# INF200_H21_Ju07

June 7, 2022

# 1 INF200 Lecture No Ju07

### 1.0.1 Hans Ekkehard Plesser / NMBU

### 1.0.2 8 June 2022

## 1.1 Today's topics

- Schedule for remainder of block
- Class diagrams
- Profiling and optimization
- Exam information
  - Exam schedule
  - How to submit your code
  - How to submit your presentation material

---

## 1.2 Schedule for remainder of block

- Today & Tomorrow
  - 09.15 Morning meeting
  - 14.30 Afternoon meeting
- Friday 10 June
  - 14.30 Afternoon meeting
- Monday 13 June
  - 09.15 Morning meeting
  - 14.30 Afternoon meeting
- Tuesday 14 June
  - 09.15 Morning meeting
- Wednesday 15 June
  - 09.15 Morning meeting
  - **15.00 Deadline for delivery**
- Lectures on model dynamics and programming in C++ are available via Panopto.
  - INF200_H20_J08_ModelDynamics
  - INF200_H20_C++_Part1
  - INF200_H20_C++_Part2
  - Study them on your own, **but do not ignore them!**
- Thursday & Friday 16/17 June
  - Individual work on your presentations

    - *No* mandatory attendance
    - TAs will be available for some time for questions
        - Precise times will be posted later
- Friday 17 June
    - **Presentation material deadline 15.00 CEST**
- Monday/Tuesday 20/21 June
    - Exam

---

## 1.3 Class diagrams

- PyCharm allows you to draw class diagrams
- Right-click on your package, then `Diagrams > Show diagram`
- See also Unified Modeling Languange (UML)

---

## 1.4 Profiling and optimization

1. See also L12 from the fall term
2. Profile to understand where you program uses time
    - **Disable all graphics and file output for profiling**
    - Profile in PyCharm using `Profile` from the `Run` menu
        - See call statistics and call graph
        - Graph can also be stored as figure
        - If the graph never appears, update PyCharm to newest version (at least 2021.3.1)
        - When the graph appears but looks weird
            - click this icon < to draw it properly
            - right click on the graph background and remove the tick on `Appearance > Show Grid` to remove the grid behind the graph
3. Optimize the parts that require most time first
    - See if you can find "low hanging fruits": individual functions taking a lot of time
4. How to optimize
    - Reduce number of function calls required
    - Make functions run faster
5. Techniques
    - Lazy evaluation
        - If a computation is costly, perform it only when really necessary
        - Mark value as invalid if changes occur that will require recomputation at some point
        - When a value is marked invalid when requested, recompute and store
    - Implementation of lazy evaluation
        - Use *properties* in Python
    - Just in time compilation using Numba
        - See discussion in lectures in the fall term
        - Use only for compact functions or methods performing mathematical operations
    - Coding critical parts with Cython (or even in C++, but that is a whole new story …)

---

## 1.5 Exam schedule

See Canvas: https://nmbu.instructure.com/courses/7165/pages/june-block-exam-schedule

## 1.6 Submission of material

### 1.6.1 Project Code

- **Deadline: Wednesday, 15 June 2022, 18.00 CEST**
- Submit your code as follows, working in your team repository (see also video INF200_H20_GitTags on Panopto):
  1. Commit your final changes
  2. Make sure all your code is in your **main branch** (or *master branch*)
  3. Push everything to you GitLab repository
  4. On GitLab, got to the History for your repo and choose the commit with your final version and use `Options > Tag` to add the tag `BiosimSubmission` (no spaces)
  5. **Confirm that tag is visible on GitLab!**
- The tagged commit must be dated no later than 15 June 2022, 15.00 CEST.
- Test creating tags **today**!

### 1.6.2 Presentation files

- All presentation files must be handed in, and will be available on the examiners machine during the presentation, this is to save time on setup between students.
- **Deadline: Friday, 17 June, 15.00 CEST**
- Submit your material (one PDF plus one mp4 or GIF) as follows, working in your **team repository**:
  1. At the top level in the repository, create a folder `Exam`
  2. Put your files into the `Exam` folder, on branch `main`
     - If you do not have an animation, add a file `no_animation.txt`. The file can be empty.
       * This is just for my information, so I don't need to double-check with you.
  3. Commit and push to GitLab
  4. Add a tag `INF200Exam` to the final commit of the material
  5. Make sure material and tag are visible on GitLab
- Files and tag must be in place by the deadline.
- I will contact you by mail if there is any issues getting hold of your pdf/mp4/gif so **please check your mail during Saturday and Sunday!**

[ ]:

```

```