48024 Applications Programming

Dr Angela Huo

Zoom Manner

- Chat: Message the lecturer if something is wrong
- Icon: Slow down or faster?
    - When you have a question, click “Raise hand”, I will unmute you and you can speak
    - Vote for a poll, click “Yes” or “No”
- Padlet: https://padlet.com/angelahuo/appsprog
    - I will answer the posted questions at the end of the lecture.

This subject adopts “flipped learning” teaching strategy. You need to complete the “Pre-class” activities to prepare for the interactive quiz and application questions in lectures, which help you prepare for the LMS exam.

Contents

Pre: Open Question Board(https://padlet.com/angelahuo/appsprog)

- Announcement
- Key points of Study 11
- Week 12 Preview
- Question Board

Post your question to https://padlet.com/angelahuo/appsprog

Announcement

- Consultation session runs pretty well!
  - All questions are addressed and solved!
  - This week's consultation will run at **3:30pm on Tuesday**!
- Book your 1:1 meeting on Tuesday/Friday through the page:
  - https://outlook.office365.com/owa/calendar/ApplicationsProgramming2022AUT@studentutsedu.onmicrosoft.com/bookings/
- Lab 6, Study 7 and Lab 8 Marks are released tonight(23:59pm)

Post your question to https://padlet.com/angelahuo/appsprog

STUDENT FEEDBACK SURVEY (SFS)

UTS
UNIVERSITY OF TECHNOLOGY SYDNEY

- **What is it?**

  Confidential online survey for each subject at the end of semester.

- **Why do it?**

  Give academic staff feedback on your learning experience in each subject.
  Improve the learning experience for all students.

- **How do I do it?**

  Log in at www.sfs.uts.edu.au

- **What else do I need to know?**

  Please be constructive in your feedback!
  By participating, you could win a prize and support a charity.

THANK YOU!

Post your question to https://padlet.com/angelahuo/appsprog

Week 12 Preview

12th May	13th May	17th May	19-20th May	Delay Penalty	8th June	9th June	10th June
TMSfx Feedback Support	TMSfx Feedback Support	Assignment 2 Due (25%)	Demo Check				
	Consultation	Consultation	Consultation		Consultation	Sample Quiz Due	
						Timed LMS Due(20%)	
							Advanced Challenges Due (6%)

Post your question to https://padlet.com/angelahuo/appsprog

GUI Table

- TableView
- Change Listeners
- Catching Exceptions
- Throwing Exceptions

Post your question to https://padlet.com/angelahuo/appsprog

TableView

- Linking a TableView to the model
  - MUST Expose a “customer” property in the controller
  - MUST Expose an “accounts” property in the customer model

- Linking each TableColumn to a model property
  - Use a PropertyValueFactory to link the column to a property value
  - MUST Expose the corresponding properties in the model

Post your question to https://padlet.com/angelahuo/appsprog

Cell value factories

- A cell value factory generates the contents of a cell. Two options:
  - PropertyValueFactory is a cell value factory that just displays a property value.
  - Define your own custom cell value factory to display data how you want.
- Setting a custom cell value factory
  - Assign an ID to the column
  - In your controller:

```
@FXML private TableColumn<Account, String> balanceClm;
@FXML private void initialize() {
balanceClm.setCellValueFactory(cellData ->
cellData.getValue().balanceProperty().asString("$%.2f"));
}
```

Post your question to https://padlet.com/angelahuo/appsprog

Change Listeners

- Interface ChangeListener<X>
- Register observer with observable.addListener(observer);

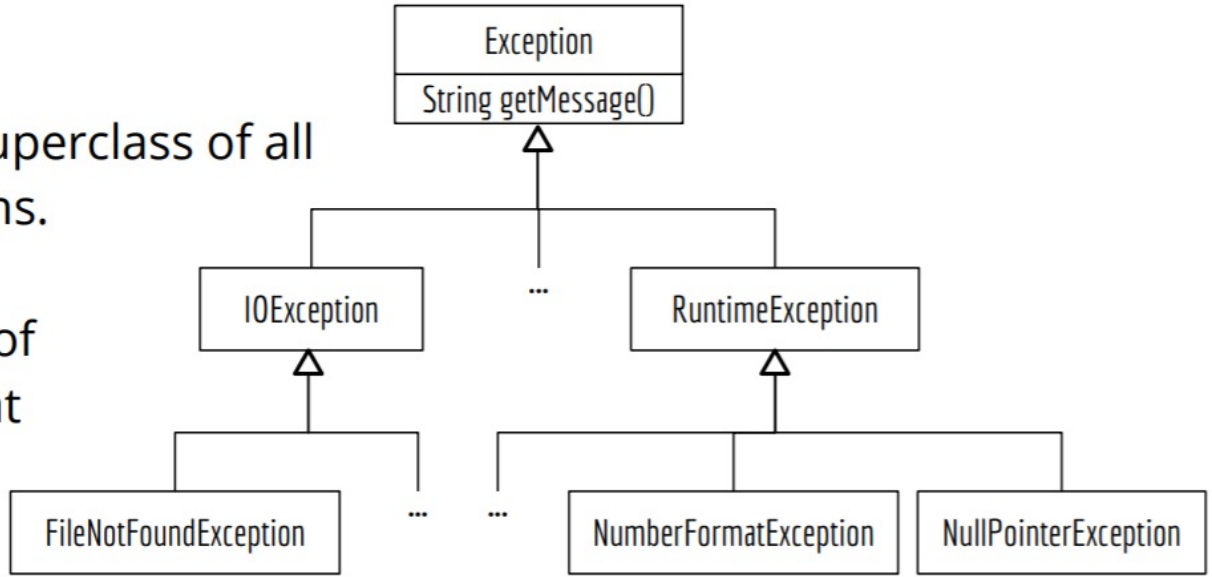
- Goal #1: Enable button when account is selected
    - Update the disable property of the button whenever the selectedItem property changes.

- Goal #2: Enable button when Type >= 4 characters
    - Update the button's disable property when the text property changes.

Post your question to https://padlet.com/angelahuo/appsprog

Exception Inheritance Hierarchy

- Class `Exception` is the superclass of all exceptions.
- `IOException` is the superclass of all input/output exceptions.
- Catching a superclass of exceptions catches that entire category of exceptions



Catching Exceptions

```
try {
scanner = new Scanner(new File(“data.txt”));
int a = Integer.parseInt(scanner.nextLine());
int b = Integer.parseInt(scanner.nextLine());
int c = a / b;
System.out.println(a + “ / “ + b + “ = “ + c);
} catch (Exception e) {
System.out.println(“An error occurred: “ + e.getMessage());
} finally {
if (scanner != null) scanner.close();
}
```

Post your question to https://padlet.com/angelahuo/appsprog

Throwing an exception

- Throwing a generic exception

  The method header specifies a comma-separated list of exceptions it can throw.

  Eg: public void foo() throws IOException, NumberFormatException

- Throwing a custom exception

  - Define a custom exception as a subclass of Exception.

```
public void withdraw(double amount) throws InsufficientFundsException {
    if (amount > balance.get())
            throw new InsufficientFundsException();
    balance.set(balance.get() - amount);
}
```

Post your question to https://padlet.com/angelahuo/appsprog

The catch or specify requirement

- If you write code that might throw an exception, you must either specify that the exception might be thrown, or catch that exception.
- If you don’t catch the exception, you push back the catch or specify requirement to the caller.
- If an exception is thrown all the way to the top level without being caught, the user sees a stack trace

- General practice: throw an exception up as high as possible, but catch and handle it before the user sees the stack trace.

Post your question to https://padlet.com/angelahuo/appsprog

Unchecked exceptions

Unchecked exceptions are not subject to the catch or specify requirement. There are two kinds:

- Any subclass of **RuntimeException** is unchecked. Runtime exceptions are typically due programming bugs. Examples:
  - NullPointerException
  - NumberFormatException
  - ArrayIndexOutOfBoundsException
- Any subclass of **Error** is unchecked. Errors are failures of the environment. Examples:
  - OutOfMemoryError
  - IOError

Post your question to https://padlet.com/angelahuo/appsprog

Unchecked exceptions

- **RuntimeException**
  - NullPointerException
  - NumberFormatException
  - ArrayIndexOutOfBoundsException
- **Error**
  - OutOfMemoryError
  - IOError

The following question is based on this code:

```
Student student = null;
try {
      System.out.println(student);
      System.out.println(student.getName());
} catch (Exception e) {
      System.out.println(null);
} finally {
      System.out.println(null);
}
```

Post your question to https://padlet.com/angelahuo/appsprog

Lab 11

- 5-10min Intro/Demo
- 30min Lab exercise
- Assignment support - remaining time

You will learn:

- TableView
- TableColumn
- Cell value factory
- PropertyValueFactory

Post your question to https://padlet.com/angelahuo/appsprog

Lab 11

NOTE：

- if an element's tag name begins with an uppercase letter, then that element represents a class instance.
- If an element's tag name begins with a lowercase letter, then that element represents an object property.

- More advanced examples can be found in the lecture code download on the Patterns Book page.

Post your question to https://padlet.com/angelahuo/appsprog

Misconduct consequence is severe!!!

- High similarity submissions and suspicious cases will be reported to the committee.

- The committee report process is extremely tedious and will cause delays in your studies (and perhaps visa complications for international students).
    - You won't receive the mark within a year.
    - Your academic life will risk a failure ending.

Post your question to https://padlet.com/angelahuo/appsprog

Contact

- Subject Coordinator and Lecturer: Angela Huo
- Email: huan.huo@uts.edu.au
- Contact information on Canvas

See you next week!

Post your question to https://padlet.com/angelahuo/appsprog