ITA 2. Semester

Introduktion

NIFR 2022

Dagsorden

Opstart

- Fagområder, læringsmål & projekter
- Eksamen
- Kompilerede programmer
- Java Virtual Machine
- Programmeringssproget: Java
  - Typer & Syntax

Fagområder

2. Semester

- Applikationsudvikling
  - CS101
  - Android Development
- Agil Udvikling
- Digital Kultur 2
- Brugerinddragelse og Design

1	2	3	4	5	6	7
DATA OG DATAVISUALI-SERING	APPLIKATIO-NER OG APPLI-KATIONSUD-VIKLING	SYSTEMER OG SYSTEMINTE-GRATION	ARKITEKTUR OG ARKITEK-TURUDVIKLING	SPECIALISE-RING	PRAKTIK	BACHELOR-PROJEKT

2. SEMESTER: APPLIKATIONER OG APPLIKATIONSUDVIKLING

Vi bygger applikationer med fokus på anvendelse af et programmeringssprog og -miljø, hvordan data kan gemmes og manipuleres i et databasesystem og den arbejdsproces som bringer os fra krav til fungerende IT-system.

På semestret arbejder vi fx med spørgsmålene:

- Hvilke værktøjer anvender vi til at bygge en applikation?
- Hvordan gennemfører vi den arbejdsproces, som bringer os fra krav og behov til fungerende applikation?

2 SEMESTER

APPLIKATIONSUDVIKLING MED BRUGERINDDRAGELSE - EKSTERN

30 ECTS

- → APPLIKATIONSUDVIKLING
- → BRUGERINDDRAGELSE OG DESIGN
- → AGIL UDVIKLING
- → DIGITAL KULTUR 2

Kombineret skriftlig og mundtlig eksamen, projekteksamen

Digital Kultur 2

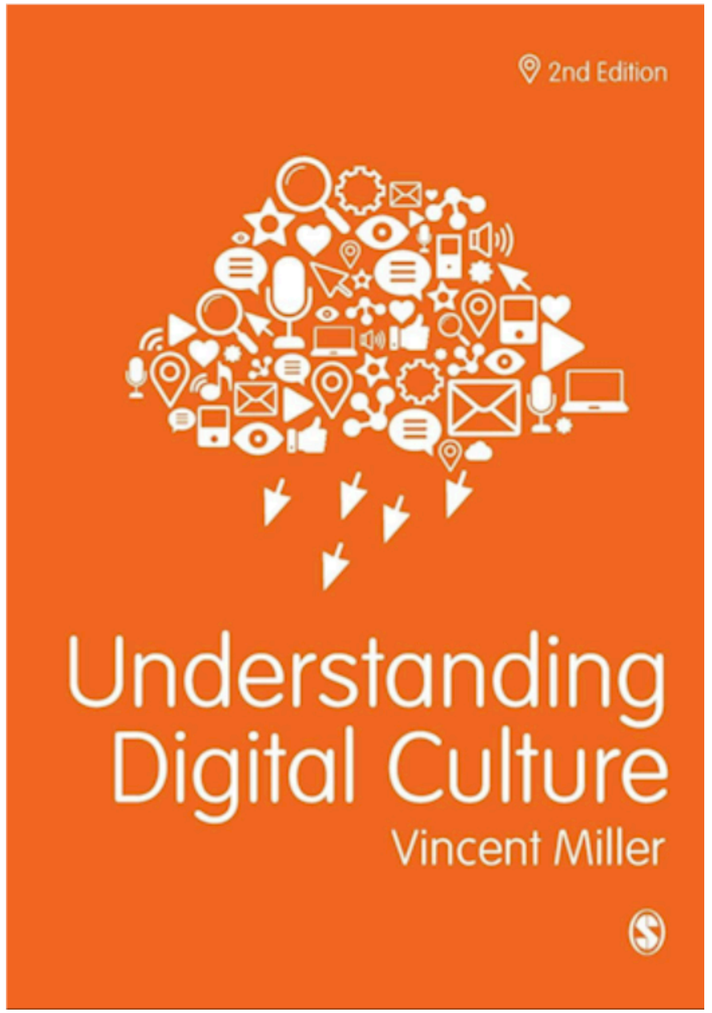
Den studerende har:

- viden om hvordan forskellige informationsteknologier påvirker samfund og kultur, såvel som eget liv
- viden om anvendt teori og metode omkring informationsteknologiens indflydelse på kultur og borger
- viden om hvordan værdi flyttes og skabes i relationen mellem borger of privat virksomhed i en digital kultur.
- viden om og forståelse for teoretiske digitaliseringsmodeller samt kan reflektere over hvordan de fungerer i praksis

Viden

Hvem?

- Dr. Vincent Miller
- Sociologi og Kulturstudier
- Digital Kultur, Media & ‘New Media’
- Social Teori & ‘Memetics’ / Meme Theory



Digital Kultur 2

- vurdere og analysere praksisnære og teoretiske problemstillinger omkring digital kultur samt begrunde og vælge relevante løsningsmodeller
- formidle praksisnære og faglige problemstillinger og løsninger inden for digital kultur til samarbejdspartnere og brugere

Vurdering, Analyse & Formidling

Brugerinddragelse og Design

Lars Kruse

Agil Udvikling

Brian Everitt

Brugerinddragelse & Design

Applikationsudvikling

Fagområde

- Består af 2 komponenter
- CS101 & Androidudvikling
  - CS101 starter I dag
  - Androidudvikling
- CS101 er en forudsætning for Androidudvikling
  - 27. Februar

Teoretiske fundament: Datalogi / Computer Science

Applikationsudvikling

Den studerende har:

- viden om anvendt teori, praksis og metoder inden for programarkitektur, herunder teknikker og metoder inden for programmering af applikationer
- viden om algoritmer, datastrukturer og gængse programmeringsparadigmer

Moneymaker: Softwareudvikling

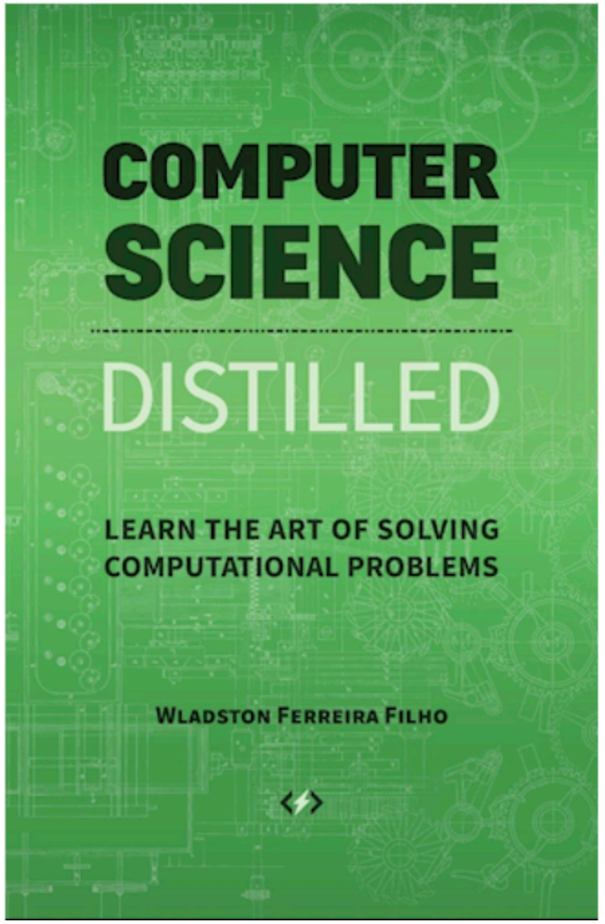
Applikationsudvikling

- anvende metoder og redskaber til udvikling og kvalitetssikring af software.
- anvende relevante programmeringssprog til udvikling af software ved anvendelse af algoritmer, abstraktioner og mønstre.
- anvende værktøjer til modellering og dokumentation af domænespecifikt programkode- og data.
- anvende værktøjer til deling, versionsstyring og samarbejde omkring udvikling af programkode.

CS101

Computer Science

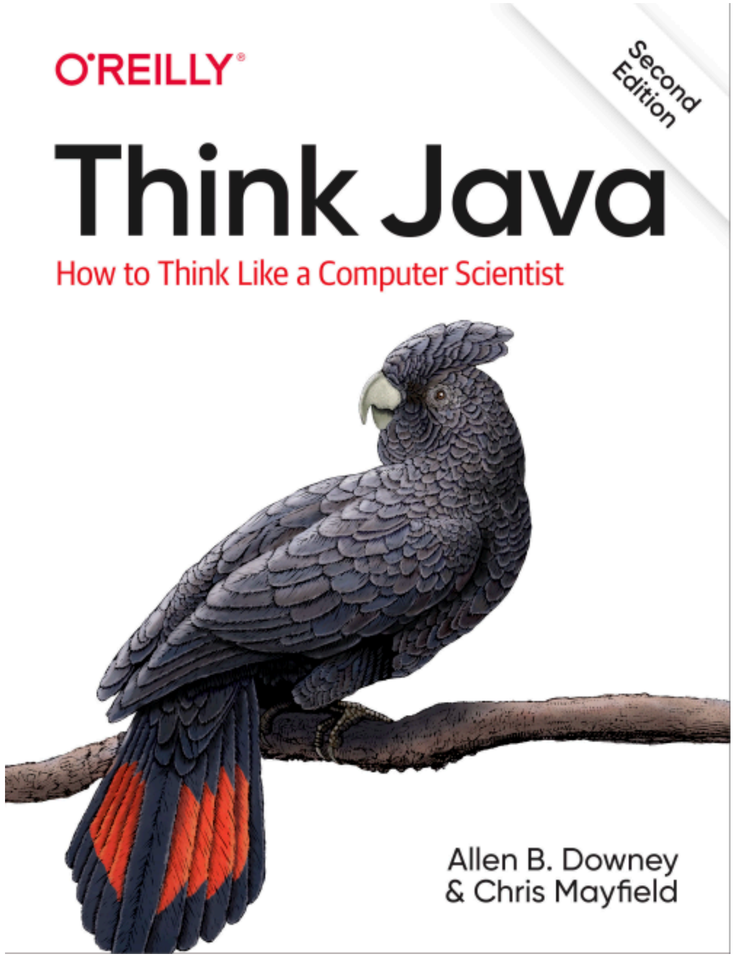
- Destilleret & kortfattet
- Dækker de vigtigste områder af datalogien
- http://www.wladston.net/
- Algoritmer, logik, kompleksitet, paradigmer & strategier



CS101

Computer Science

- Destilleret & kortfattet
- Grundlæggende Java programmering
- Gratis version .pdf
- Læsevejledning: Øvelser
  - https://codingbat.com/java



Computer science != Software Development

Computer Science / Datalogi

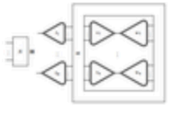
Automata theory

Formal languages

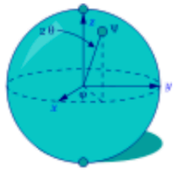
$$M = \{X : X \notin X\}$$

Computability theory

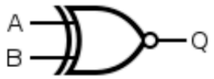
Computational complexity theory



Models of computation



Quantum computing theory

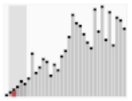


Logic circuit theory

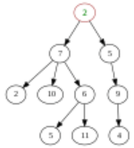
Cellular automata

$O(n^2)$

Analysis of algorithms



Algorithm design



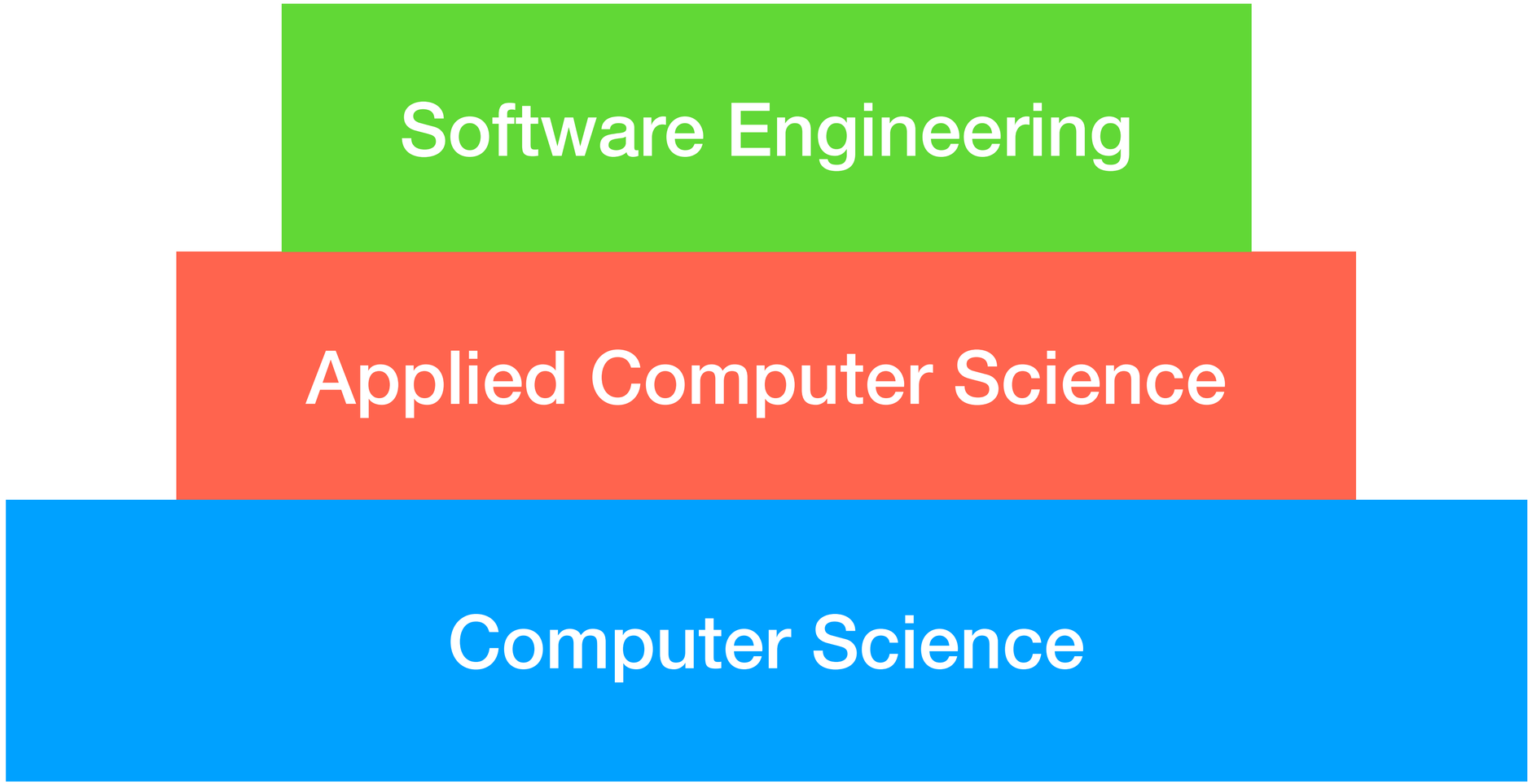
Data structures

Combinatorial optimization

Computational geometry

Randomized algorithms

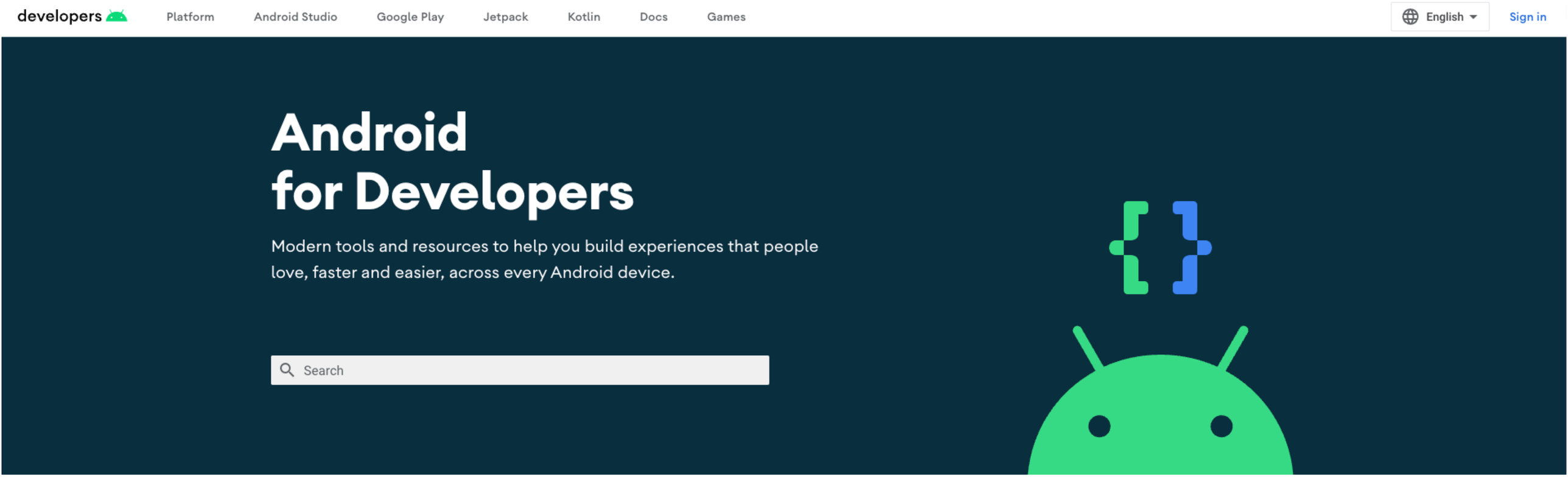
The **P versus NP problem** is a major unsolved problem in theoretical computer science. In informal terms, it asks whether every problem whose solution can be quickly verified can also be quickly solved.


Software Engineering
Applied Computer Science
Computer Science

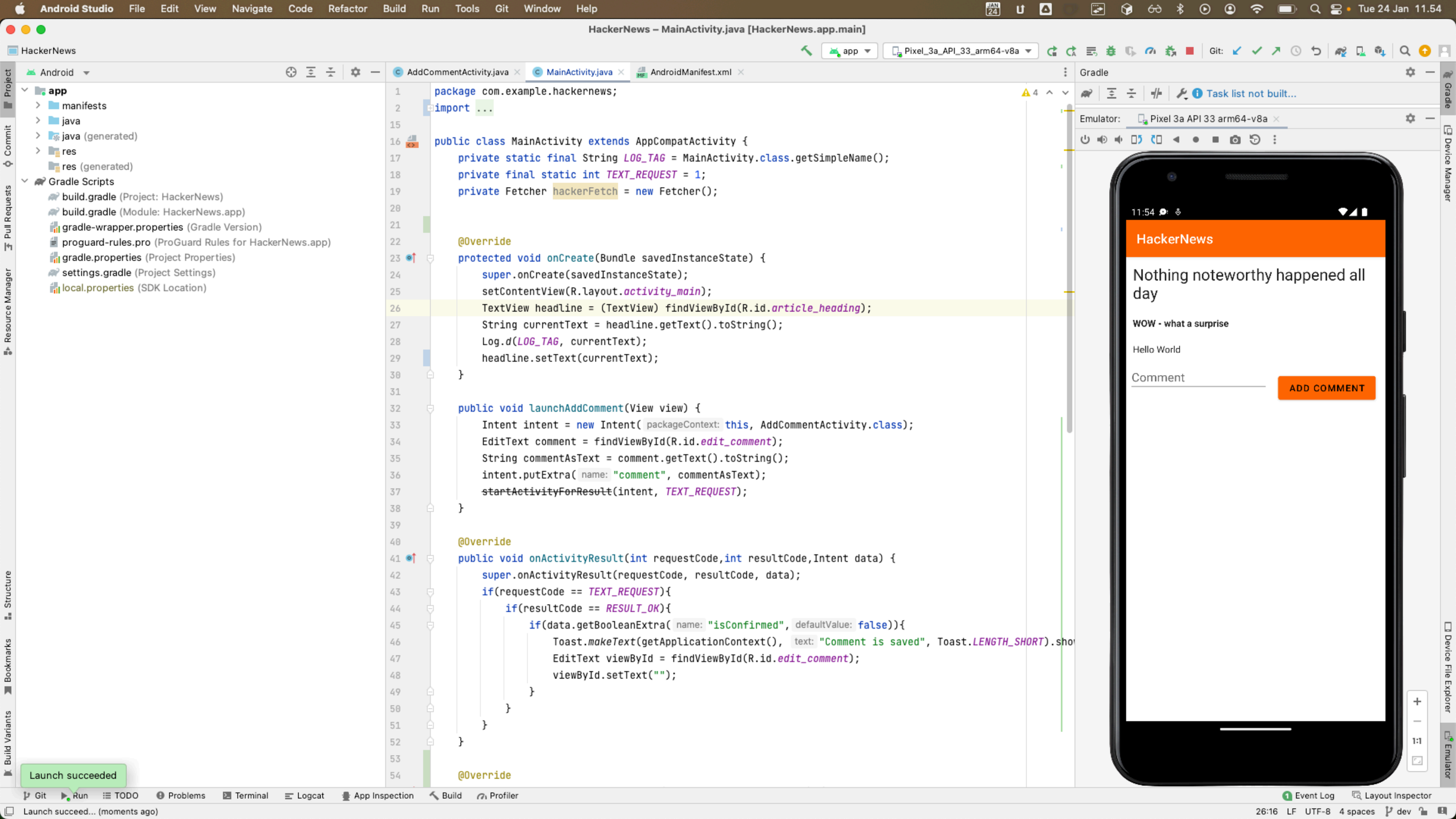
Learning Computer Science with the programming language Java

Næste 3 uger: Ugentlige afleveringer
Assessment test

I kan snyde med ChatGPT! But please don’t



https://developer.android.com/?gclsrc=ds&gclsrc=ds


Android Studio File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Tue 24 Jan 11.54
HackerNews – MainActivity.java [HackerNews.app.main]
HackerNews
app
Pixel_3a_API_33_arm64-v8a
Git:
Android
AddCommentActivity.java
MainActivity.java
AndroidManifest.xml
Gradle
Task list not built...
Emulator:
Pixel 3a API 33 arm64-v8a
app
manifests
java
java (generated)
res
res (generated)
Gradle Scripts
build.gradle (Project: HackerNews)
build.gradle (Module: HackerNews.app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for HackerNews.app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
package com.example.hackernews;
import ...
public class MainActivity extends AppCompatActivity {
private static final String LOG_TAG = MainActivity.class.getSimpleName();
private final static int TEXT_REQUEST = 1;
private Fetcher hackerFetch = new Fetcher();
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
TextView headline = (TextView) findViewById(R.id.article_heading);
String currentText = headline.getText().toString();
Log.d(LOG_TAG, currentText);
headline.setText(currentText);
}
public void launchAddComment(View view) {
Intent intent = new Intent(packageContext: this, AddCommentActivity.class);
EditText comment = findViewById(R.id.edit_comment);
String commentAsText = comment.getText().toString();
intent.putExtra(name: "comment", commentAsText);
startActivityForResult(intent, TEXT_REQUEST);
}
@Override
public void onActivityResult(int requestCode,int resultCode,Intent data) {
super.onActivityResult(requestCode, resultCode, data);
if(requestCode == TEXT_REQUEST){
if(resultCode == RESULT_OK){
if(data.getBooleanExtra(name: "isConfirmed", defaultValue: false)){
Toast.makeText(getApplicationContext(), text: "Comment is saved", Toast.LENGTH_SHORT).sho
EditText viewById = findViewById(R.id.edit_comment);
viewById.setText("");
}
}
}
}
@Override
11:54
HackerNews
Nothing noteworthy happened all day
WOW - what a surprise
Hello World
Comment
ADD COMMENT
Launch succeeded
Git
Run
TODO
Problems
Terminal
Logcat
App Inspection
Build
Profiler
Event Log
Layout Inspector
Launch succeed... (moments ago)
26:16
LF
UTF-8
4 spaces
dev

Google Developers Certification

Search English

Earn a certification to gain recognition for your skills as a developer.

Associate Android Developer

This certification is being updated. New registrations for the exam are currently closed, until the update is complete. If you already registered for the exam but haven't taken it yet, please see the FAQ section below.

Study and prepare

Take the exam

Complete exit interview

Earn your certification

Projekter & Virksomhedsbesøg

2. semester

- 2 Tværfaglige projekter
  - Efter påskeferien: 11/4
  - Eksamensprojekt: 8/5
- Virksomhedsbesøg: Shape

Eksamen

2. Semester

- Projekteksamen
- Rapport 30 sider
- 8/5 - 31/5
- Vejledning
- Gruppeopgave

Welcome CS101

Javascript: Interpreted language

Interpreted Language

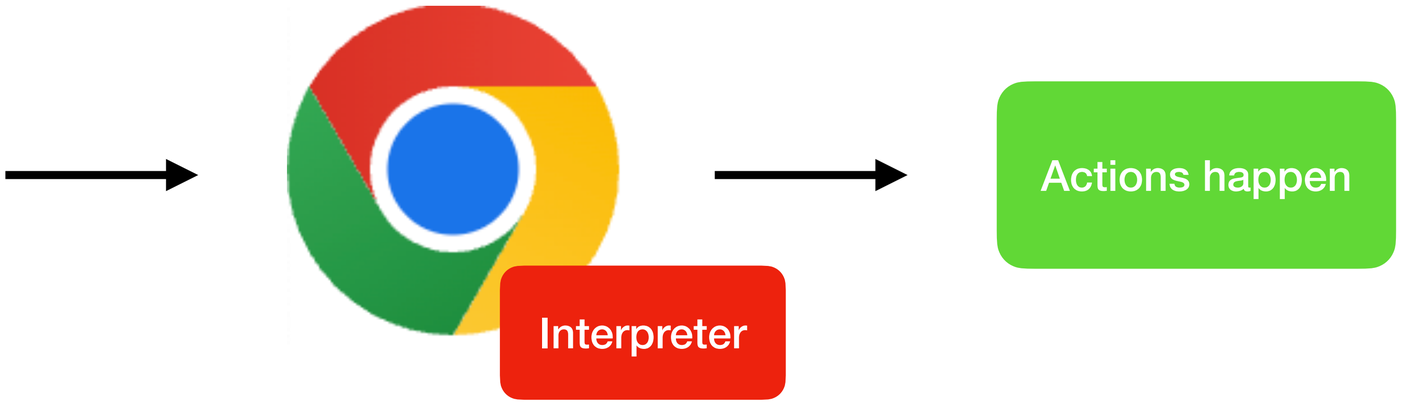
Front-end Javascript

```
let drinks = ['lemonade', 'soda', 'tea', 'water'];
let food = ['beans', 'chicken', 'rice'];
//non DRY code
console.log(drinks[0]);
console.log(drinks[1]);
console.log(drinks[2]);
console.log(drinks[3]);

console.log(food[0]);
console.log(food[1]);
console.log(food[2]);

// DRY code
function logItems (array) {
  for (let i = 0; i < array.length; i++) {
    console.log(array[i]);
  }
}

logItems(drinks);
logItems(food);
```



Compiled Language

Java source code

```java
public class MainActivity extends AppCompatActivity {
    private static final String LOG_TAG = MainActivity.class.getSimpleName();
    private final static int TEXT_REQUEST = 1;
    private Fetcher hackerFetch = new Fetcher();


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView headline = (TextView) findViewById(R.id.article_heading);
        String currentText = headline.getText().toString();
        Log.d(LOG_TAG, currentText);
        headline.setText(currentText);
    }

    public void launchAddComment(View view) {
        Intent intent = new Intent( packageContext: this, AddCommentActivity.class);
        EditText comment = findViewById(R.id.edit_comment);
        String commentAsText = comment.getText().toString();
        intent.putExtra( name: "comment", commentAsText);
        startActivityForResult(intent, TEXT_REQUEST);
    }
```

Compiler

Java Byte code

```
1  cafe babe 0000 003e 004d 0a00 0200 0307
2  0004 0c00 0500 0601 0010 6a61 7661 2f6c
3  616e 672f 4f62 6a65 6374 0100 063c 696e
4  6974 3e01 0003 2829 5607 0008 0100 136a
5  6176 612f 7574 696c 2f41 7272 6179 4c69
6  7374 0a00 0700 0308 000b 0100 0354 776f
7  0a00 0700 0d0c 000e 000f 0100 0361 6464
8  0100 1528 4c6a 6176 612f 6c61 6e67 2f4f
9  626a 6563 743b 295a 0800 1101 0005 5468
10 7265 6508 0013 0100 0446 6f75 7208 0015
11 0100 034f 6e65 0900 1700 1807 0019 0c00
12 1a00 1b01 0010 6a61 7661 2f6c 616e 672f
13 5379 7374 656d 0100 036f 7574 0100 154c
14 6a61 7661 2f69 6f2f 5072 696e 7453 7472
15 6561 6d3b 0a00 1d00 1e07 001f 0c00 2000
16 2101 0013 6a61 7661 2f69 6f2f 5072 696e
17 7453 7472 6561 6d01 0007 7072 696e 746c
18 6e01 0015 284c 6a61 7661 2f6c 616e 672f
19 4f62 6a65 6374 3b29 560a 0007 0023 0c00
20 2400 2501 0003 6765 7401 0015 2849 294c
21 6a61 7661 2f6c 616e 672f 4f62 6a65 6374
22 3b07 0027 0100 106a 6176 612f 6c61 6e67
23 2f53 7472 696e 670a 001d 0029 0c00 2000
24 2a01 0015 284c 6a61 7661 2f6c 616e 672f
25 5374 7269 6e67 3b29 560a 0007 002c 0c00
26 2d00 2501 0006 7265 6d6f 7665 0a00 0700
27 2f0c 0030 0031 0100 0473 697a 6501 0003
28 2829 4912 0000 0033 0c00 3400 3501 0017
29 6d61 6b65 436f 6e63 6174 5769 7468 436f
```

Actions happen (faster)

Compiled Language

Compilation step

Runtime

Java source code

```java
public class MainActivity extends AppCompatActivity {
    private static final String LOG_TAG = MainActivity.class.getSimpleName();
    private final static int TEXT_REQUEST = 1;
    private Fetcher hackerFetch = new Fetcher();


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView headline = (TextView) findViewById(R.id.article_heading);
        String currentText = headline.getText().toString();
        Log.d(LOG_TAG, currentText);
        headline.setText(currentText);
    }

    public void launchAddComment(View view) {
        Intent intent = new Intent( packageContext: this, AddCommentActivity.class);
        EditText comment = findViewById(R.id.edit_comment);
        String commentAsText = comment.getText().toString();
        intent.putExtra( name: "comment", commentAsText);
        startActivityForResult(intent, TEXT_REQUEST);
    }
```

Compiler

Java Byte code

```
cafe babe 0000 003e 004d 0a00 0200 0307
0004 0c00 0500 0601 0010 6a61 7661 2f6c
616e 672f 4f62 6a65 6374 0100 063c 696e
6974 3e01 0003 2829 5607 0008 0100 136a
6176 612f 7574 696c 2f41 7272 6179 4c69
7374 0a00 0700 0308 000b 0100 0354 776f
0a00 0700 0d0c 000e 000f 0100 0361 6464
0100 1528 4c6a 6176 612f 6c61 6e67 2f4f
626a 6563 743b 295a 0800 1101 0005 5468
7265 6508 0013 0100 0446 6f75 7208 0015
0100 034f 6e65 0900 1700 1807 0019 0c00
1a00 1b01 0010 6a61 7661 2f6c 616e 672f
5379 7374 656d 0100 036f 7574 0100 154c
6a61 7661 2f69 6f2f 5072 696e 7453 7472
6561 6d3b 0a00 1d00 1e07 001f 0c00 2000
2101 0013 6a61 7661 2f69 6f2f 5072 696e
7453 7472 6561 6d01 0007 7072 696e 746c
6e01 0015 284c 6a61 7661 2f6c 616e 672f
4f62 6a65 6374 3b29 560a 0007 0023 0c00
2400 2501 0003 6765 7401 0015 2849 294c
6a61 7661 2f6c 616e 672f 4f62 6a65 6374
3b07 0027 0100 106a 6176 612f 6c61 6e67
2f53 7472 696e 670a 001d 0029 0c00 2000
2a01 0015 284c 6a61 7661 2f6c 616e 672f
5374 7269 6e67 3b29 560a 0007 002c 0c00
2d00 2501 0006 7265 6d6f 7665 0a00 0700
2f0c 0030 0031 0100 0473 697a 6501 0003
2829 4912 0000 0033 0c00 3400 3501 0017
6d61 6b65 436f 6e63 6174 5769 7468 436f
```

Actions happen (faster)

Example

Break