zweites Datum
                int day2 = Integer.parseInt(date2[0]);
                int month2 = Integer.parseInt(date2[1]);
                int year2 = Integer.parseInt(date2[2]);

                return "" + jd.getDateDifference(jd.convertDateToJulianDate(day1,
month1, year1), jd.convertDateToJulianDate(day2, month2, year2));
        }catch(Exception e) {
                return "Error: Wrong formatted input. Please clear!";
        }
    }
```



15.3.1990, 30.7.1989

Evaluates to



228

When *difference-Button* is pressed.

Source code:

Calculator class:

```
/**
 * The main class of a simple calculator. Create one of these and you'll
 * get the calculator on screen.
 *
 * @author David J. Barnes and Michael Kolling (edited by ma & jw)
 * @version 2012.11.13
 */
public class Calculator
{
    private UserInterface gui;

    /**
     * Create a new calculator and show it.
     */
    public Calculator() {
        gui = new UserInterface();
    }

    public static void main(String args[]) {
       new Calculator();
    }

    /**
     * In case the window was closed, show it again.
     */
    public void show() {
        gui.setVisible(true);
    }
}
```

UserInteface class:

```
public class UserInterface implements ActionListener {
      private String displayContent = "";
      private JDateFunctions func;

      private JFrame frame;
      private JTextField display;
      private JLabel status;

      public UserInterface() {
            func = new JDateFunctions();
            makeFrame();
            frame.setVisible(true);
      }

      public void setVisible(boolean visible) {
            frame.setVisible(visible);
      }

      private void makeFrame() {
            frame = new JFrame("Date-Calculator");
```

```
        JPanel contentPane = (JPanel) frame.getContentPane();
        contentPane.setLayout(new BorderLayout(8, 8));
        contentPane.setBorder(new EmptyBorder(10, 10, 10, 10));

        display = new JTextField();
        contentPane.add(display, BorderLayout.NORTH);

        JPanel buttonPanel = new JPanel(new GridLayout(0, 4, 2, 2));
        addButton(buttonPanel, "weekday");
        addButton(buttonPanel, "add");
        addButton(buttonPanel, "subtract");
        addButton(buttonPanel, "difference");

        contentPane.add(buttonPanel, BorderLayout.CENTER);

        status = new JLabel("jw & ma");
        contentPane.add(status, BorderLayout.SOUTH);

        frame.pack();
    }

    private void addButton(Container panel, String buttonText) {
        JButton button = new JButton(buttonText);
        button.addActionListener(this);
        panel.add(button);
    }

    public void actionPerformed(ActionEvent event) {
        String command = event.getActionCommand();

        if (command.equals("weekday")) {
            displayContent = func.showWeekday(display.getText());
        } else if (command.equals("add")) {
            displayContent = func.addDays(display.getText());
        } else if (command.equals("subtract")) {
            displayContent = func.subtractDays(display.getText());
        } else if (command.equals("difference")) {
            displayContent = func.differenceDates(display.getText());
        }

        redisplay();
    }

    private void redisplay() {
        display.setText(displayContent);
    }
}
```

JDateFunctions class:

```
import java.util.Calendar;


public class JDateFunctions {
	private JDate jd;

	public JDateFunctions() {
		jd = new JDate(0);
	}

    public String showWeekday(String s) {
	String[] a = s.split("\\.");
	try {
		//Lese Datum
		int day = Integer.parseInt(a[0]);
		int month = Integer.parseInt(a[1]);
		int year = Integer.parseInt(a[2]);

		return jd.getWeekday(jd.convertDateToJulianDate(day, month, year));
	}catch(Exception e) {
		return "Error: Wrong formatted input. Please clear!";
	}
    }

    public String addDays(String s){
	//Entferne evtl. vorhandene Leerzeichen aus dem String
	s = s.replaceAll(" ", "");
	//Teile String bei "-" in 2 Strings (Datum & Anzahl Tage)
	String[] comma = s.split(",");
	//Teile ersten String bei "." in 3 Strings (Tage, Monat, Jahr)
	String[] date = comma[0].split("\\.");
	int jDate = 0, numDays = 0;

	try{
		//Lese Datum
		int day = Integer.parseInt(date[0]);
		int month = Integer.parseInt(date[1]);
		int year = Integer.parseInt(date[2]);
		jDate = jd.convertDateToJulianDate(day, month, year);

		//Lese Anzahl Tage
		numDays = Integer.parseInt(comma[1]);
		return jd.convertJulianDateToDate(jDate+numDays);
	}catch(Exception e){
		return "Error: Wrong formatted input. Please clear!";
	}
    }

    public String subtractDays(String s) {
	//Entferne evtl. vorhandene Leerzeichen aus dem String
	s = s.replaceAll(" ", "");
	//Teile String bei "-" in 2 Strings (Datum & Anzahl Tage)
	String[] comma = s.split(",");
	//Teile ersten String bei "." in 3 Strings (Tage, Monat, Jahr)
	String[] date = comma[0].split("\\.");
	int jDate = 0, numDays = 0;

	try{
		//Lese Datum
		int day = Integer.parseInt(date[0]);
		int month = Integer.parseInt(date[1]);
```

```
             int year = Integer.parseInt(date[2]);
             jDate = jd.convertDateToJulianDate(day, month, year);

             //Lese Anzahl Tage
             numDays = Integer.parseInt(comma[1]);
             return jd.convertJulianDateToDate(jDate-numDays);
      }catch(Exception e){
             return "Error: Wrong formatted input. Please clear!";
      }
    }

    public String differenceDates(String s) {
      //Entferne evtl. vorhandene Leerzeichen aus dem String
      s = s.replaceAll(" ", "");
      //Teile String bei "-" in 2 Strings (Datum & Datum 2)
      String[] comma = s.split(",");
      //Teile beide Strings bei "."
      String[] date1 = comma[0].split("\\.");
      String[] date2 = comma[1].split("\\.");

      try {
             //Erstes Datum
             int day1 = Integer.parseInt(date1[0]);
             int month1 = Integer.parseInt(date1[1]);
             int year1 = Integer.parseInt(date1[2]);
             //Zweites Datum
             int day2 = Integer.parseInt(date2[0]);
             int month2 = Integer.parseInt(date2[1]);
             int year2 = Integer.parseInt(date2[2]);

             return "" + jd.getDateDifference(jd.convertDateToJulianDate(day1,
month1, year1), jd.convertDateToJulianDate(day2, month2, year2));
      }catch(Exception e) {
             return "Error: Wrong formatted input. Please clear!";
      }
    }
}
```