Automated Software Engineering

Example exam

MŰEGYETEM 1782

BME FACULTY OF ELECTRICAL ENGINEERING AND INFORMATICS

ftsrg

Critical Systems
Research Group

Scoring

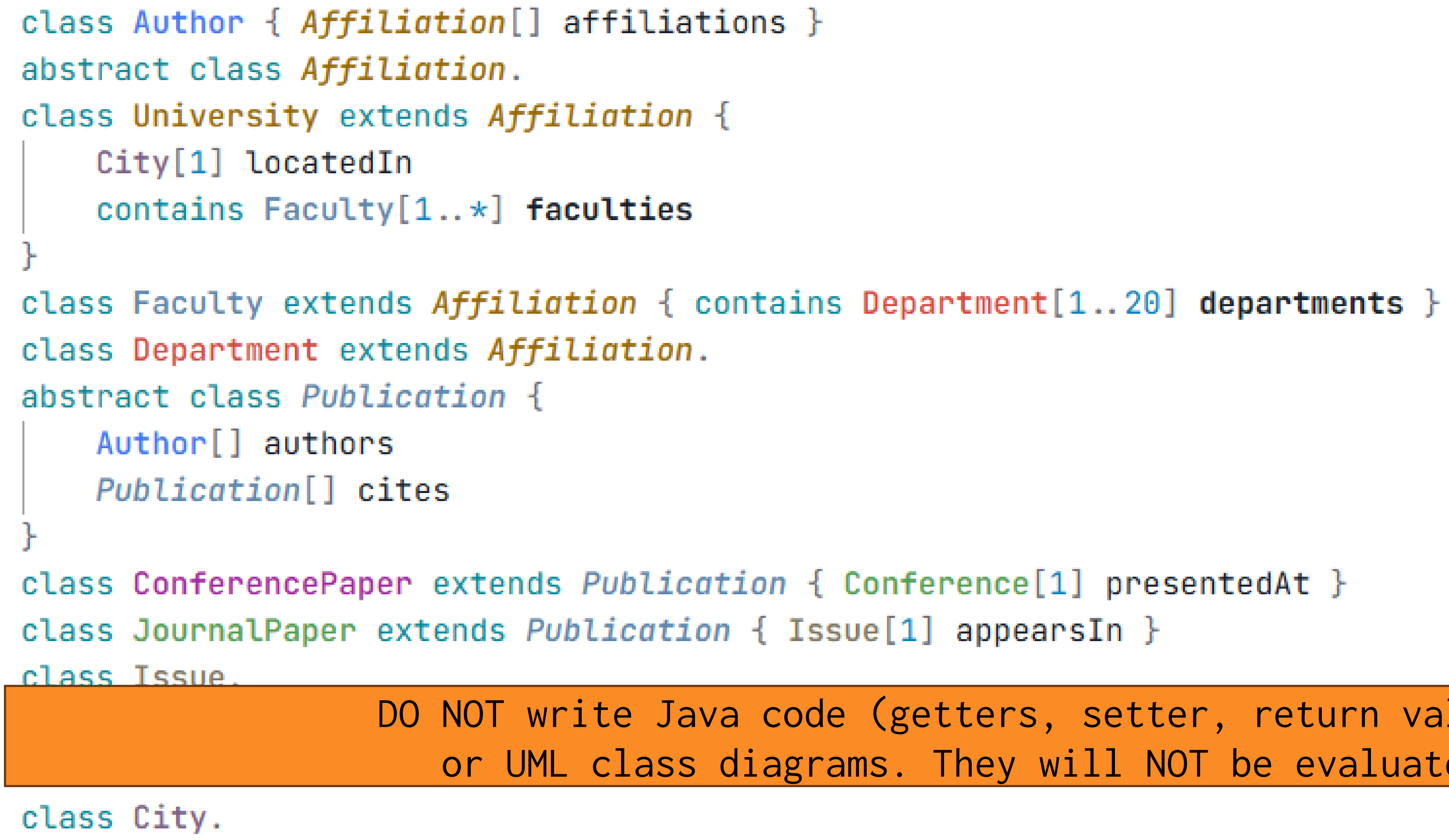
- Lab (practice) session attendance (3 can be missed)
- Accepted homework
  - (must obtain 40% of the possible score in both stages,
  - constitutes 30% of the total score)
- Exam
  - (must obtain 40% of the possible score,
  - constitutes 70% of the total score)
- English + Hungarian
- Final mark is as usual (85%, 70%, 55%, 40%)
- IMSC task: until last exam. IMSC scores → extra scores

Task 1.1: Domain Modeling

In a *bibliographical database,* we store information about *authors* and *publications*. An author may have *affiliations*, which can be either a *university*, a *faculty*, or a *department*. Authors without any affiliation are also permitted. Each university is *located in* a *city* and contains at least 1 faculty, and each faculty contains at least 1, but at most 20 departments. A publication has authors and may *cite* other publications. Publications can be either *conference papers* and journal papers. Conference papers are *presented* at a *conference*, while journal papers *appear* in *issues* of *journal*. Each conference is *held in* a city. Each journal has at least 1 issue.

Create a **metamodel** for this domain using the **Refinery language.**
Use the following names for **concepts:** `Author`, `Affiliation`, `University`, `Faculty`, `Department`, `affiliations`, `faculties`, `departments`, `Publication`, `cites`, `ConferencePaper`, `presentedAt`, `JournalPaper`, `appearsIn`, `Conference`, `heldIn`, `City`, `Issue`, `issues`, `Journal`.

Reference solution



```
class Author { Affiliation[] affiliations }
abstract class Affiliation.
class University extends Affiliation {
    City[1] locatedIn
    contains Faculty[1..*] faculties
}
class Faculty extends Affiliation { contains Department[1..20] departments }
class Department extends Affiliation.
abstract class Publication {
    Author[] authors
    Publication[] cites
}
class ConferencePaper extends Publication { Conference[1] presentedAt }
class JournalPaper extends Publication { Issue[1] appearsIn }
class Issue.
```

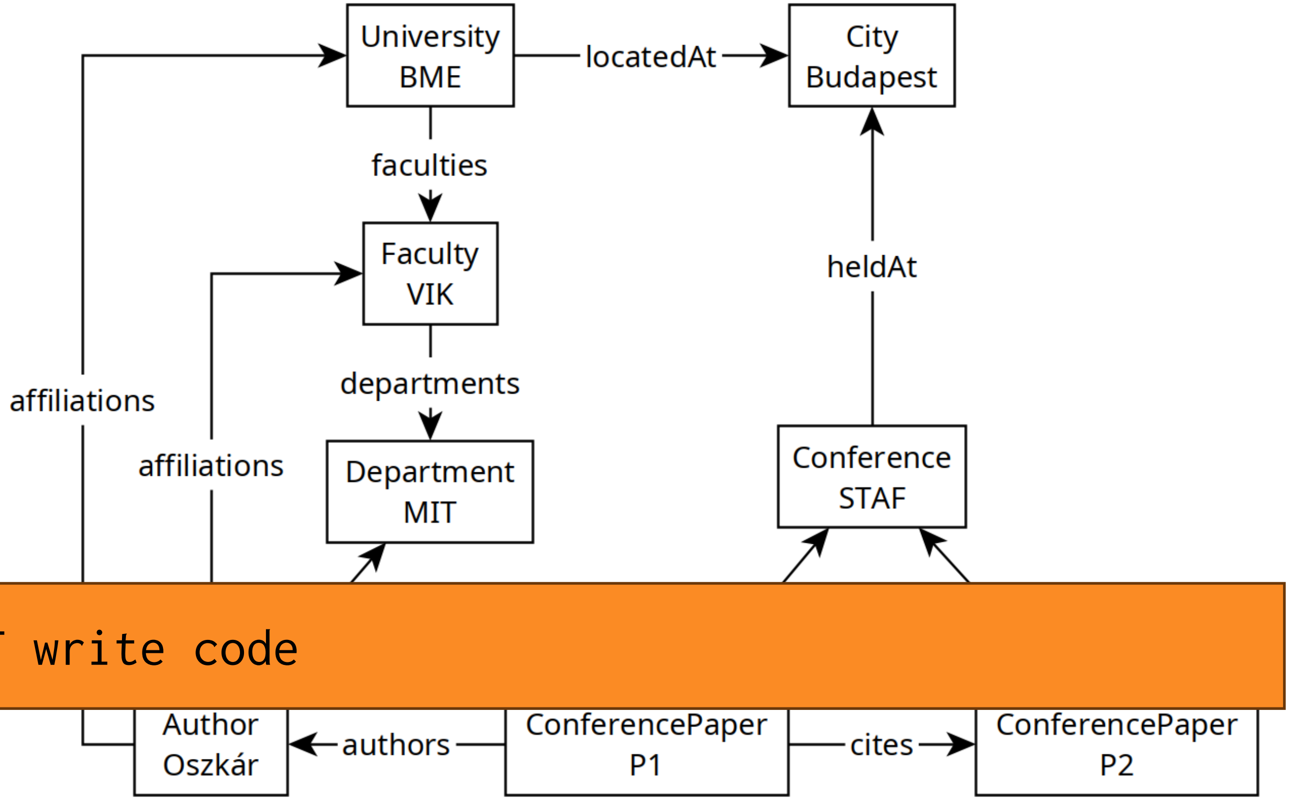
DO NOT write Java code (getters, setter, return values),
or UML class diagrams. They will NOT be evaluated.

```
class City.
```

Task 1.2: Graph Modeling

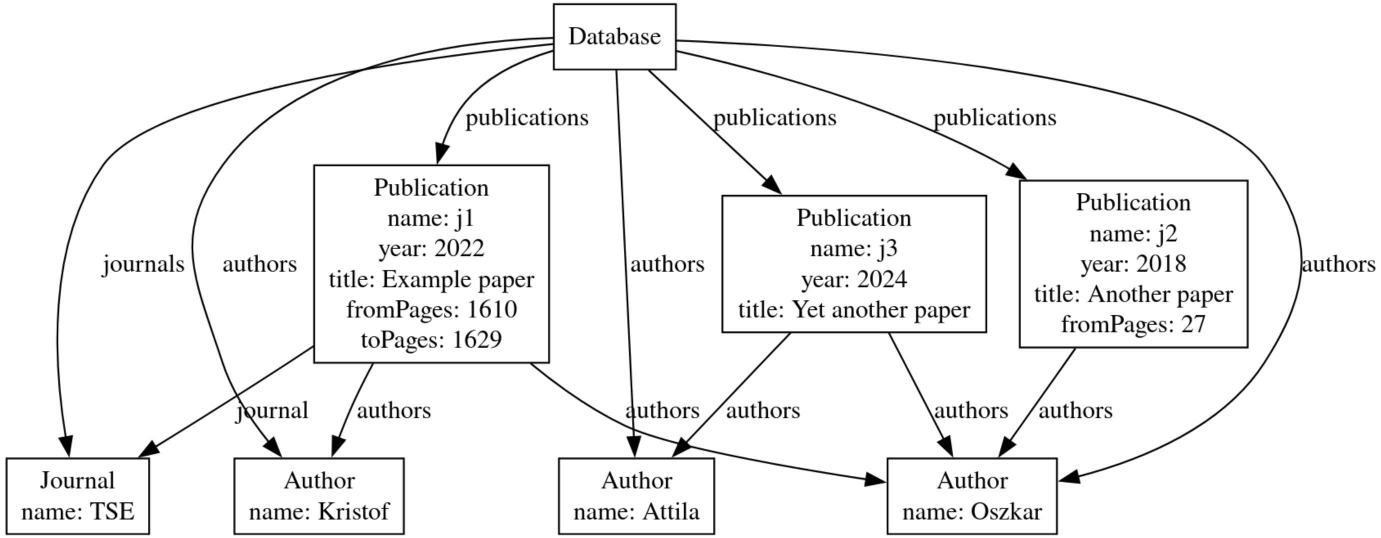
Oszkár works at the *MIT* department of the *VIK* faculty of the *BME* university, which is located in *Budapest*. He is affiliated with all 3 of FTSRG, VIK, and BME. He is an author of a conference paper *P1* that was presented at the *STAF* conference held at Budapest. His paper cites another paper *P2* that was also presented at *STAF*.

Draw the **instance graph** of this scenario using graphical abstract syntax:



DO NOT write code

Task 2.1: Grammar



- Create a grammar for the following language:

Any order

List of authors

List of journals

List of citations

Reference to authors

(Year)

“Title”

Journal

Optional pages

```
author Oszkar
author Kristof
author Attila

journal TSE

[j1] Kristof, Oszkar (2022) "Example paper" in TSE pp 1610-1629
[j2] Oszkar (2018) "Another paper" p 27
[j3] Attila, Oszkar (2024) "Yet another paper"
```

- Write the grammar!

Task 2.1: Grammar solution

Header is given!

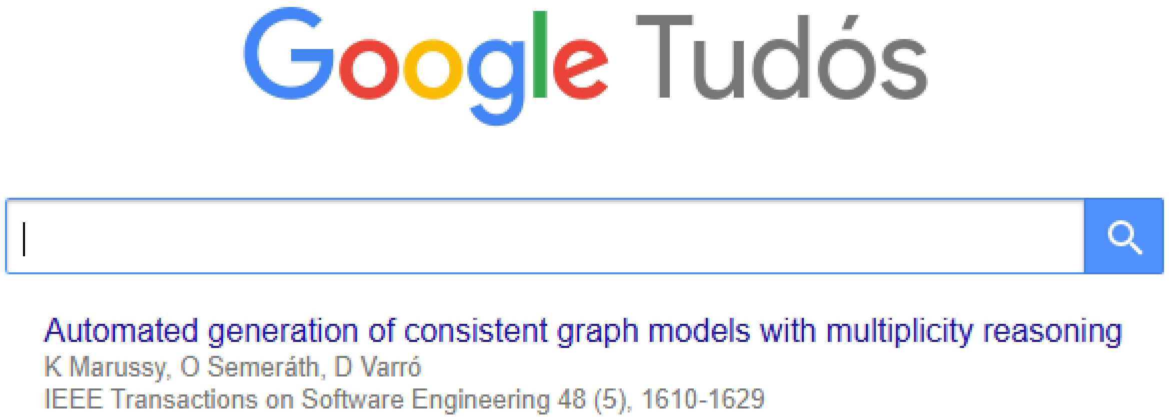
```
grammar Bibliography
entry Database:
(authors+=Author | journals+=Journal | publications+=Publication)*;

Author: "author" name=ID;
Journal: "journal" name=ID;

Publication:
  "[" name=ID "]"
  authors+=[Author] ("," authors+=[Author])*
  "(" year=INT ")" title=STRING
  ("in" journal=[Journal])?
  ("p" fromPages=INT | "pp" fromPages=INT "-" toPages=INT)?;

hidden terminal WS: /\s+/;
terminal ID: /[_a-zA-Z][\w_]*/;
terminal INT returns number: /[0-9]+/;
terminal STRING: /"[^"]*"/;
```

Should not be complex

Terminals are given!

Task 2.2: Code Generation

- Write a code generation that outputs a .bib file!



Automated generation of consistent graph models with multiplicity reasoning
K Marussy, O Semeráth, D Varró
IEEE Transactions on Software Engineering 48 (5), 1610-1629

```
@article{j1,
  title={Automated generation ...},
  author={Kristof and Oszkar and Daniel},
  pages={1610--1629},
  year={2020},
}
```

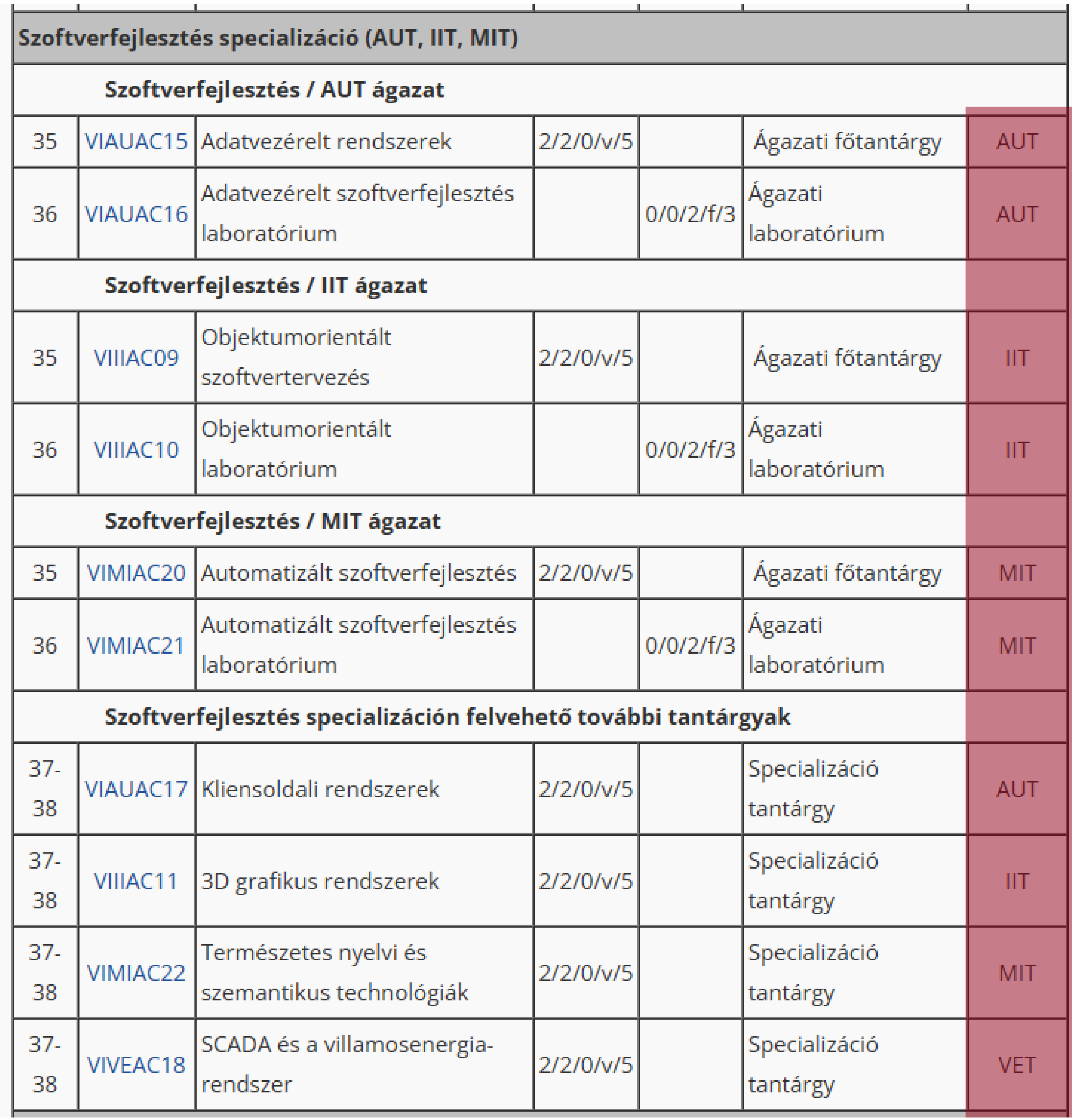
- Write Jinja code that outputs the example output from the example output!

Task 3: 3x short questions, short answers

- *During the performance evaluation of a piece of code, we measure the runtime of the software 30 times. We discover that every time we start the application, the first 2 measurement are extremely slow (more than 100x time).*
  Please, explain the reason of tis phenomenon, and propose a solution to correct the performance evaluation.
- *We would like to evaluate the correctness of an important piece of source code. We try to run a model checker to prove its correctness, but due to the complexity of the code, the model checker does not terminate within hours.*
  Does this mean that the code is correct or incorrect?
- *We are developing a new development tool supported by an AI-based component. Our first prototype provide very good performance, so we do not provide new training set for fine tuning.*
  Do we need to care about any copyright issues when we use this new tool?

Önálló labor

- Szakirány:
  - Választható tárgyak köre
  - Választható tárgyakba beoktatott tanszékek → **Önlab + Szakdolgozat** BMEVI<Tanszék>AL01
- Ágazat:
  - Kötelező választható tárgy
  - Szakirány labor
  - Alapértelmezett tanszék
- **HF:** Konzulens keresése
  - https://tdk.bme.hu/
  - https://diplomaterv.vik.bme.hu
  - emailezés



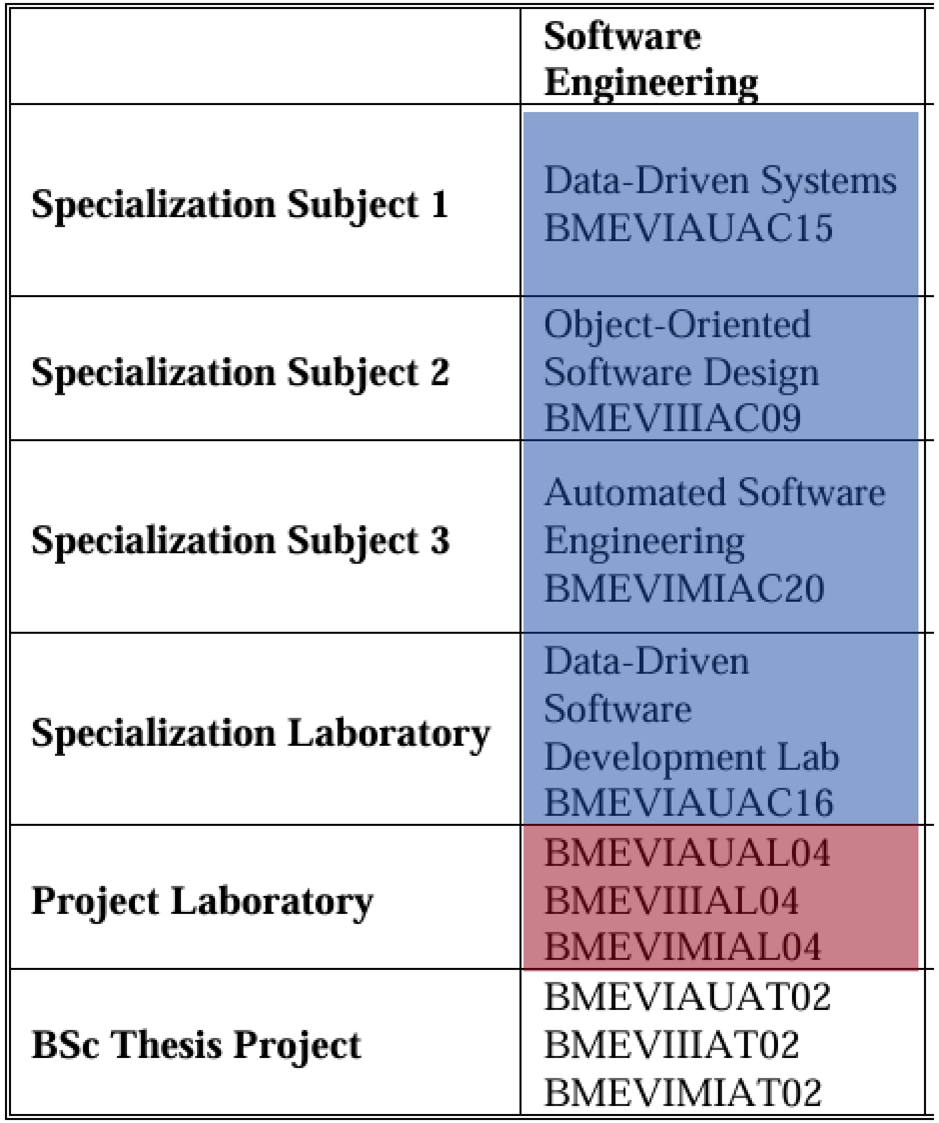
Szoftverfejlesztés specializáció (AUT, IIT, MIT)						
Szoftverfejlesztés / AUT ágazat						
35	VIAUAC15	Adatvezérelt rendszerek	2/2/0/v/5		Ágazati főtantárgy	AUT
36	VIAUAC16	Adatvezérelt szoftverfejlesztés laboratórium		0/0/2/f/3	Ágazati laboratórium	AUT
Szoftverfejlesztés / IIT ágazat						
35	VIIIAC09	Objektumorientált szoftvertervezés	2/2/0/v/5		Ágazati főtantárgy	IIT
36	VIIIAC10	Objektumorientált laboratórium		0/0/2/f/3	Ágazati laboratórium	IIT
Szoftverfejlesztés / MIT ágazat						
35	VIMIAC20	Automatizált szoftverfejlesztés	2/2/0/v/5		Ágazati főtantárgy	MIT
36	VIMIAC21	Automatizált szoftverfejlesztés laboratórium		0/0/2/f/3	Ágazati laboratórium	MIT
Szoftverfejlesztés specializáción felvehető további tantárgyak						
37-38	VIAUAC17	Kliensoldali rendszerek	2/2/0/v/5		Specializáció tantárgy	AUT
37-38	VIIIAC11	3D grafikus rendszerek	2/2/0/v/5		Specializáció tantárgy	IIT
37-38	VIMIAC22	Természetes nyelvi és szemantikus technológiák	2/2/0/v/5		Specializáció tantárgy	MIT
37-38	VIVEAC18	SCADA és a villamosenergia-rendszer	2/2/0/v/5		Specializáció tantárgy	VET

https://www.vik.bme.hu/page/1267/

Project Laboratory

- Specialization:
  - Group of courses you can select
  - Departments that provide supervisors →
  - **Project Laboratory + Thesis**
  - Available departments:
    - AUT (default)
    - IIT
    - MIT (mine)
- **Homework:** look for supervisor
  - https://tdk.bme.hu/
  - https://diplomaterv.vik.bme.hu/en
  - Emailing, asking around
- **Otherwise:** random topic

https://vik.bme.hu/document/5728/original/BSC_CE_20230703.pdf



	Software Engineering
Specialization Subject 1	Data-Driven Systems BMEVIAUAC15
Specialization Subject 2	Object-Oriented Software Design BMEVIIIAC09
Specialization Subject 3	Automated Software Engineering BMEVIMIAC20
Specialization Laboratory	Data-Driven Software Development Lab BMEVIAUAC16
Project Laboratory	BMEVIAUAL04 BMEVIIIAL04 BMEVIMIAL04
BSc Thesis Project	BMEVIAUAT02 BMEVIIIAT02 BMEVIMIAT02