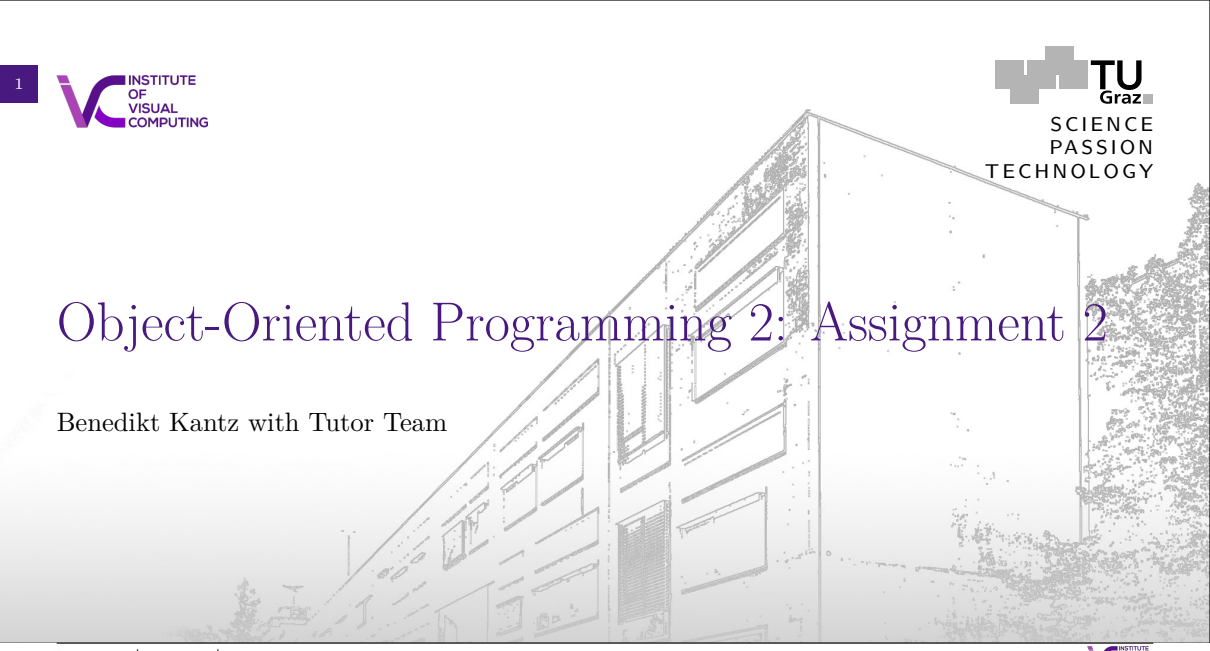
INSTITUTE OF VISUAL COMPUTING

TU Graz
SCIENCE
PASSION
TECHNOLOGY

Object-Oriented Programming 2: Assignment 2

Benedikt Kantz with Tutor Team



Slido Q&A

Practical Schedule

KW	Date	#	KU Topic
40	2.10.2025	00	Modalities
41	9.10.2025	01	Introduction to A0
42	16.10.2025	-	Q&A A0
43	23.10.2025	-	Q&A A0, Deadline A0, A1 Group Sign-Up
44	30.10.2025	02	Intro to A1, Deadline A1 Group Sign-Up
45	6.11.2025	03	Libraries Usage, Q&A A1
46	13.11.2025	-	Q&A A1
47	20.11.2025	-	Q&A A1
48	27.11.2025	-	Q&A A1
49	4.12.2025	-	Q&A A1
50	11.12.2025	04	**Deadline A1, Introduction to A2**
51	17.12.2025	-	Q&A A2
2	8.1.2026	-	Q&A A2
3	15.1.2025	-	Q&A A2, A2 Deadline
4	23.-29.1.2026	-	Assignment Interviews (AG)

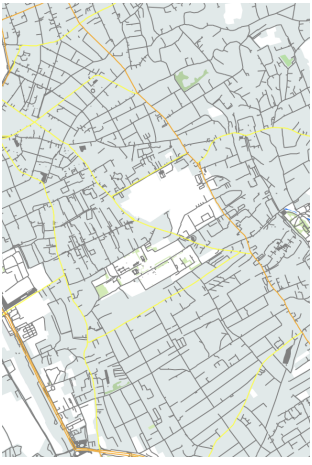
Assignment 2: Introduction I

- Land Usage Requests (4P)
  - Simple Geo Processing Requests
  - Aggragate Data using the JTS
- Routing System (6P)
  - Use the GeoTools to provide route functionality
  - Tutorial: today!

```
Route Found!
Length:  1163.29m
Time:  1.49 min
```

Assignment 2: Introduction II

- Mapping Requests (10P)
  - Provide Mapping Tiles for Web Rendering
  - We provide a strict scheme!
- Heatmap Requests (10P)
  - Combination of analysis and mapping!



Usage

- Calculate & aggregate data
- Use JTS intersection & calculate area correctly
- Use CRS tools (you have used them already!)

Usage

- Calculate & aggregate data
- Use JTS intersection & calculate area correctly
- Use CRS tools (you have used them already!)

GeoTools[a]

[a] geotools.org

- A lot of often-used functionality like parsing of Geo Data (GeoJSON, . . .)
- We use the **routing**!

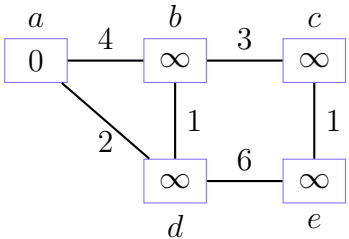
GeoTools[a]

[a] geotools.org

- A lot of often-used functionality like parsing of Geo Data (GeoJSON, . . .)
- We use the **routing**!

Excurs: Dijkstra's Algorithm[1]

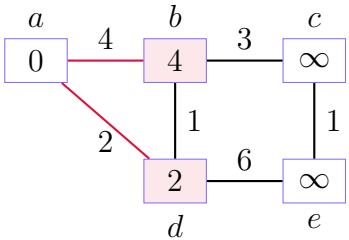
- Shortest Path finding algorithm
- Always take the minimum of all incoming edges



[1] T. H. Cormen and C. E. Leiserson, **Introduction to Algorithms, fourth edition**, 2022

Excurs: Dijkstra's Algorithm[1]

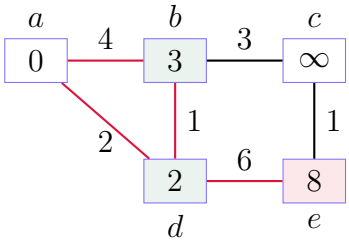
- Shortest Path finding algorithm
- Always take the minimum of all incoming edges



[1] T. H. Cormen and C. E. Leiserson, **Introduction to Algorithms, fourth edition**, 2022

Excurs: Dijkstra's Algorithm[1]

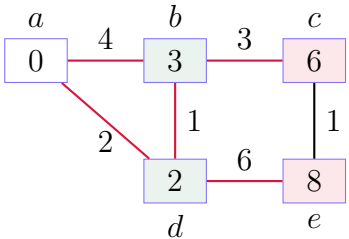
- Shortest Path finding algorithm
- Always take the minimum of all incoming edges



[1] T. H. Cormen and C. E. Leiserson, **Introduction to Algorithms, fourth edition**, 2022

Excurs: Dijkstra's Algorithm[1]

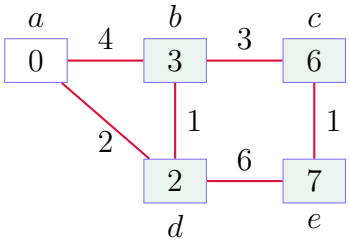
- Shortest Path finding algorithm
- Always take the minimum of all incoming edges



[1] T. H. Cormen and C. E. Leiserson, **Introduction to Algorithms, fourth edition**, 2022

Excurs: Dijkstra's Algorithm[1]

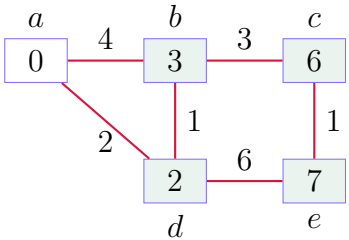
- Shortest Path finding algorithm
- Always take the minimum of all incoming edges



[1] T. H. Cormen and C. E. Leiserson, **Introduction to Algorithms, fourth edition**, 2022

Excurs: Dijkstra's Algorithm[1]

- Shortest Path finding algorithm
- Always take the minimum of all incoming edges



[1] T. H. Cormen and C. E. Leiserson, **Introduction to Algorithms, fourth edition**, 2022

Map Tiles: Slippy Maps

Visual

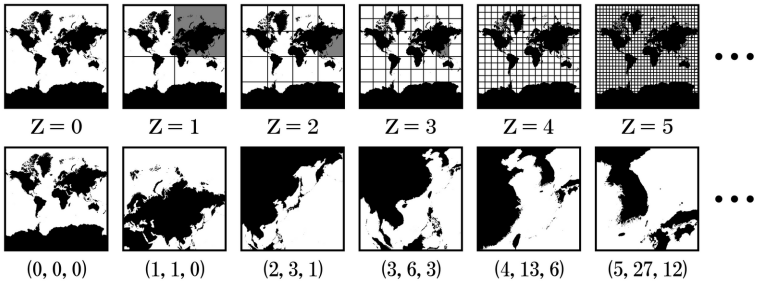


Figure: Web Map Tiles[1]

[1] https://en.wikipedia.org/wiki/Tiled_web_map

Map Tiles: Slippy Maps

Calculation: `/zoom/x/y.png`[1]

Calculation: lat/lon[1]

$$lon = \frac{x}{2^z} \cdot 360 - 180$$

$$lat = \arctan\left(\sinh\left(\pi - \frac{y}{2^z} \cdot 2\pi\right)\right) \cdot \frac{180}{\pi}$$

[1] https://wiki.openstreetmap.org/wiki/Slippy_map_tilenames

Map Tiles: Slippy Maps

Calculation: `/zoom/x/y.png`[1]

Calculation: x/y[1]

$$x = \left\lfloor \frac{lon + 180}{360} \cdot 2^z \right\rfloor$$

$$y = \left\lfloor \left(1 - \frac{\ln\left(\tan\left(lat \cdot \frac{\pi}{180}\right) + \frac{1}{\cos\left(lat \cdot \frac{\pi}{180}\right)} \right)}{\pi} \right) \cdot 2^{z-1} \right\rfloor$$

[1] https://wiki.openstreetmap.org/wiki/Slippy_map_tilenames

Heatmap

- Same as mapping!
- But aggregate over data!

GeoTools
Examples

- Example in KU Examples repository
- Example provided by Assignment Sheet

GeoTools
Examples

- Example in KU Examples repository
- Example provided by Assignment Sheet

Graphics2D
Examples

- Example in KU Examples repository
- *Note*: JTS Geometries can be written to Graphics2D!

Graphics2D
Examples

- Example in KU Examples repository
- *Note*: JTS Geometries can be written to Graphics2D!

Self-assessment & Feedback

Bibliography I

References

[1] T. H. Cormen and C. E. Leiserson, **Introduction to Algorithms, fourth edition**, en. London, England: MIT Press, Apr. 2022.