TECHNISCHE HOCHSCHULE LÜBECK

SOFTWARE ENGINEERING II

11.04.2019

USABILITY ENGINEERING

LENKA KLEINAU, M.SC.

SLIDES BASED ON MATERIAL FROM MRS. BAHTA

CONTENT

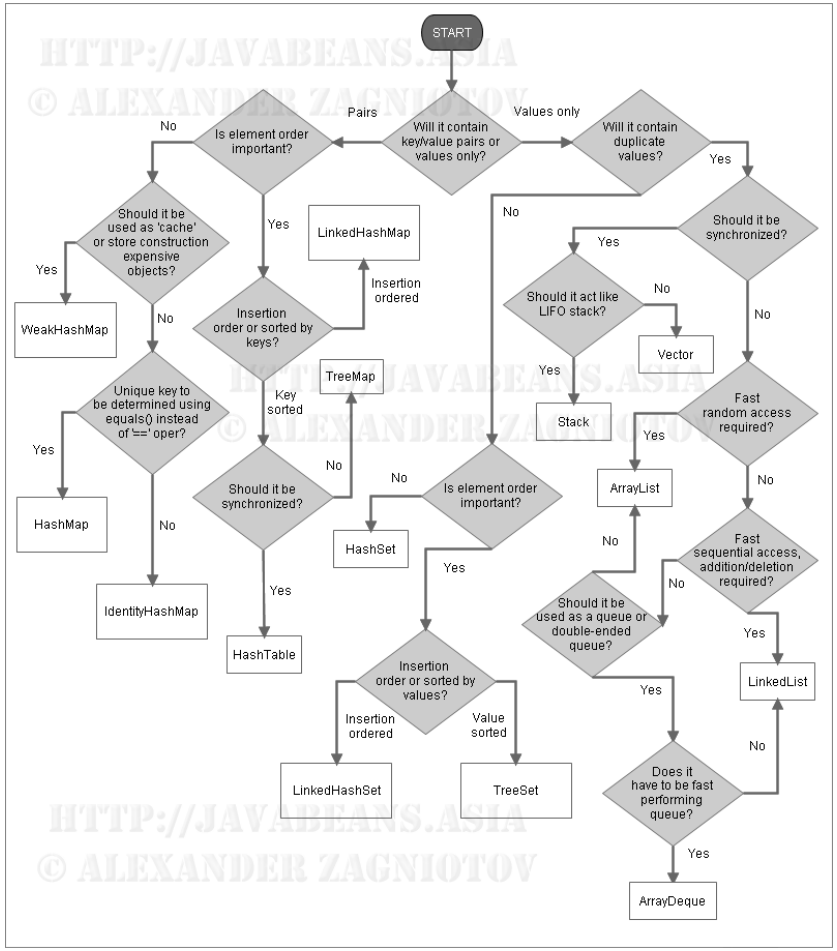
TECHNISCHE HOCHSCHULE LÜBECK

- Repetition
- Feedback for exercises
- Usability Engineering
- User Interfaces
- Java Swing/AWT/FX

- Project organization
- Exercise/project consultations

REPETITION

- Java code conventions
- Why important?



https://kaanmutlu.files.wordpress.com/2011/12/collections.png

PERSON CLASS ASSIGNMENT

TECHNISCHE HOCHSCHULE LÜBECK

- Encapsulation → private!
- System.out.println(this) ↔System.out.println(this.toString())
- Some of you forgot some parts of task

- **Solutions for destructor?** (finalize, null, System.gc)

→ http://docstore.mik.ua/orelly/java-ent/jnut/ch03_03.htm

TECHNISCHE HOCHSCHULE LÜBECK

- How usable is something?
- **Quality attribute** that assesses how easy user interfaces are to use
- **Methods for improving ease-of-use** during the design process

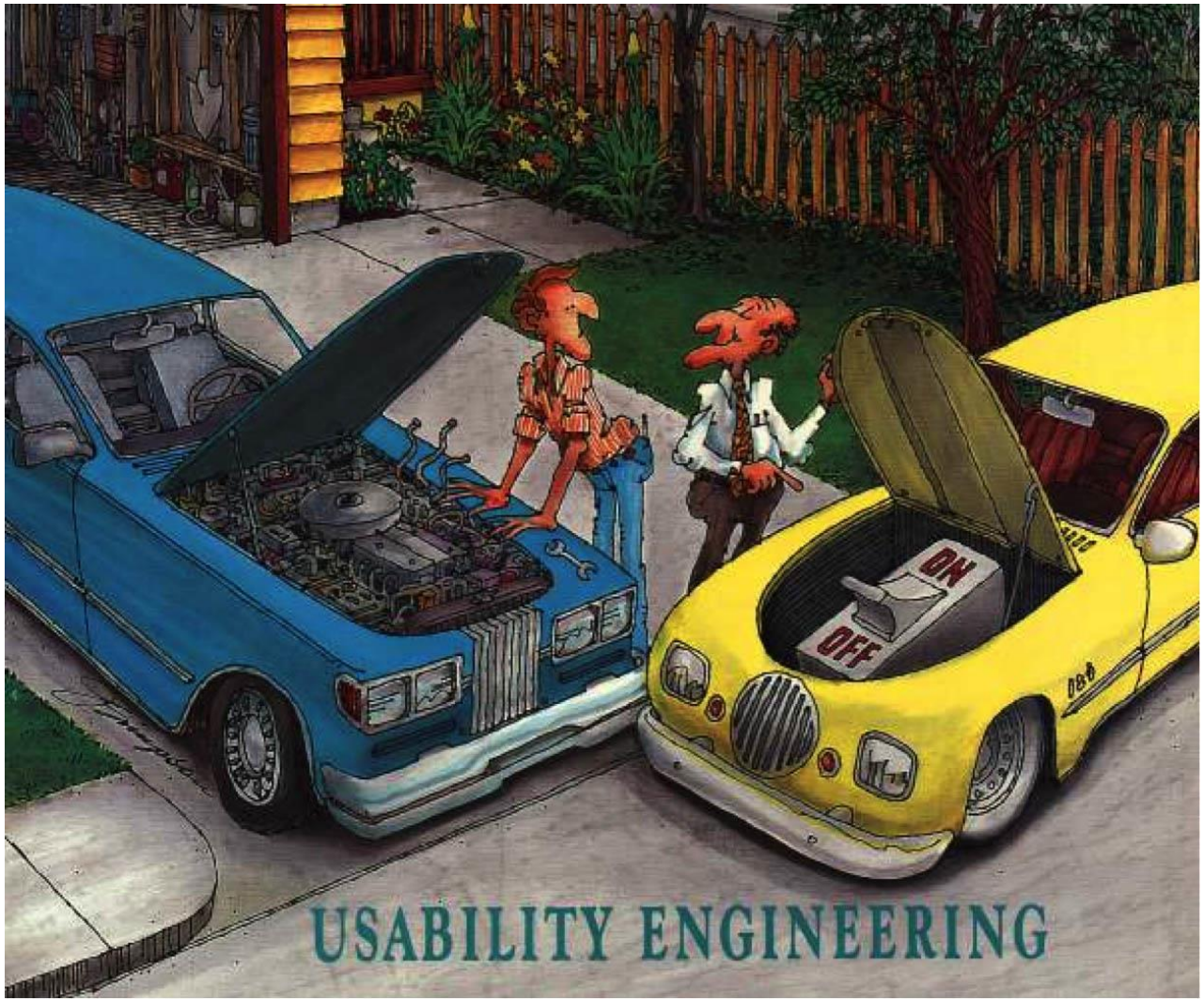
CHANGE OF EASE-OF-USE

TECHNISCHE HOCHSCHULE LÜBECK

Past	Presence
High acquisition costs for computer	Low acquisition costs for computer
Only few computer users („experts“)	Everyone uses computers
Very specialized systems	Systems for everything
Lack of resources (e.g. memory)	Few constraints on resources
No usability was the rule	Users are used to high usability
Usability was secondary	Usability most important for sales/marketing

USABILITY ENGINEERING – WHY?

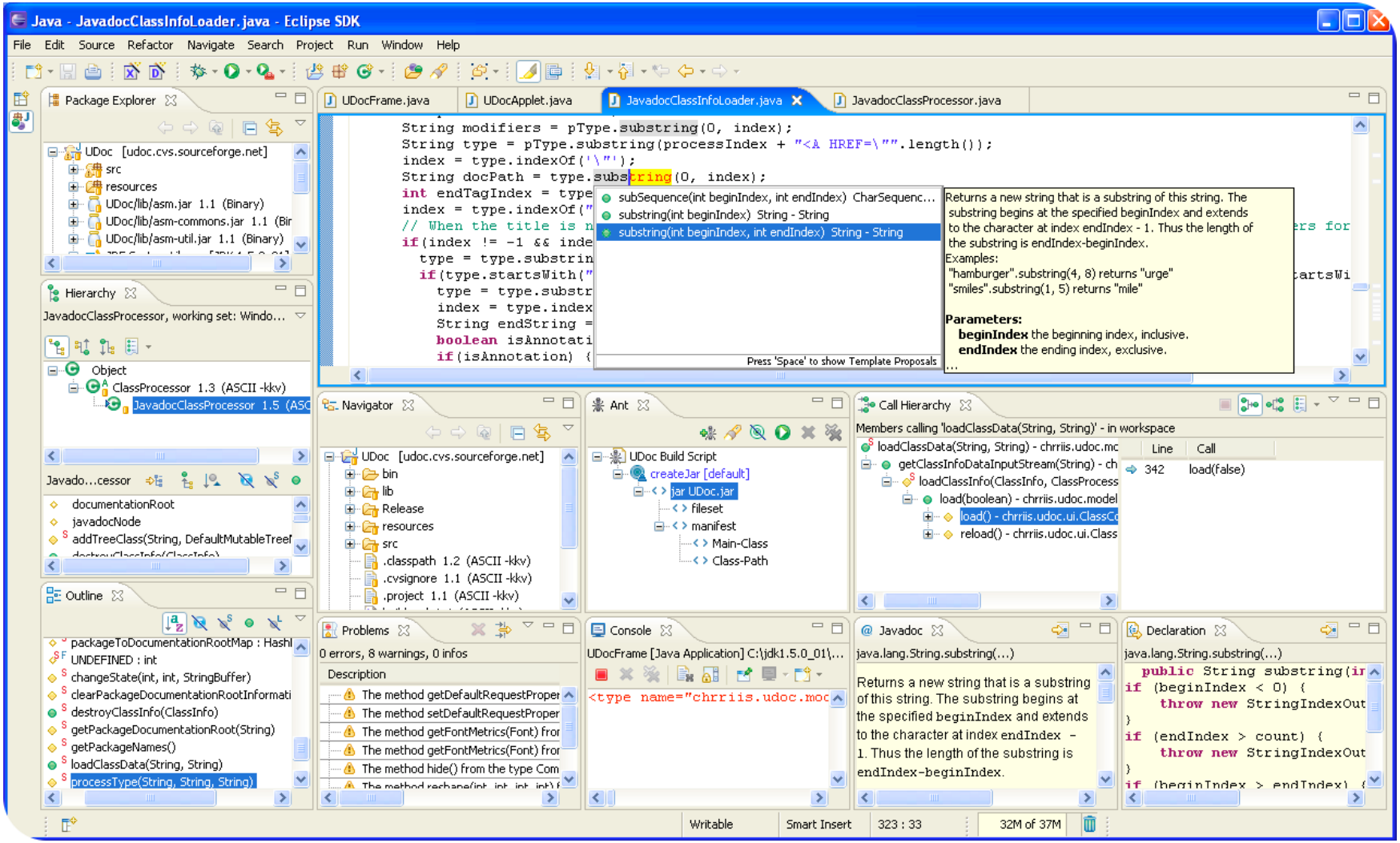
TECHNISCHE HOCHSCHULE LÜBECK



http://usability-tips.com/usability-post-1/

USABILITY ENGINEERING – WHY?

TECHNISCHE HOCHSCHULE LÜBECK



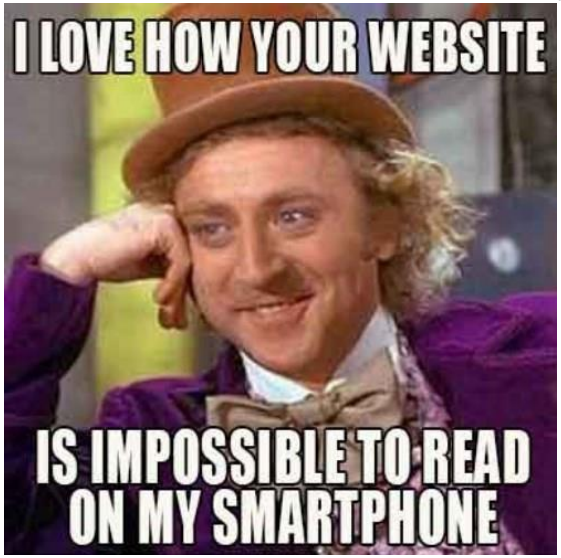
http://asserttrue.blogspot.de/2012/12/more-thoughts-on-feature-richness.html

USABILITY ENGINEERING – WHY?

TECHNISCHE HOCHSCHULE LÜBECK

http://www.opefficiently-testing-and-reporting-web-site-usability.html
timum7.com/internet-marketing/usability-testing/

http://halls-of-valhalla.org/beta/articles/modern-trends-in-ux-that-ruin-usability,50/



https://www.perth-web-design.com.au/
5-steps-to-optimize-your-websites-user-experience-ux/

TECHNISCHE HOCHSCHULE LÜBECK

- **The user..**
  - ..is not like you!
  - ..thinks and works differently from you!
  - ..knows and expects other things than you!

→ Need of incorporating user‘s view into the developing process!

TECHNISCHE HOCHSCHULE LÜBECK

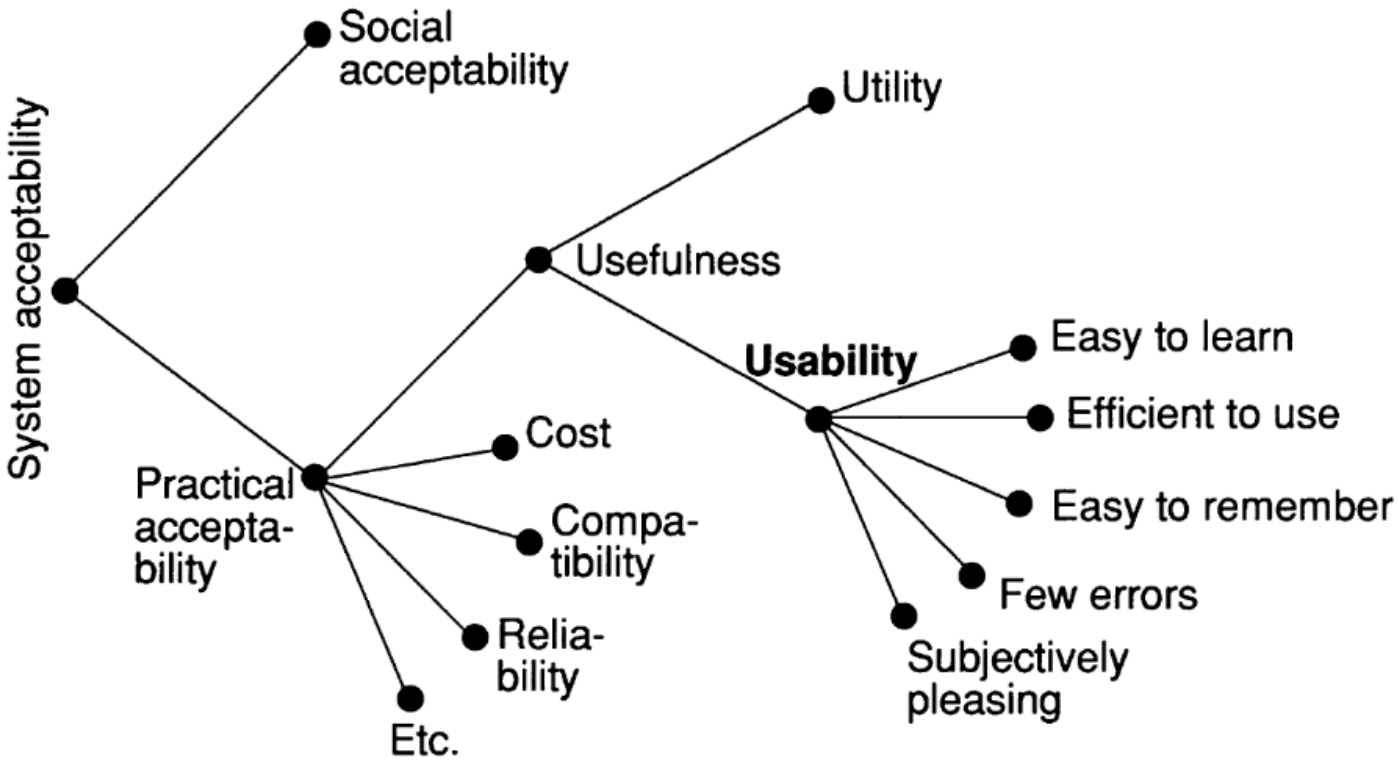
- **Learnability**
  - How easy is it for users to accomplish basic tasks the first time they encounter the design?
- **Efficiency**
  - Once users have learned the design, how quickly can they perform tasks?
- **Memorability**
  - When users return to the design after a period of not using it, how easily can they reestablish proficiency?

TECHNISCHE HOCHSCHULE LÜBECK

- **Errors**
  - How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- **Satisfaction**
  - How pleasant is it to use the design?

TECHNISCHE HOCHSCHULE LÜBECK

- **Acceptance**
  - System is good enough to satisfy all needs and requirements of users



TECHNISCHE HOCHSCHULE LÜBECK

- **Cost savings**
  - Example: improvement of Boeing 757 flight deck interface:
    - Operation of two instead of three pilots
    - 35% time decrease in production line for integrated circuits
    - Down from 3000 to 150 words of instructions needed for operating the paging device
  - Once a system is in development, correcting a problem costs 10 times more than fixing the same problem in design. If the system has been released, it costs 100 times more relative to fixing in design.

http://info.humanfactors.com

TECHNISCHE HOCHSCHULE LÜBECK

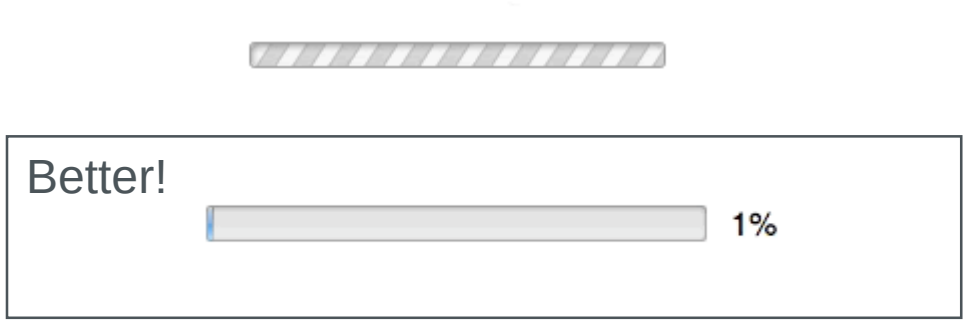
- **Cost savings**
  - By helping employees work faster, they become more productive. The financial benefit of this should not be underestimated. For example, by reducing the time spent choosing the right option from your company's Intranet home page by just 1 minute per day, a typical company with 5,000 employees could achieve £0.25m year in efficiency savings. (Calculated assuming a weighted hourly salary cost of £15 and assuming that the average employee works 200 days per year).
  - https://www.userfocus.co.uk/articles/usabilitybenefits.html

TECHNISCHE HOCHSCHULE LÜBECK

- **Development should depend on**
  - User (motivation, age, experience)
  - Goals
  - Work tasks
  - Work resources (HW, SW, ..)
  - Physical and social environment

TECHNISCHE HOCHSCHULE LÜBECK

- **Visibility of system status**
  - The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.



http://www.barchart.com/shared/images/progress_bar.gif
http://blog.teamtreehouse.com/wp-content/uploads/2014/04/loading.gif

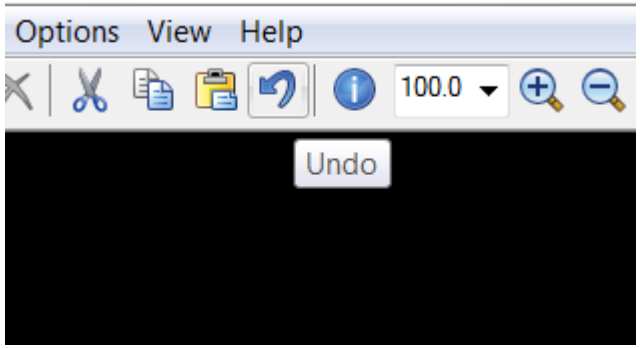
TECHNISCHE HOCHSCHULE LÜBECK

- **Match between system and the real world**
  - The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

http://bookstore.santarosa.edu/StoreImages/47-shopping-cart.png

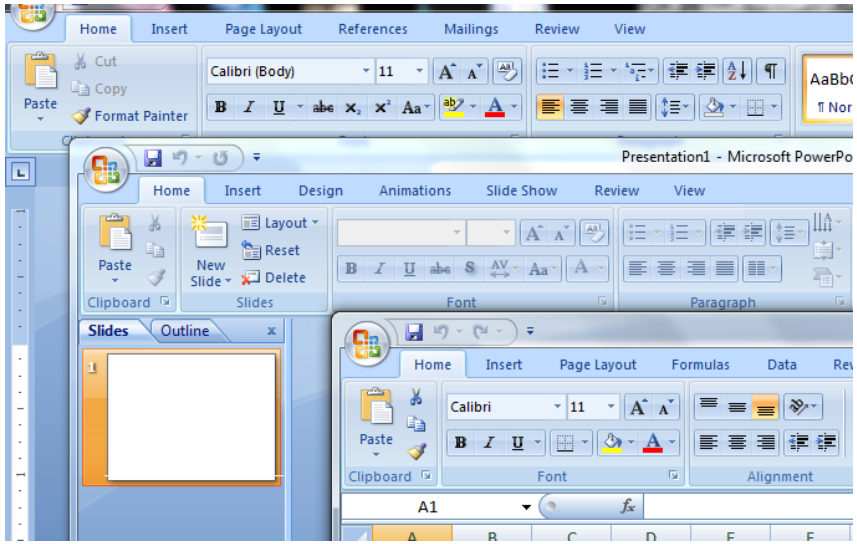
TECHNISCHE HOCHSCHULE LÜBECK

- **User control and freedom**
  - Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.



TECHNISCHE HOCHSCHULE LÜBECK

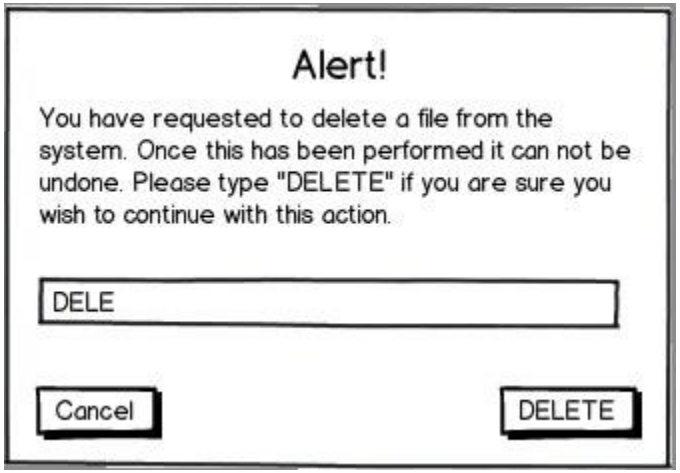
- **Consistency and standards**
  - Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.



https://summerproductions.files.wordpress.com/2012/08/5.png

TECHNISCHE HOCHSCHULE LÜBECK

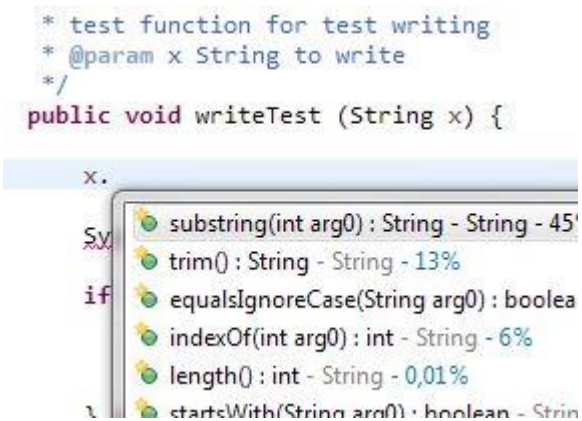
- **Error prevention**
  - Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.



http://www.labor.ny.gov/ux/guide-assets/images/prevent-errors-ff.jpg

TECHNISCHE HOCHSCHULE LÜBECK

- **Recognition rather than recall**
  - Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

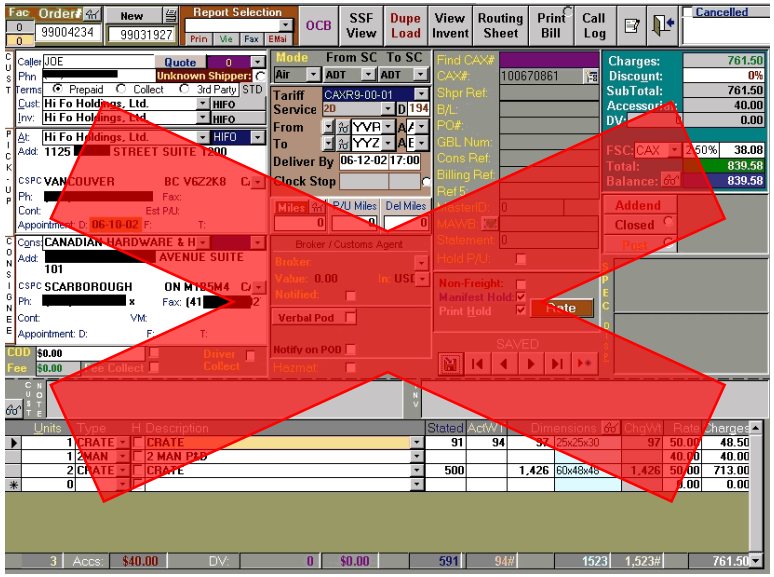


TECHNISCHE HOCHSCHULE LÜBECK

- **Flexibility and efficiency of use**
  - Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

TECHNISCHE HOCHSCHULE LÜBECK

- **Aesthetic and minimalist design**
  - Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.



http://www.usabilityinstitute.com/resources/images/oneclickinterface.gif

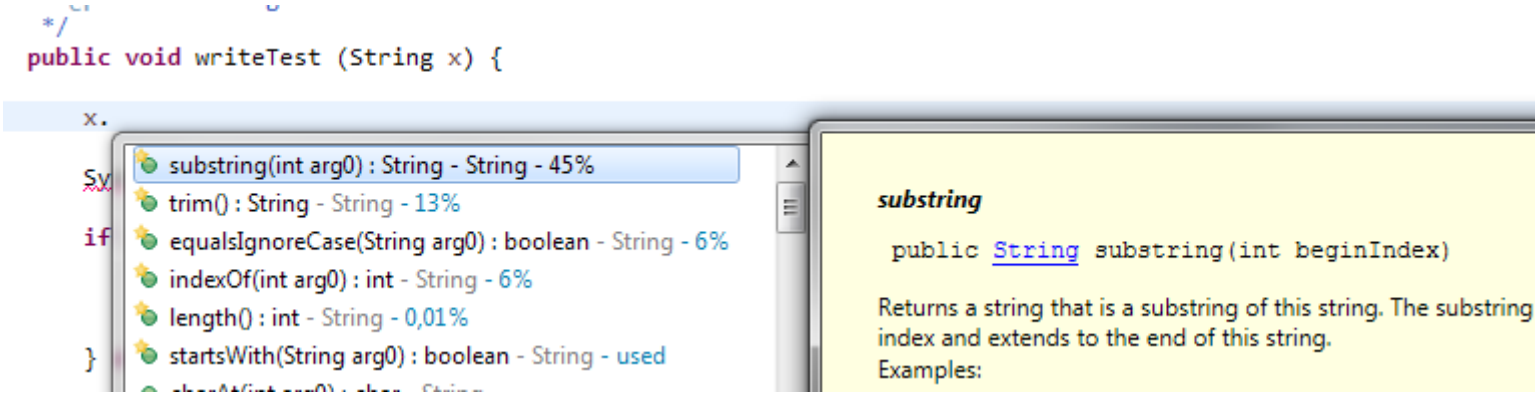
TECHNISCHE HOCHSCHULE LÜBECK

- **Help users recognize, diagnose, and recover from errors**
  - Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

http://s1.hubimg.com/u/4039752_f260.jpg

TECHNISCHE HOCHSCHULE LÜBECK

- **Help and documentation**
  - Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.



TECHNISCHE HOCHSCHULE LÜBECK

- **Issue: usability engineering/testing is often seen as too expensive and time consuming**

→ Discount usability engineering

- Simple methodology
- Better than ‘none’
- Four techniques:
  - User and task observation
  - Scenarios (paper-mockups, prototypes)
  - Simplified thinking aloud
  - Heuristic evaluation

Let‘s have a closer look!
Important for your upcoming project!

TECHNISCHE HOCHSCHULE LÜBECK

- Don’t let the customer or developer test the system!
- Get to know the user
  - Define user groups e.g. students, elderly, experts,..
- How many users do you want to test?
- Let the user think aloud

TECHNISCHE HOCHSCHULE LÜBECK

- **Example book purchasing website:**

- Define full task
  - Instead of giving the task to ‘log in’
  - Task of ‘buy a specified book from a website’

- Avoid giving hints
  - By giving the task ‘purchase book XY’ we can observe the user’s navigation on the website

TECHNISCHE HOCHSCHULE LÜBECK

- Time
  - Plan time amount for each user
  - Inform user about the duration of test
- Location
  - Make user as comfortable as possible
  - Create typical environment of task for testing

TECHNISCHE HOCHSCHULE LÜBECK

- Roles of testing team
  - Facilitator: main point of contact, conducting beginning/end interviews with user
  - Log keeper: record every action of user
  - Observer: back-up person for other members
- Recording
  - Pen and paper
  - Audio/video recording
  - Screen recording

TECHNISCHE HOCHSCHULE LÜBECK

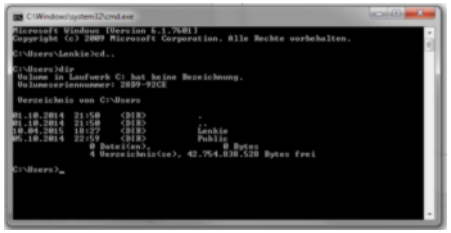
- **Usability Engineering**
  - User is in focus of development
  - Systematical improvement of usability and design
  - Can be incorporated in different development process stages
  - Provides different methodologies

- **Requirements Engineering**
  - Investigation of customer/user centered needs
  - Documentation using UML, processes, workflows, specifications

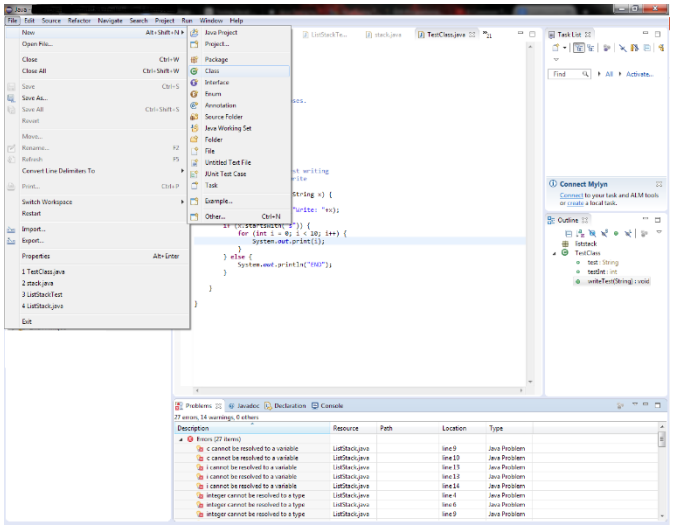
TECHNISCHE HOCHSCHULE LÜBECK

➢ Line-oriented interfaces

- Command line interface (CLI)



➢ Graphical user interfaces



➢ Modern user interfaces

- Voice
- Gestures
- Brain-computer interface

https://s3.amazonaws.com/ksr/assets/001/199/734/dc64fe6683000482122013 3e418ac957_large.jpg?1381960541

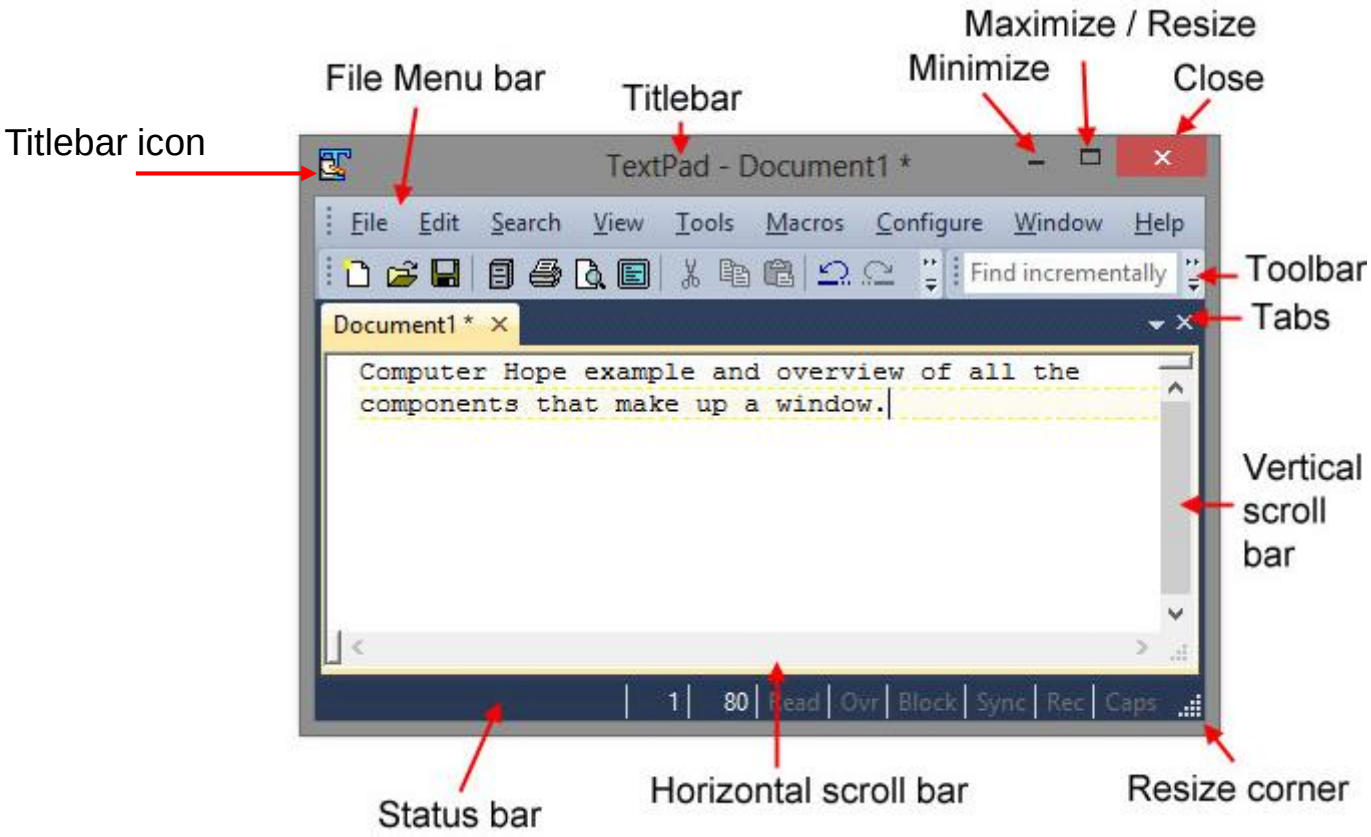
TECHNISCHE HOCHSCHULE LÜBECK

- **Dialog design**
  - *Primary dialog*
    - leads to task accomplishment
    - Closed if task is finished
  - *Secondary dialogs (e.g. printing dialog)*
    - Situation-dependent service dialog
    - Primary dialog continues after closing
  - *Modal dialog (must be finished until continuing task)*
  - *Modeless dialog (can be interrupted anytime)*

→ Goals: modeless dialogs ensure freedom of action for use
but sometimes restriction of action necessary

TECHNISCHE HOCHSCHULE LÜBECK

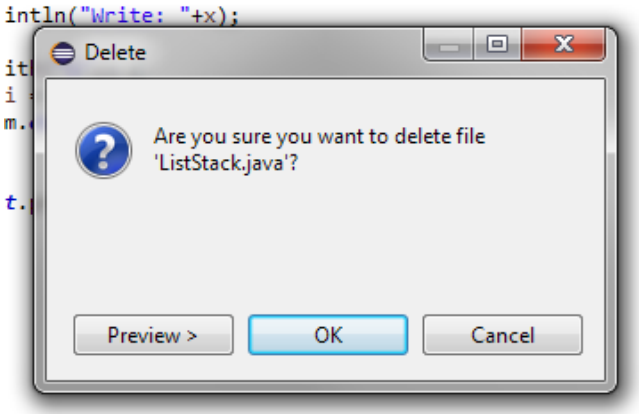
- **Window design**
  - Application Window (represents primary dialog)

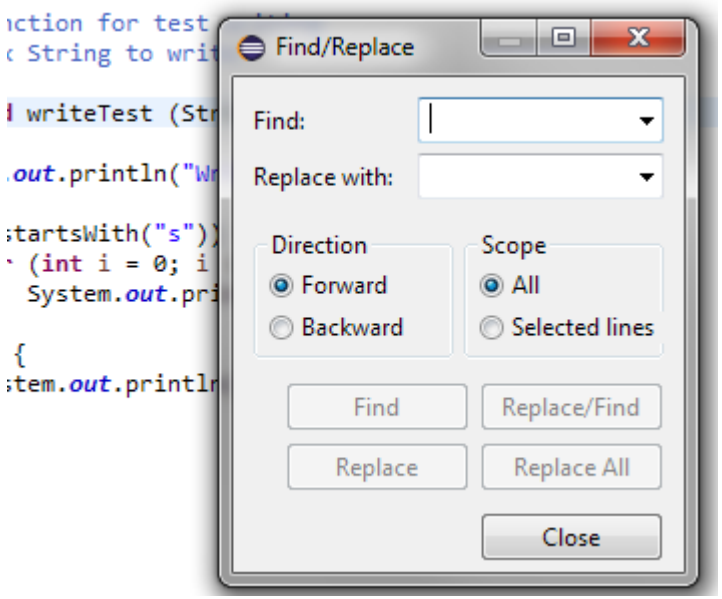


http://cdn.computerhope.com/window.jpg

TECHNISCHE HOCHSCHULE LÜBECK

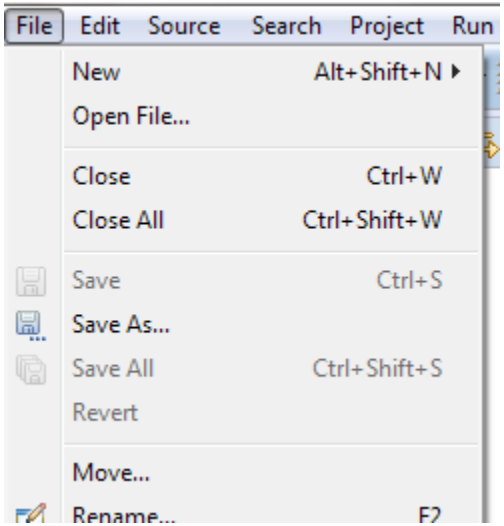
- Sub-window (opened from application window)
- Dialog window
  - Often designed for a secondary dialog
  - For data input
  - Often modal
- Alert/Popup window
  - Often modal

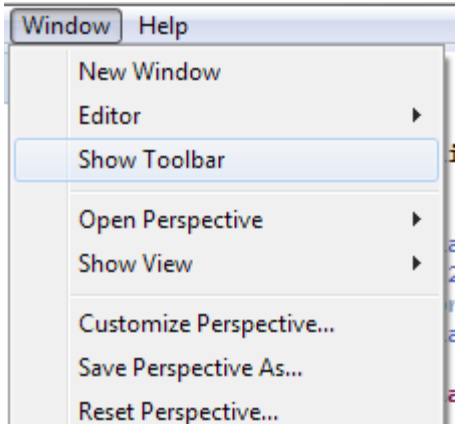




TECHNISCHE HOCHSCHULE LÜBECK

- **Menus**





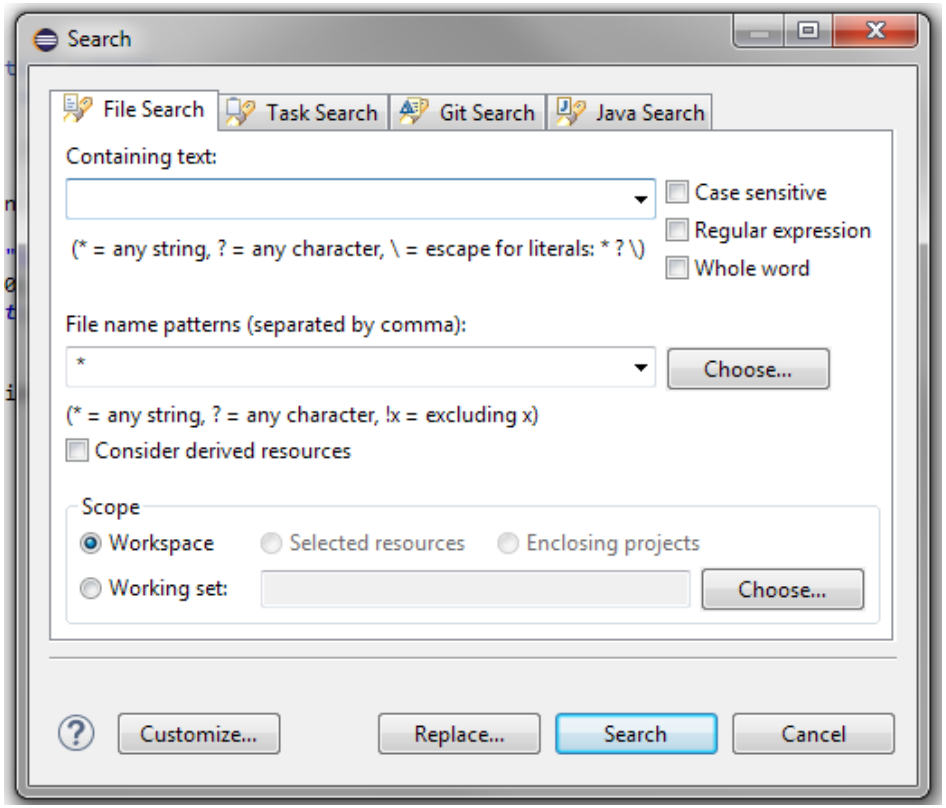
  - Pop up vs. pull/drop down menu

- **Sorted menu items assure good usability**
  - Time to find an item
  - Reasonable name for item

- **Menu items can be grouped by…**
  - Functionality
  - Alphabet
  - Natural order (from big to small)
  - The most used/importance

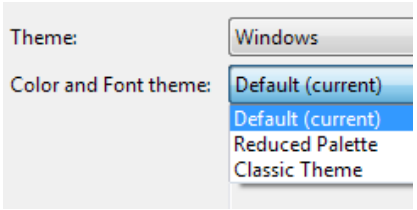
TECHNISCHE HOCHSCHULE LÜBECK

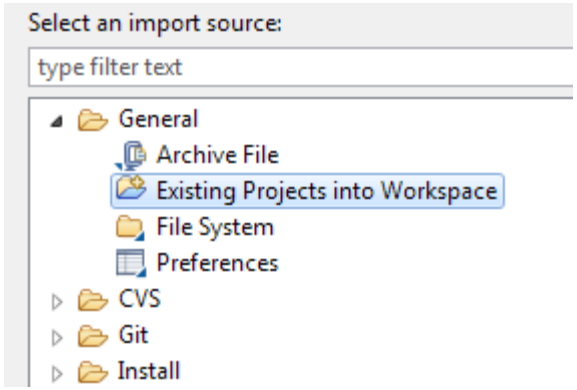
- **Graphical control elements**
  - Text field
  - Button
  - Radio button
  - Check box
  - Drop-down list box
  - Tabs
  - Icon

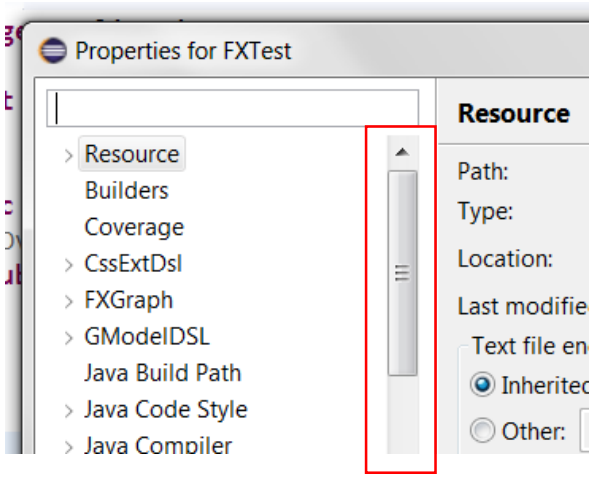


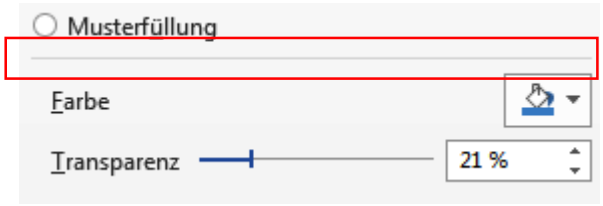
TECHNISCHE HOCHSCHULE LÜBECK

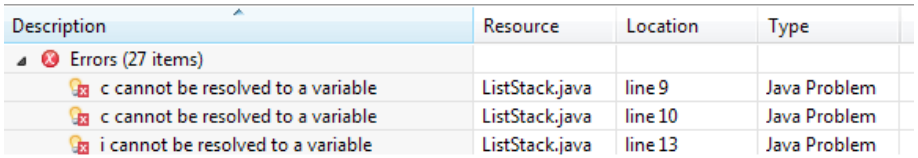
- **Graphical control elements**
  - Tree view
  - Combo box
  - List box
  - Scroll bar
  - Slider
  - Split bar
  - Table











TECHNISCHE HOCHSCHULE LÜBECK

- AWT (Abstract Window Toolkit)
  - Older GUI toolkit for Java
  - Heavyweight components

- Swing
  - Part of API for providing native look and feel GUI for Java
  - More components than AWT
  - Lightweight components

- JavaFX
  - Part of JDK (>1.7)
  - Intended to replace Swing but will still be supported
  - Supports touch device gestures
  - Supports various animations, CSS → Rich Internet Applications

TECHNISCHE HOCHSCHULE LÜBECK

- Documentation
  - https://docs.oracle.com/javase/8/docs/technotes/guides/swing/index.html

- Needed imports
  - import javax.swing.*;
  - import java.awt.*;

TECHNISCHE HOCHSCHULE LÜBECK

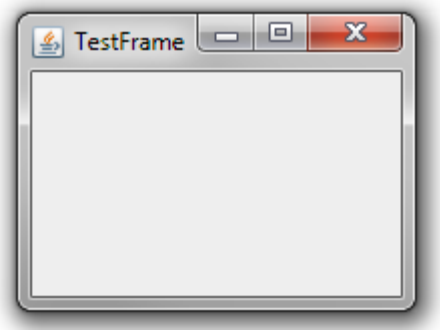
- **Components**
  - Windows and dialogs
  - Menus
  - Container
  - Control elements
  - Event handling
  - Layout manager

- **Windows and dialogs**

Class	Description
JFrame	Main window, base of most GUIs, includes menu bar
JDialog	Dialog window, is dependent on other windows, no menu bar
JColorChooser	Dialog for choosing colors from a palette
JFileChooser	Dialog for choosing files
JOptionPane	Dialog for alerts/errors

TECHNISCHE HOCHSCHULE LÜBECK

- Windows and dialogs: example




```
import javax.swing.JFrame;

public class JFrameTest {

        public static void main(String[] args) {

                JFrame testFrame = new JFrame();
                testFrame.setSize(200, 150);
                testFrame.setTitle("TestFrame");
                testFrame.setVisible(true);
        }

}
```

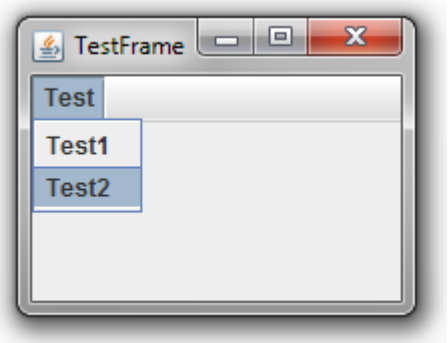
TECHNISCHE HOCHSCHULE LÜBECK

- **Menus**

Class	Description
JMenuItem	Defines one menu item
JMenu	Container for JMenuItems
JMenuBar	Container for Jmenus
JCheckBoxMenuItem	Like JMenuItem, using checkboxes
JRadioButtonMenuItem	Like JMenuItem, using radio buttons
JPopupMenu	Context menu for right-clicking
JSeparator	Dividing line for structuring menus

TECHNISCHE HOCHSCHULE LÜBECK

- Menus: example
  - (add to previous JTestFrame example)




```
JMenuBar testBar = new JMenuBar();

JMenu testMenu = new JMenu("Test");
JMenuItem testItem1 = new JMenuItem("Test1");
JMenuItem testItem2 = new JMenuItem("Test2");
testMenu.add(testItem1);
testMenu.add(testItem2);

testBar.add(testMenu);
testFrame.setJMenuBar(testBar);
```

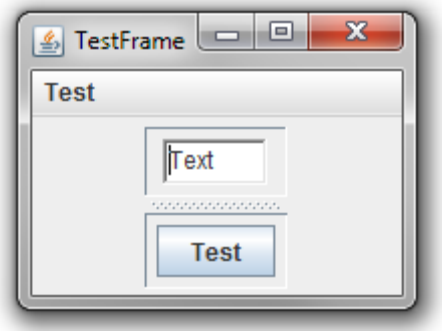
- **Container**

Class	Description
JPanel	Basic container
JTappedPane	Container for multiple container sorted by tabs
JSplitPane	Container divided in two parts (hor. or vert.)
JScrollPane	Enables scrolling in container
JToolBar	Enables tool bar in container
JDesktopPane	Can contain internal sub-windows (JInternalFrames)
JInternalFrame	Added by JDesktopPane
JLayeredPane	Like Jpanel, possibility of moving components to background/foreground

TECHNISCHE HOCHSCHULE LÜBECK

- Container: example
  - (add to previous example Menu example)

```
testFrame.getContentPane().setLayout(new FlowLayout());

Panel p1 = new Panel();
p1.add(new TextField("Text"));
Panel p2 = new Panel();
p2.add(new JButton("Test"));

JSplitPane jsplit = new JSplitPane(JSplitPane.VERTICAL_SPLIT, p1, p2);

testFrame.getContentPane().add(jsplit);
```



TECHNISCHE HOCHSCHULE LÜBECK

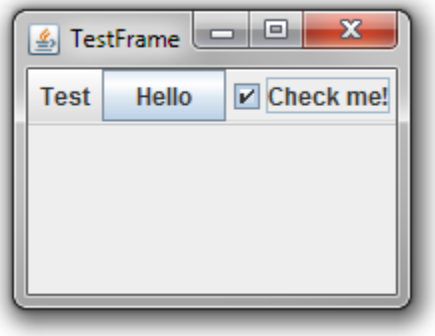
- **Control elements**

Class	Class
JLabel	JScrollBar
JButton	JSlider
JToggleButton	JProgressBar
JCheckBox	JFormattedTextField
JRadioButton	JPasswordField
ButtonGroup	JSpinner
JComboBox	JEditorPane
JList	JTextPane
JTextField	JTree
JTextArea	JTable
JSeperator	

TECHNISCHE HOCHSCHULE LÜBECK

- Control elements: example
  - (add to previous Menu example)

```
JButton testButton = new JButton("Hello");
JCheckBox testCB = new JCheckBox("Check me!");

testBar.add(testButton);
testBar.add(testCB);
```

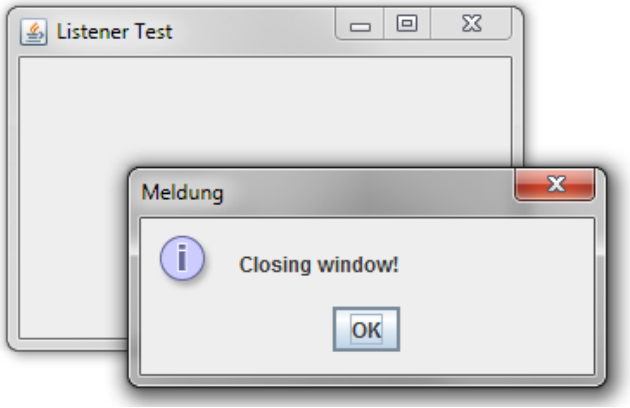


- **Event handling**

Class	Description
ActionEvent	Triggers event, e.g. pressing button
AdjustmentEvent	Triggers event, e.g. changing scroll bar
FocusEvent	If component loses or gets focus
ItemEvent	Selecting check box items
MouseEvent	Pushing mouse button
TextEvent	Change of text in text field
WindowEvent	Change of window state (open, close)
MenuEvent	Menu interaction (open, select)
ListSelectionEvent	Selected list area
TableModelEvent	Change of table items

TECHNISCHE HOCHSCHULE LÜBECK

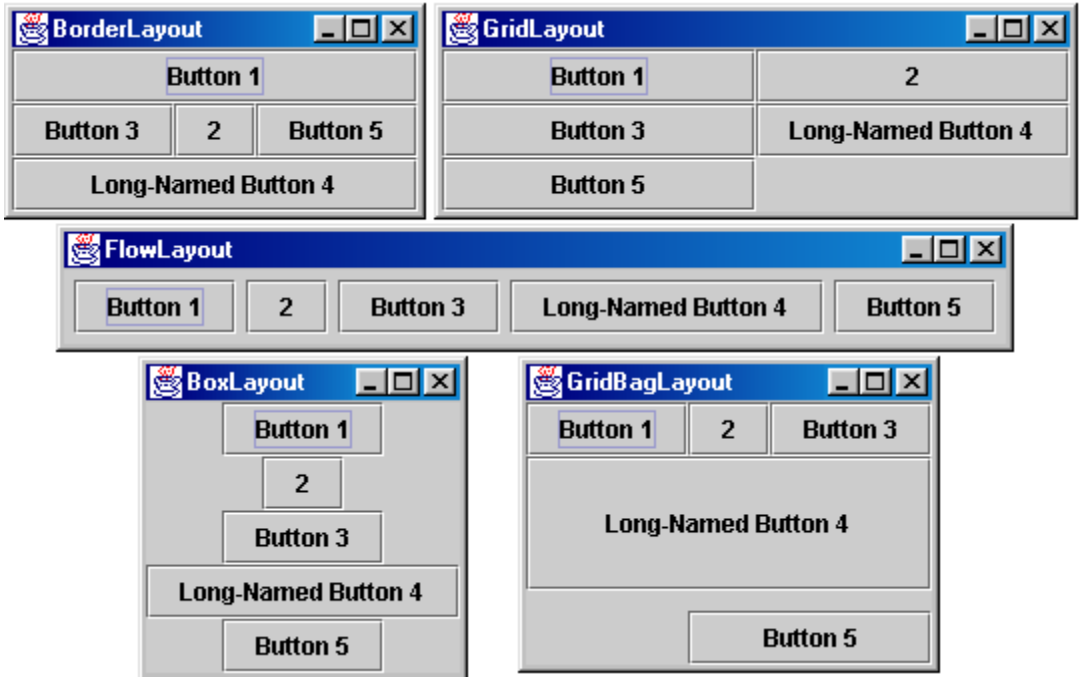
- **Event handling: Listener**




```
JFrame testFrame = new JFrame("Listener Test");
testFrame.setSize(300, 200);

testFrame.addWindowListener(new WindowAdapter() {

        public void windowClosing(WindowEvent e) {
                JOptionPane testOpt = new JOptionPane();
                testOpt.showMessageDialog(new JFrame(), "Closing window!");
        }

});

testFrame.setVisible(true);
```

TECHNISCHE HOCHSCHULE LÜBECK

Class
FlowLayout
GridLayout
BorderLayout
BoxLayout
CardLayout
GridBagLayout
GroupLayout
SpringLayout



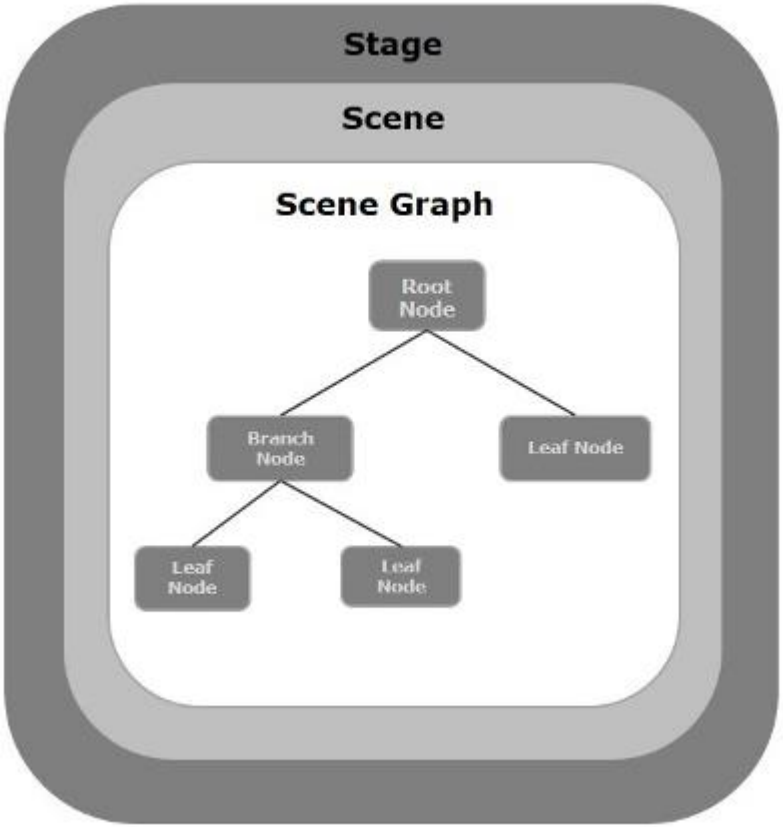
http://users.dcc.uchile.cl/~lmateu/CC60H/Trabajos/edavis/images/allLayouts.gif

testFrame.getContentPane().setLayout(new FlowLayout());

(from previous container example)

TECHNISCHE HOCHSCHULE LÜBECK

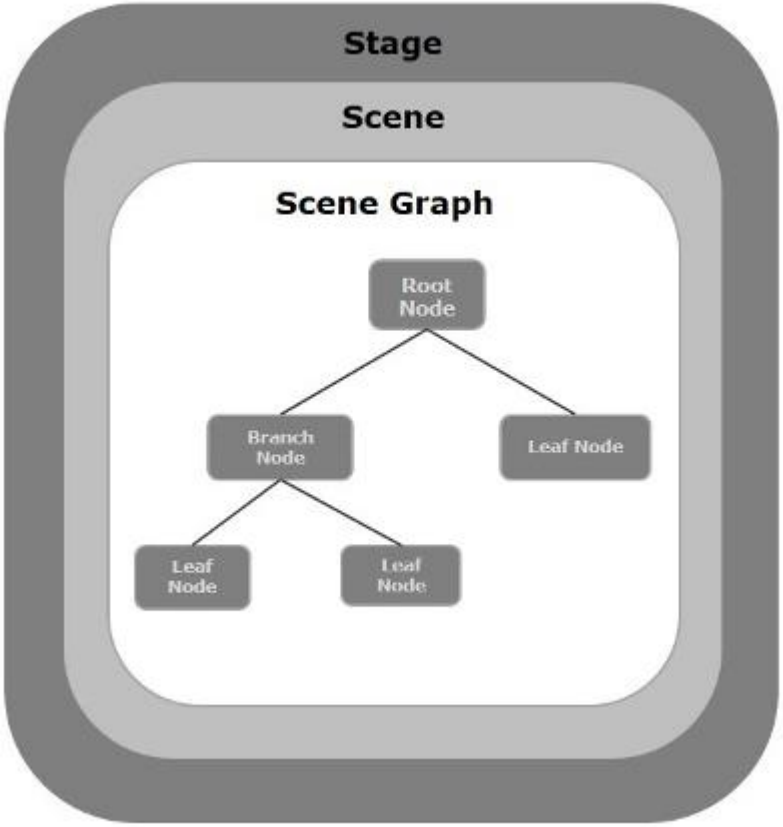
- **Application is a stage**
  - Theater metaphor
  - Stage represents typically a window
- **Individual controls which make up UI are displayed in a scene**
  - More than one scene on a stage allowed, but only one can be shown at the same time



https://www.tutorialspoint.com/javafx/javafx_application.htm

TECHNISCHE HOCHSCHULE LÜBECK

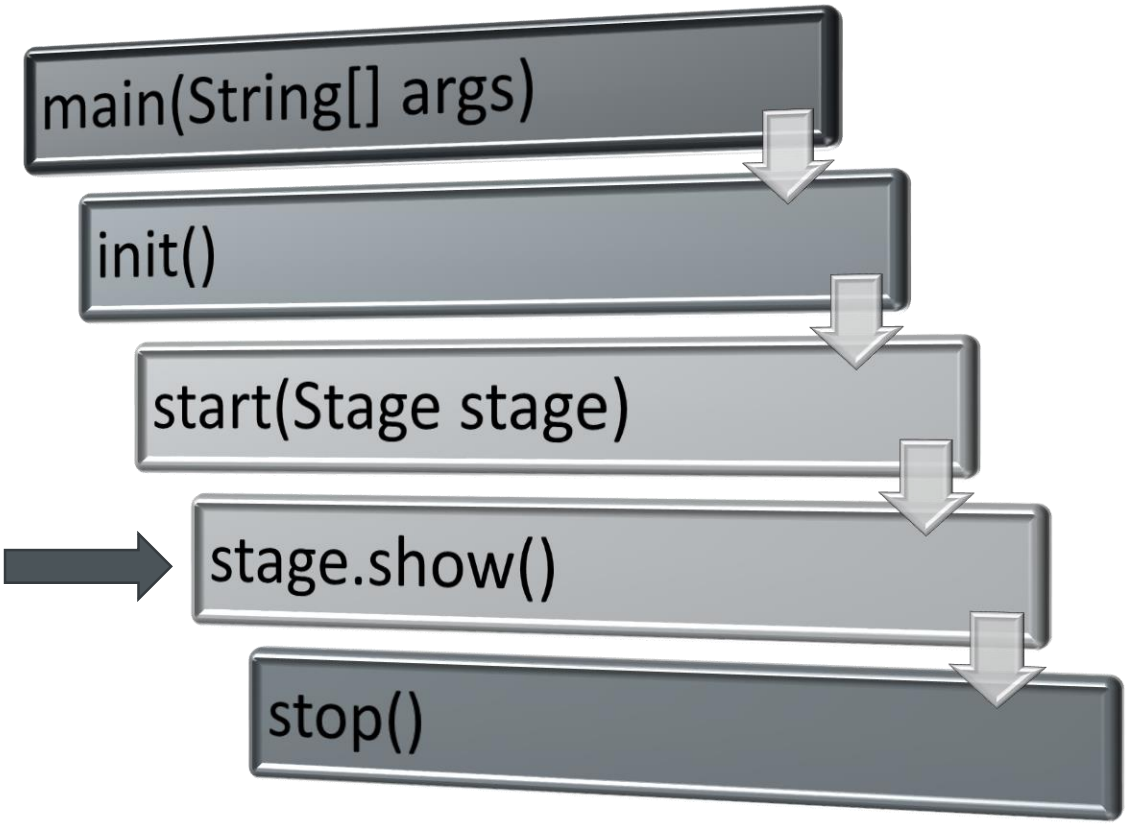
- **Nodes in scene graph**
  - Geometrical objects 2D/3D
  - UI controls like Button, Checkbox, Text Area
  - Containers (layouts)
  - Media elements like audio, video and image objects



https://www.tutorialspoint.com/javafx/javafx_application.htm

TECHNISCHE HOCHSCHULE LÜBECK

- **Lifecycle**



Call show to display content of a stage!

TECHNISCHE HOCHSCHULE LÜBECK

- **Application frame**

javafx.application

```
public class JavafxSample extends Application {
  @Override
  public void start(Stage stage) throws Exception {
    /*
    Code for JavaFX application.
    (Stage, scene, scene graph)
    */
  }
  public static void main(String args[]){
    launch(args);
  }
}
```

public void start(Stage stage) throws Exception {

Prepare scene graph with nodes

- E.g. Group root = new Group();

Prepare scene with dimensions and add scene graph to it

- E.g. Scene scene = new Scene(root, 600, 300);

Prepare stage and add the scene to the stage and display the contents of the stage

- E.g. stage.setTitle("Test");
- stage.setScene(scene);
- stage.show();

}

TECHNISCHE HOCHSCHULE LÜBECK

- **For Eclipse you need**
  - JavaFX Runtime e(fx)clipse

1. In Eclipse, click on *Help* → *Install New Software…*
2. In the *‚Work with' drop down* choose release repository and search for e(fx)clipse
3. Install and restart
4. Under *File* → *New* create new JavaFX project

REFERENCES

TECHNISCHE HOCHSCHULE LÜBECK

- [6-7,11-31] Jakob Nielsen, Usability Engineering, 1993, Academic Press
- [32-36] Heinrich Balzert, Lehrbuch der Softwaretechnik, 2000, Akademischer Verlag
- [40] Marc Loy, Robert Eckstein, Dave Wood, James Elliott, Brian Cole, Java Swing (2 ed.), 2012, O'Reilly Media
- https://www.tutorialspoint.com/javafx/javafx_overview.htm

TECHNISCHE HOCHSCHULE LÜBECK

- **Create a small application using Swing and JavaFX**
  - Main window: user can choose from 6 different fruits/books/bands which are the user's favorite ones
  - After confirming the choice of fruits another window will pop up and tell the user the choice of fruits
  - Feel free to alter the application!

  - **Java Swing Tutorials**
    - http://www.wideskills.com/java-tutorial/java-swing-tutorial
    - http://zetcode.com/tutorials/javaswingtutorial/
  - **JavaFX Tutorials**
    - https://www.tutorialspoint.com/javafx/javafx_application.htm
    - http://tutorials.jenkov.com/javafx/your-first-javafx-application.html

REMINDER: GRADING

TECHNISCHE HOCHSCHULE LÜBECK

- **Portfolio examination**
  - Project 50%
    - Oral/participation grade 10%
    - Written part and application 40%
  - Written exam 50%

- **Project and written exam must be passed with at least 4.0 to pass the whole course!**
- If you fail the project, you have to repeat it one year later (prolongation of studies!)
- If you fail the written part in June, you can repeat it twice (once in September/October and once in February)

TECHNISCHE HOCHSCHULE LÜBECK

- **Needed knowledge**
  - Java
  - UML
  - DB
  - Design patterns
  - Test
- **Project work in groups**
  - Partial tasks for every week
  - Presentation at the end
  - Submission of final project (Project report, code including tests, DB dump)

PROJECT ORGANIZATION

TECHNISCHE HOCHSCHULE LÜBECK

- 10 Groups of 3 students and one of 2 students (assigned)
- Every week explanation of part assignment during SWE II lecture
- Every week consultation with supervisor about project (~ 30 minutes)
- At the end: every group presents their project, each student presents a part!

TECHNISCHE HOCHSCHULE LÜBECK

- **Grading includes**
  - Project report → group/individual grading
  - Presentation/oral participation → individual grading

- **Project report**
  - Specification
  - UML (e.g. Use Cases, Class diagram)
  - GUI Design (Screenshots and explanation)
  - Test description
  - Evaluation + who did which part

TECHNISCHE HOCHSCHULE LÜBECK

- **Digital cookbook**
  - Stores multiple recipies
  - Recipies have ingredients
  - Ingredient has …
  - Everything stored in DB

http://static1.squarespace.com/static/522a32bee4b02163b5a2e6de/522a4464e4b0572b247cb29d/522a44b9e4b0572b247cc2f7/1379342776368/cookbook-cover.jpg%3Fformat%3D1000w

TECHNISCHE HOCHSCHULE LÜBECK

Hong Shao Rou—Red Braised Pork Belly

Prep Time: 10 minutes
Cook Time: 1 hour

Serve: 4

Ingredients

- 500g pork belly cut into cubes around 2 inches
- 4 tablespoons light soy sauce
- 2 tablespoons brown sugar, broken if you have large pieces
- 2 inches ginger, cut into slices
- 4 green onions, 1 finely chopped for garnish and the left into long sections
- 1 cup hot water
- Oil for brushing (optional if you are using iron wok)
-

Instructions

1. Clean and cut the pork belly into cubes around 2 inches long.
2. Boil a large pot of water, add 2 slices of ginger and 2 green onions, cook the pork belly for around 4 minutes. Transfer out and wash with warm water. Set aside and drain.
3. Heat up wok on medium fire; brush some oil on the bottom. Sautee the pork belly until the surface becomes slightly brown. Transfer the pork cubes out to a pre-heat clay pot with green onion and ginger slices laid in bottle or a plate and leave the oil in.

TECHNISCHE HOCHSCHULE LÜBECK

- **Main tasks for each group**
  - Add new recipe/ingredient/preparation step
  - Edit recipe/ingredient/preparation step
  - Delete recipe
  - Search for recipe by name
  - Display recipe with picture
    - Change serve amount -> change of ingredient amounts

- **Each group will have to implement own additional functionality, e.g.**
  - Export recipes into PDF cookbook
  - Search for ingredients

→ Your ideas?

TECHNISCHE HOCHSCHULE LÜBECK

1. **Write digital cookbook specification**
2. Create business layer
3. Create data access layer
4. Connect business and data access layer
5. Implement JUnit tests for your code so far
6. Design GUI
7. Connect GUI and business layer
8. Create test cases for GUI
9. Evaluate usability tests → change software?
10. Prepare presentation

SWE II – PROJECT SCHEDULE

TECHNISCHE HOCHSCHULE LÜBECK

- **CW 16 deadline for specification**
- CW 17 deadline for use case diagrams
- CW 18 deadline for class diagrams
- CW 19 introduction into coding part of project and deadline for ER model
- …
- Later: introduction into DB and GUI connection
- **18.6. deadline for project upload**
- 20.6. project presentations

FINAL GROUPS

TECHNISCHE HOCHSCHULE LÜBECK

Group 1 (IB)	Shi Tianzhe	Wang Yuting	Xiao Zijian
Group 2 (AW)	Chen Zidi	Qiu Mengke	Yang Bowen
Group 3 (IB)	Hou Shiying	Song Ge	
Group 4 (LK)	Liu Jiayu	Wu Hao	Zhou Yuhan
Group 5 (AW)	Shi Wenjie	Qiu Sixiang	Wu Qinyang
Group 6 (MG)	Chen Sihan	Ling Wei	Shen Yu
Group 7 (MG)	Huang Zili	Wang Yiqi	Yang Yifan
Group 8 (LK)	Huang Hao	Song Ce	Tian Hao
Group 9 (MG)	Liu Tianyuan	Wu Tong	Xiong Yiqiu
Group 10 (IB)	Fu Minyan	Li Zongdi	Shao Yifan
Group 11 (AW)	Cai Yiwei	Liu Jingyan	Shen Jiashun

→ Seat yourself in groups now!

TECHNISCHE HOCHSCHULE LÜBECK

- **Elect group leader**
  - Responsible for communication in group
  - Responsible for communication with supervisor if there are issues
  - Responsible for Moodle uploads!
  - No influence on final grade

- Come up with a **group/project name**
  - **Group leader sends me an email with your group name and leader now! (lenka.kleinau@th-luebeck.de)**

→You have 10 minutes!

TECHNISCHE HOCHSCHULE LÜBECK

- **Exercise in groups**
  1. Develop idea for additional task
  2. Inform me about it (I have to approve)

- **Individual exercise: Create a small application using Swing and JavaFX**
  - Main window: user can choose from 6 different fruits/books/bands which are the user's favorite ones
  - After confirming the choice of fruits another window will pop up and tell the user the choice of fruits
  - Feel free to alter the application!

ASSIGNMENTS FOR NEXT WEEK

TECHNISCHE HOCHSCHULE LÜBECK

- **Upload your specification until next lecture! (Deadline 17.4.2019, 20 o‘clock)**

- **Group leader writes me an email with the following information:**
  - **Group members + leader**
  - **Name of group**
  - **Additional task**

- **I will forward the email to your supervisor, they will reply and arrange the specific time/day for the weekly project meetings**

→ Be sure to check your emails!!

TECHNISCHE HOCHSCHULE LÜBECK

Next SWE II lecture:

18.4.2019, 12:15 – 15:30, room 1-1.11

https://www.reddit.com/r/ProgrammerHumor/

Questions?

Contact: lenka.kleinau@th-luebeck.de, Room 18-2.02