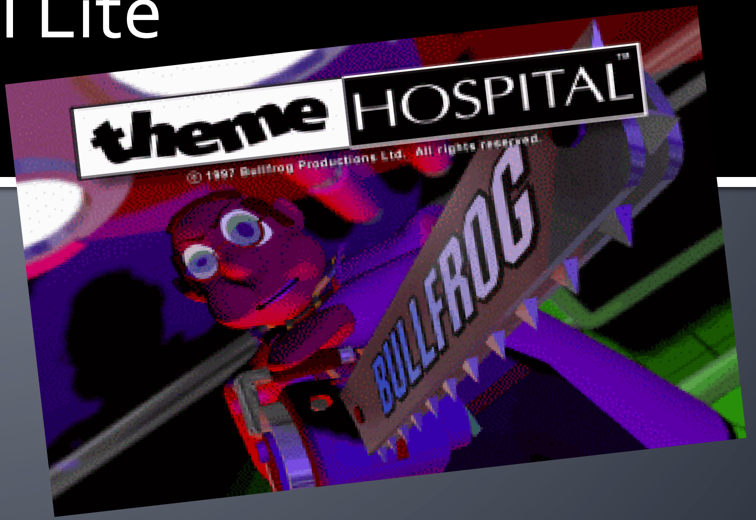
Faculty of Mathematics and Physics
Charles University in Prague
14th April 2016

C# Made Easy!

Programming II

Workshop 07 –Theme Hospital Lite
Part 3 – Code Architecture



Lab 07

Outline

1. Test
2. Task Architecture Brainstorming
3. Full Assignment Specification
    - The Simulation

Test 07

Test

Find the test here (no-ads):

`https://goo.gl/IOSM0B`

`O vs. 0, i vs. l vs. 1`

Permanent link:

https://docs.google.com/forms/d/1dvJFH2wEaYNUrdvpV_DXGhYMU33nkZy7HWro-bIi9-Y/viewform

Time for the test:

15 min

Topic

Let's talk about architecture

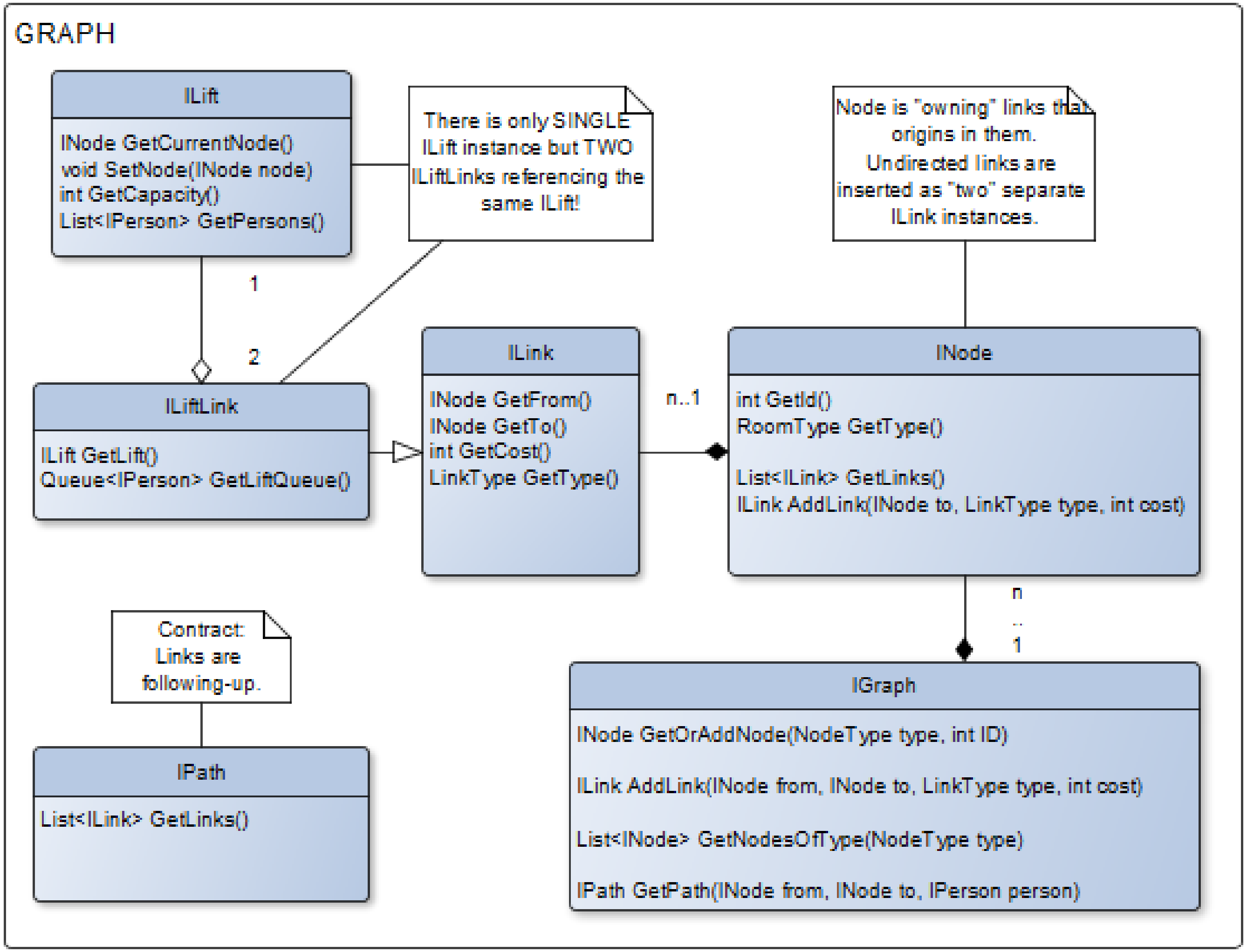
Architectural Requirements

What functionalities should be modelled?

- Let’s write them down using plain language:
- Points of focus:
  - How to model LIFTs?
    - **https://goo.gl/hGA4kA**
    - https://docs.google.com/forms/d/1cNDD3Gt3ngbYNYh-ouJM2-wiqeEEVKxKTL6d91-Kjig/viewform
  - How to model PERSONs?
    - **https://goo.gl/4lAEeC**
    - https://docs.google.com/forms/d/1Q291uruXX4BeBJi8u5_BLFC_qRoCbio9lo-BBwCOmyM/viewform
  - `O vs. 0, i vs. l vs. 1`

Architectural Requirements

LIFTs (Graph)



GRAPH

ILift

INode GetCurrentNode()
void SetNode(INode node)
int GetCapacity()
List<IPerson> GetPersons()

There is only SINGLE ILift instance but TWO ILiftLinks referencing the same ILift!

Node is "owning" links that origins in them.
Undirected links are inserted as "two" separate ILink instances.

ILiftLink

ILift GetLift()
Queue<IPerson> GetLiftQueue()

ILink

INode GetFrom()
INode GetTo()
int GetCost()
LinkType GetType()

INode

int GetId()
RoomType GetType()

List<ILink> GetLinks()
ILink AddLink(INode to, LinkType type, int cost)

Contract:
Links are following-up.

IPath

List<ILink> GetLinks()

IGraph

INode GetOrAddNode(NodeType type, int ID)

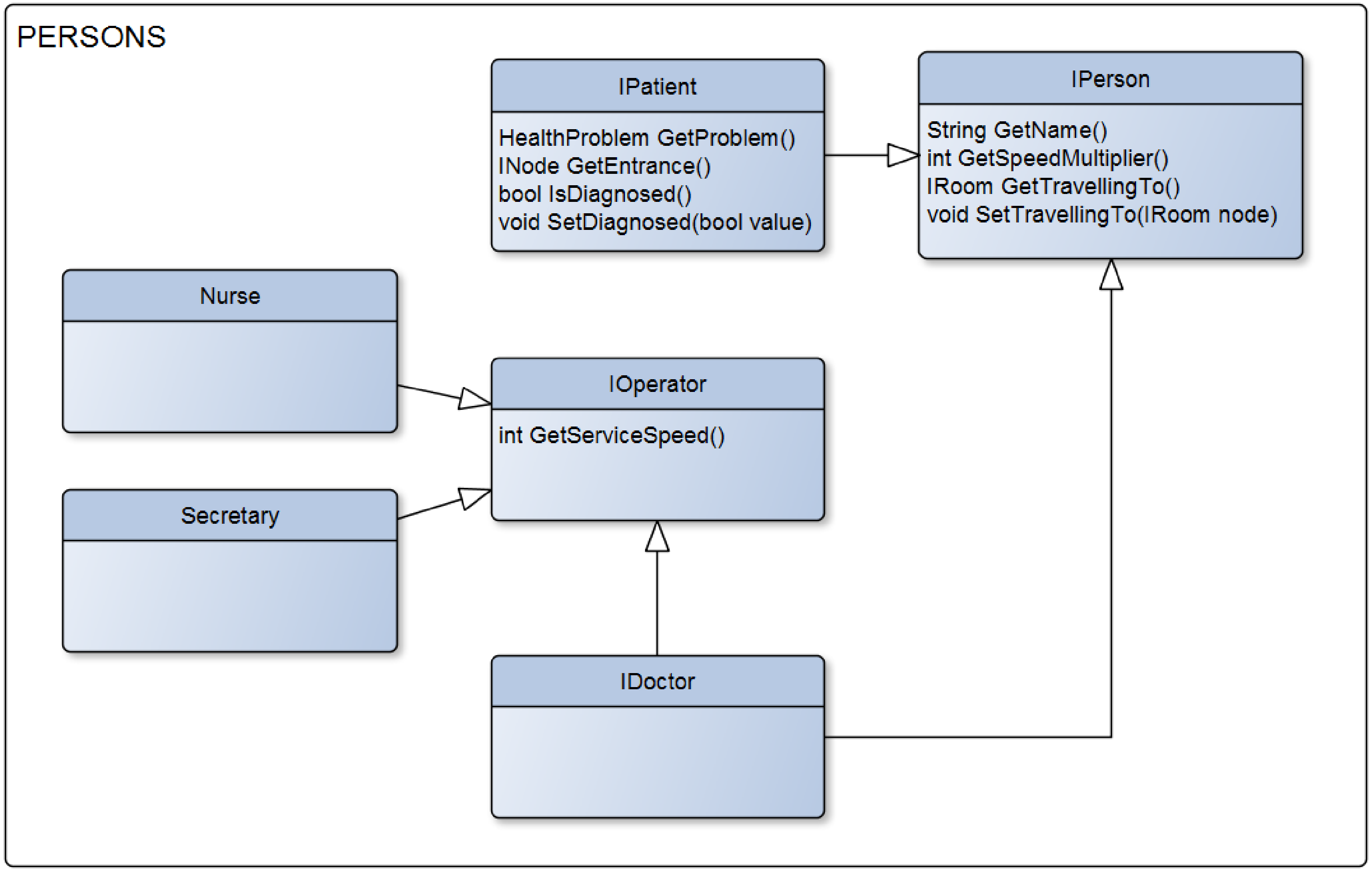
ILink AddLink(INode from, INode to, LinkType type, int cost)

List<INode> GetNodesOfType(NodeType type)

IPath GetPath(INode from, INode to, IPerson person)

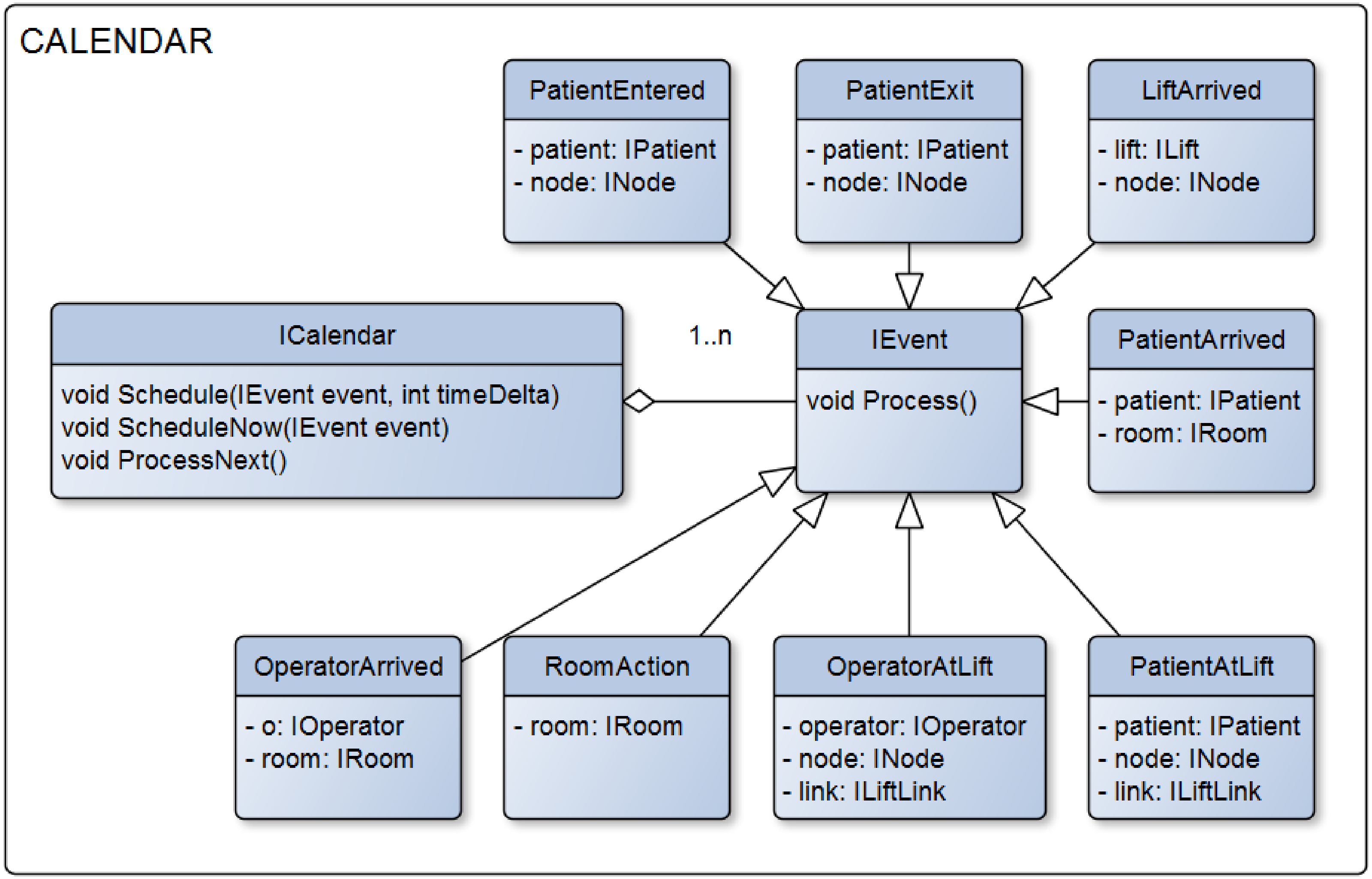
Architectural Requirements

Persons



Architectural Requirements

LIFTs (Events)



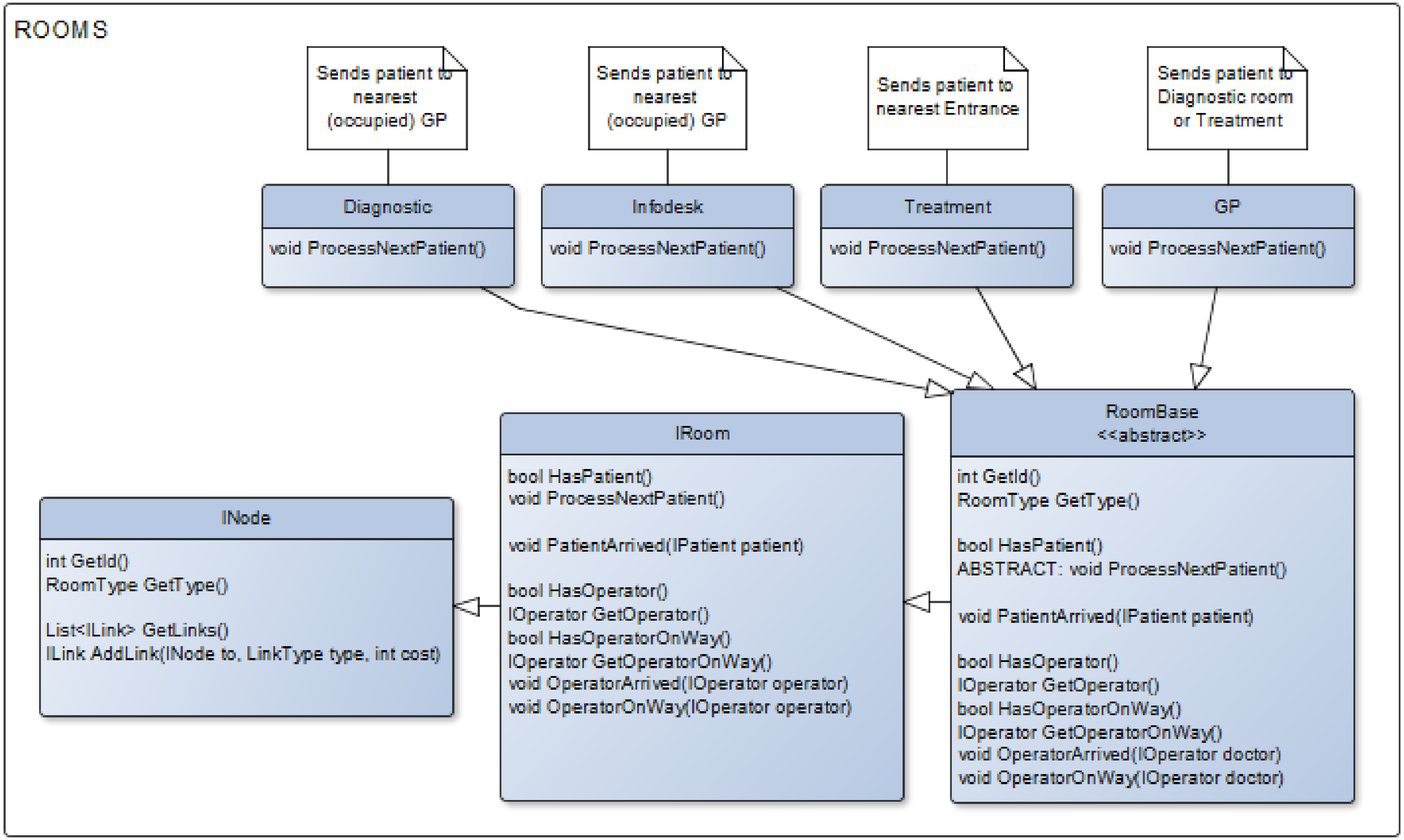
Architectural Requirements

What functionalities should be modelled?

- Points of focus:
  - How to model ROOMs?
    - https://goo.gl/hS0jb4
    - https://docs.google.com/forms/d/1PtB3RYJTz3zLJCHMi7IspI8MZY9AGks-ZDSOCWRUeFI/viewform
  - How to prepare for “experimentation with simulation”?
    - https://goo.gl/t70hsS
    - https://docs.google.com/forms/d/1vYmPXVzuW1_s1GoYQdiRmozosHFAQj53lXvvTLBc_Wo/viewform
  - `O vs. 0, i vs. l vs. 1`

Architectural Requirements

ROOMs (Events)



ROOMS

Sends patient to nearest (occupied) GP

Sends patient to nearest (occupied) GP

Sends patient to nearest Entrance

Sends patient to Diagnostic room or Treatment

Diagnostic
void ProcessNextPatient()

Infodesk
void ProcessNextPatient()

Treatment
void ProcessNextPatient()

GP
void ProcessNextPatient()

RoomBase
<<abstract>>

int GetId()
RoomType GetType()

bool HasPatient()
ABSTRACT: void ProcessNextPatient()

void PatientArrived(IPatient patient)

bool HasOperator()
IOperator GetOperator()
bool HasOperatorOnWay()
IOperator GetOperatorOnWay()
void OperatorArrived(IOperator doctor)
void OperatorOnWay(IOperator doctor)

IRoom

bool HasPatient()
void ProcessNextPatient()

void PatientArrived(IPatient patient)

bool HasOperator()
IOperator GetOperator()
bool HasOperatorOnWay()
IOperator GetOperatorOnWay()
void OperatorArrived(IOperator operator)
void OperatorOnWay(IOperator operator)

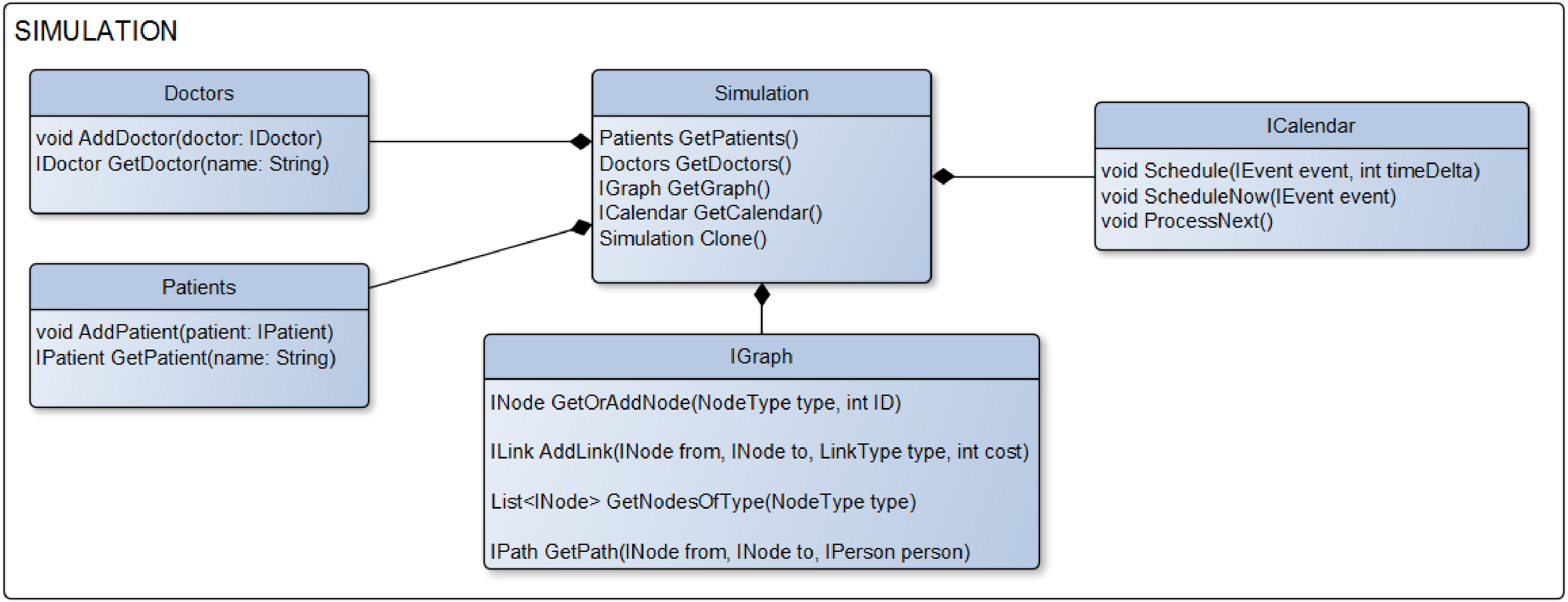
INode

int GetId()
RoomType GetType()

List<ILink> GetLinks()
ILink AddLink(INode to, LinkType type, int cost)

Architectural Requirements

Simulation



SIMULATION

Doctors

void AddDoctor(doctor: IDoctor)
IDoctor GetDoctor(name: String)

Patients

void AddPatient(patient: IPatient)
IPatient GetPatient(name: String)

Simulation

Patients GetPatients()
Doctors GetDoctors()
IGraph GetGraph()
ICalendar GetCalendar()
Simulation Clone()

ICalendar

void Schedule(IEvent event, int timeDelta)
void ScheduleNow(IEvent event)
void ProcessNext()

IGraph

INode GetOrAddNode(NodeType type, int ID)

ILink AddLink(INode from, INode to, LinkType type, int cost)

List<INode> GetNodesOfType(NodeType type)

IPath GetPath(INode from, INode to, IPerson person)

Architectural Requirements

RESPONSEs

- Brainstorming results
  - How to model LIFTs?
    - https://docs.google.com/spreadsheets/d/1i1jUI8jiYaykvamJ5bWuVbbxgygRExRDzw6czKSEzQg/edit?usp=sharing
  - How to model PERSONs?
    - https://docs.google.com/spreadsheets/d/1fyxauTq1g__NOpH_bMUcRiHNgN9CQs5sOR22RFUYFuo/edit?usp=sharing
  - How to model ROOMs?
    - https://docs.google.com/spreadsheets/d/1EGYrAnMLhz4_dfPJjt7-kf3JT8DF6r9rgap2LZFi1Gs/edit?usp=sharing
  - How to prepare for “experimentation with simulation”?
    - https://docs.google.com/spreadsheets/d/1T7o2bCHi8LBOobc_ot4y6HE4KTb3Rb8YoMovcUMu8Uo/edit?usp=sharing

Theme Hospital Lite

FULL ASSIGNMENT SPECIFICATION

1. Introduction
2. Graph & Rooms
3. Patients
4. Staff (Doctors, Nurses, Secretaries)
5. Navigation
6. The Simulation
7. Input specification
8. Output specification
9. Task summary

Theme Hospital Lite

1. Introduction

- Rooms / Places
  - Info Desk, GP, EEG, Sono, X-Ray, Psycho, Treatment
- Staff
  - Secretaries, Doctors, Nurses
- Various hospitals
  - Different topologies between rooms
  - Walking / Lift-riding
- Source of income
  - Patients
    - Various age (speed of walking)
    - Various health problems (need special type of diagnoses)
- Our objective
  - Maximize the profits!

=> Minimize the number of doctors that you need for a given day

Theme Hospital Lite

2. Graph & Rooms

- Hospital == Oriented graph with costs at edges
- Nodes are “Rooms/Places”
  - `ENTRANCE`
  - `INFODESK`
  - `GP`
  - `EEG`
  - `SONO`
  - `XRAY`
  - `PSYCHO`
  - `TREATMENT`
  - `NODE`
- There can be multiple places of a given room type!

Theme Hospital Lite

2. Graph & Rooms

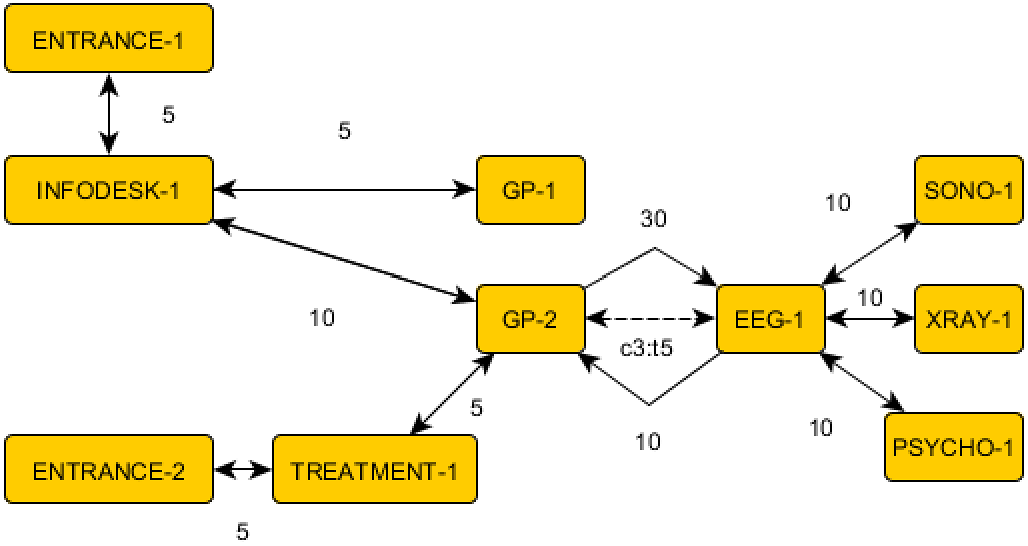
- Hospital == Oriented graph with costs at edges

- Edges are of two types
  1. Corridors/Stairs
     - Oriented edges
     - Has a base cost in “seconds” (positive integer number)
     - Cost is modified by person’s `WalkingMultiplier` !
  2. Lifts
     - Non-oriented edges (rides both ways)
     - Always runs between two nodes only
     - Has a maximum capacity
     - Has a base cost in “seconds” (positive integer number)

Theme Hospital Lite

2. Graph & Rooms

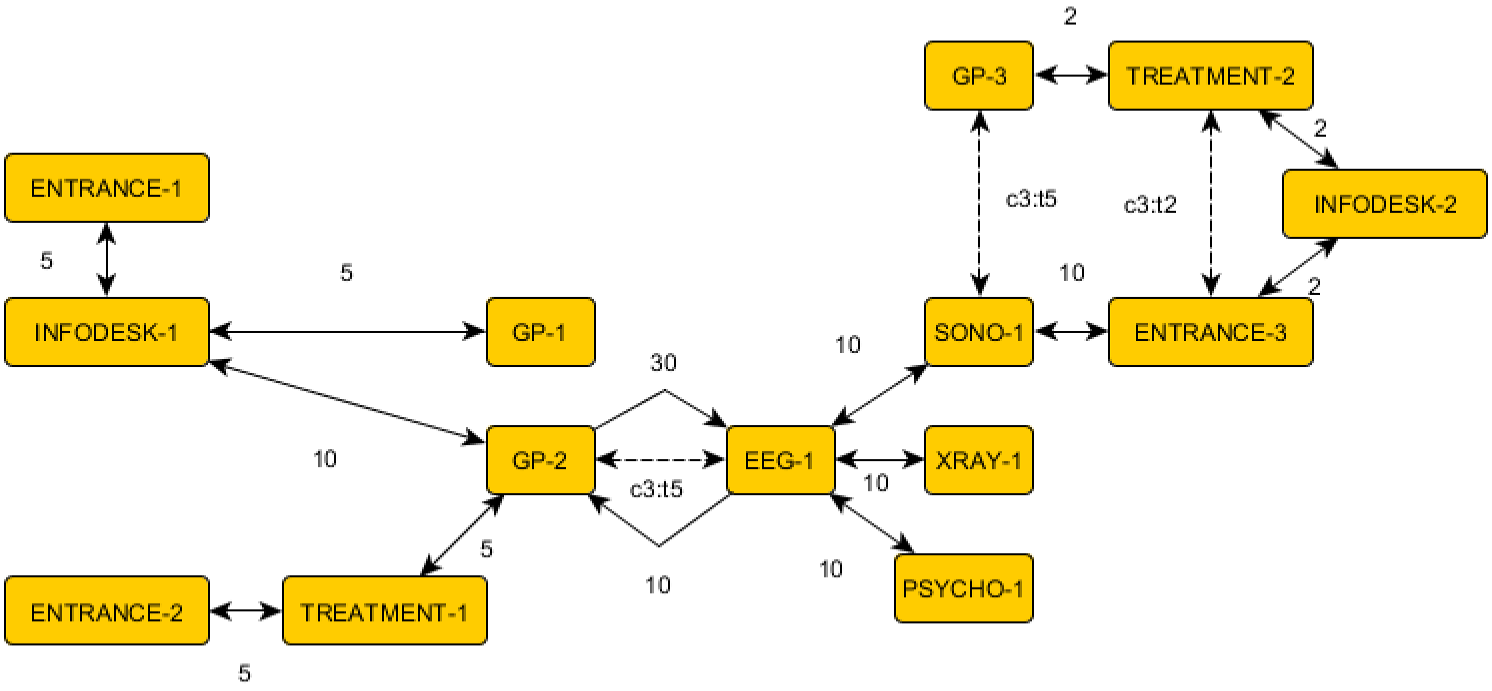
- Example 1



Theme Hospital Lite

2. Graph & Rooms

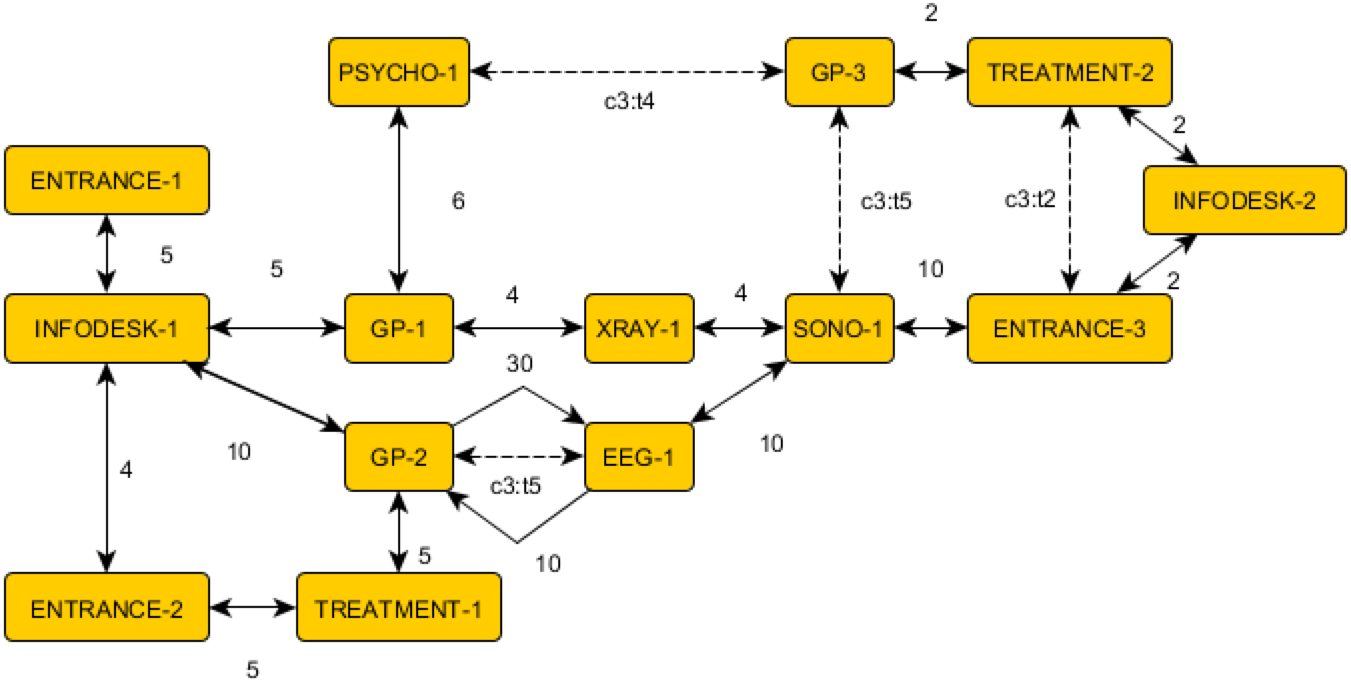
- Example 2



Theme Hospital Lite

2. Graph & Rooms

- Example 3



Theme Hospital Lite

2. Graph & Rooms

- The link’s cost is always in “seconds”
- So if lift’s cost is “10” it means it travels the link in 10 seconds.
- If person with `speedMultiplier` 2 is travelling through “walk” link of cost 20, then it means it will take them 2*20=40 seconds
- If person with `speedMultiplier` 2 is travelling through “lift” link of cost 20, then it means it will take them 20 seconds (`speedMultiplier` does not affect lift-link traversal)

Assignment 5

3. Patients

- Each patient has own health problem / disease
- Each patient enters a hospital from a pre-set entrance (you cannot choose, which entrance the patient use to get into the hospital)
- The type of disease is treated as if "known in advance", it is a part of the input

- List of health problems / diseases:
  - CARDIAC
  - PNEUMONIA
  - HIP-PAIN
  - NEUROTIC

Assignment 5

3. Patients

- The patient’s route through the hospital is “fixed” in a sense that they always travel through a “fixed” types of rooms.

  ENTRANCE (as defined)
  => INFODESK
  => GP
  => specific diagnoses room determined by patient’s health problem
  => GP
  => TREATMENT
  => ENTRANCE (used as exit)

- Mapping between Health problems and Specific diagnostics:
  - CARDIAC -> EEG
  - PNEUMONIA -> XRAY
  - HIP-PAIN -> SONO
  - NEUROTIC -> PSYCHO

- E.g. when the patient has “PENUMONIA” it will travel to “XRAY” room

Assignment 5

4. Staff

- Every room (except node) requires some staff in order to be "functional"; each room may be occupied by single "operator" only
- There are 3 types of staff persons:
  - Doctors – GPs, specific diagnose rooms
  - Nurses - TREATMENTs
  - Secretaries – INFODESKs
- Only doctors are going to be simulated in detail, they will have to travel between their rooms
- Each doctor has own `speedMultiplier` specified
- Nurses and secretaries will be fixed in a concrete room for the whole day and every such room will have one

Assignment 5

4. Staff

- Every staff member will have `serviceSpeed`
- Person's `serviceSpeed` will determine how long it takes to "process" some patient
- As nurses and secretaries won't travel between TREATMENTs resp. INFODESKs, the `serviceSpeed` will be associated with concrete room
- The same does not apply to GPs and other diagnostic rooms as the time required to "process" a patient there will be determined by a doctor who will work in the room

Assignment 5

5. Navigation

- Only Patients and Doctors needs to be navigated around the hospital
- **Patients** always travels to the “**nearest**” room of some type
- **Doctors** travel between rooms **only if some condition is met** (typically when they are required somewhere and they are in the room that noone is using / is about to use)

Assignment 5

5. Navigation

- Patient always travels (gets send) to the "nearest" room of some type
- Patient's route always looks like this

  Patient enters the hospital via pre-set ENTRANCE
  -> Travels to the nearest INFODESK
  [waits there to be processed / gets processed]
  -> Travels to the nearest GP (regardless doctor's presence)
  [waits there to be processed / gets diagnosed]
  -> Travels to the nearest diagnose room for patient's health problem (regardless doctor's presence)
  [waits there to be processed / gets diagnosed]
  -> Travels to the nearest GP (regardless doctor's presence)
  [waits there to be proessed / receive final diagnose]
  -> Travels to the nearest TREATMENT
  [waits there to be treated / receive treatments]
  -> Exits via the nearest ENTRANCE

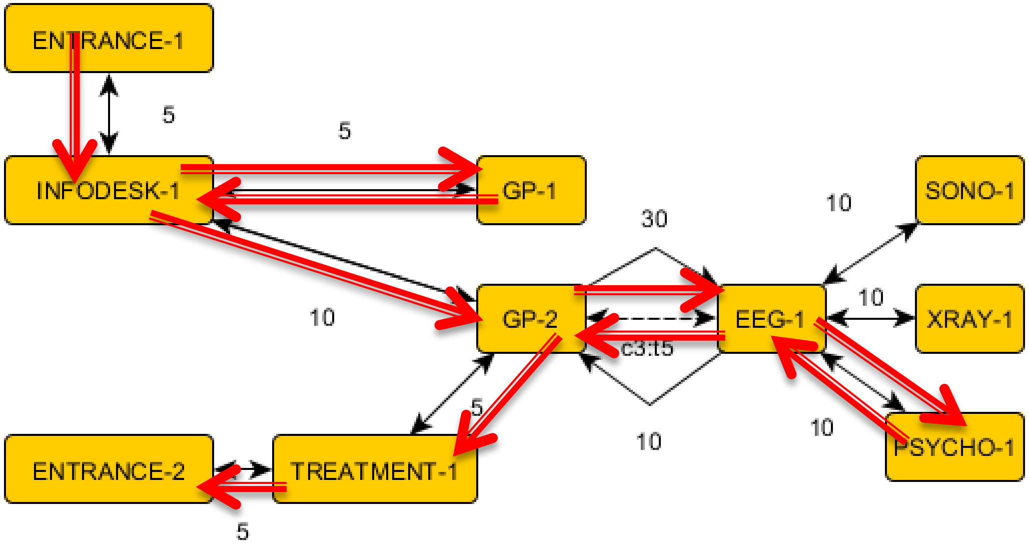
Mapping between health problems and specialized diagnostic rooms:

- CARDIAC -> EEG
- PNEUMONIA -> XRAY
- HIP-PAIN -> SONO
- NEUROTIC -> PSYCHO

Theme Hospital Lite

5. Navigation

- Example for a NEUROTIC Patient who is entering the hospital via ENTRANCE-1 and `speedMultiplier` 2



- Total path cost is:

2*5+2*5+2*5+2*10+5+2*10+2*10+5+2*5+2*5 = 120

Theme Hospital Lite

5. Navigation

- Beware of lifts! You must simulate them!
- Every lift has some `capacity` and operates between two nodes
- You have to track where the lift is
- And a person (patient or doctor) will have to wait for the lift if not available right now
- Note that a person will only wait if the lift's queue is not bigger than the lift's `capacity`

Theme Hospital Lite

6. The Simulation

- Initial state
  - Lifts are at their starting node as specified by the input
  - All nurses and secretaries are at their TREATMENTs, resp. INFODESKs
  - All doctors you wish to “use” in the current simulation run must be placed into some room (mind the requirement that there can be single “operator” within the room only)
  - The simulation begins at 8:00:00 (hospital opens at 8:00 AM)

Theme Hospital Lite

6. The Simulation

- Patients
  - Each task defines a set of patients that will visit the hospital during the day; each patient is associated with an arrival `time` and `entrance`, which they enter the hospital
  - They navigate the hospital as specified within (5)
  - They never die due to their health problem

Theme Hospital Lite

6. The Simulation

- INFODESK / TREATMENT
  - Each info desk / treatment has a „service speed associated“, that is, how much time it needs to “tell the patient how to navigate around the hospital”, resp. “cure the patient”
  - This speed is fixed
  - There can be **any number of patients** waiting in the **queue** of an infodesk / treatment

Theme Hospital Lite

6. The Simulation

- GPs / Specific diagnose rooms
  - Similar to INFODESK/TREATMENT, but this time, the speed of service is determined by the doctor who is in the room
  - There can be **any number of patients** waiting in the **queue** of this room as well
  - Doctor will never leaves his/her room while there are patients in the **queue or patients** that are **travelling** to the room right now

Theme Hospital Lite

6. The Simulation

EVENT

- **Person arrives to the node which they need to use a "lift" from**
  - Following cases may occur:

1. Lift is there => Person will immediately use it
2. Lift is not there & Waiting queue is not full & Lift is not riding => Lift gets "called" and starts to travel to the node the person is at, the person will wait for the lift to arrive
3. Lift is not there & Is riding & Waiting queue (of lift capacity length) is not full => Person will wait for the lift to arrive
4. Lift is not there & Waiting queue is full => Person will take detour

Theme Hospital Lite

6. The Simulation

EVENT

- **INFODESK / TREATMENT is not working and has a patient in a queue**
  - Next patient gets “processed” at the speed of the room
  - After it gets “processed” it is decided where it will navigate next (the nearest room) and the patient begins to navigate to the next room

Theme Hospital Lite

6. The Simulation

EVENT

- **GP / Diagnose room has a doctor, who is not working and there is a patient in a queue**
  - Next patient gets “processed” at the speed of the doctor
  - After it gets “processed” it is decided where it will navigate next (the nearest room) and the patient begins to navigate to the next room

Theme Hospital Lite

6. The Simulation

EVENT

- **Doctor finished the last patient within the queue of his/her GP room and there is no patient on their way to doctor's room**
  - Two cases may arise
    1. There is no other room that has a patient trying to "use" or navigating to in order to "use" it => doctor stays in his/her current GP room
    2. There is such a room and

       2.1 There is a doctor who is navigating there => doctor ignores it and stays

       2.2 There is no doctor travelling there =>

       2.2.1 And this doctor is the nearest (`speedMultiplicator` applies, state of lifts are ignored) one who can leave his/her GP => s/he will travel there

       2.2.2 Is not the nearest one => stays in his/her current GP room
    3. There is more than one such a room

       3.1 Same as "2", rooms are processed in the same order as the need arose there

       I.e., there will be a queue of "rooms" that require a doctor and you will assign doctors to them in that order.

Theme Hospital Lite

6. The Simulation

EVENT

- **Doctor finished the last patient within the queue of his/her <u>Specific diagnostic</u> room and there is no patient on their way to doctor's room**
  - Whenever there is no queue, two cases may arise
    1. There is no other room that has a patient trying to "use" or navigating to in order to "use" it => doctor goes to the nearest unoccupied GP
    2. There is such a room and

       2.1 There is a doctor who is navigating there => doctor ignores it and stays

       2.2 There is no doctor travelling there =>

       2.2.1 And this doctor is the nearest one who can leave his/her room => s/he will travel there

       2.2.2 Is not the nearest one => stays in his/her current room
    3. There is more than one such a room

       3.1 Same as "2", rooms are processed in the same order as the need arose there

       I.e., there will be a queue of "rooms" that require a doctor and you will assign doctors to them in that order.

Theme Hospital Lite

6. The Simulation

EVENT

- **Patient starts to navigate to the unoccupied room**

1. And there is a doctor who is neither working nor has a patient on route

   => Nearest such a doctor will start to travel to this room

2. And there is no free doctor

   => Room is added to the queue of rooms that needs a doctor

Theme Hospital Lite

7. Input Specification

- Now you will have to simulate LIFTs!
- This means that you have to know where lift “begins”

Lift link: `[ <lift-start-left-link> | <lift-start-right-link> ]`

```
lift-start-left-link: ‘L<--(lift:c’ <capacity>
  ‘:t’<cost> ‘)-->’

lift-start-right-link: ‘<--(lift:c’ <capacity>
  ‘:t’<cost> ‘)-->L’
```

Theme Hospital Lite

7. Input Specification

INPUT: <int> ‘\n’ [<node> ‘ ’ <link> ‘ ’ <node> ‘\n’]+ <int> ‘\n’ [<patient> ‘\n’]+ <int> ‘\n’ <int> [<infodesk/treatment> ‘\n’]+ ‘\n’ <int> [<doctor> ‘\n’]+ ‘\n’

<node>: <node-type> ‘-’ <id>

<node-type>: [‘ENTRANCE’ | ‘INFODESK’ | ‘GP’ | ‘EEG’ | ‘SONO’ | ‘XRAY’ | ‘PSYCHO’ | ‘TREATMENT’ | ‘NODE’]

<id>: <int>

<int>: [1-9][0-9]{0,1}

<link>: [<walk-link> | <lift-link>]

<walk-link>: [<non-oriented-walk-link> | <oriented-walk-link>]

<non-oriented-walk-link>: ‘<--(walk:’ <int> ‘)-->’

<oriented-walk-link>: ‘--(walk:’ <cost> ‘)-->’

<lift-link>: [<lift-starts-left-link> | <lift-starts-right-link>]

<lift-starts-left-link>: ‘L<--(lift:c’ <capacity> ‘:t’<cost> ‘)-->’

<lift-starts-right-link>: ‘<--(lift:c’ <capacity> ‘:t’<cost> ‘)-->L’

<cost>: <int>

<capacity>: <int>

Theme Hospital Lite

7. Input Specification

INPUT: <int> ‘\n’ [<node> ‘ ’ <link> ‘ ’ <node> ‘\n’]+ <int> ‘\n’ [<patient> ‘\n’]+ <int> ‘\n’ <int> ‘\n’ [<infodesk/treatment> ‘\n’]+ ‘\n’ <int> ‘\n’ [<doctor> ‘\n’]+ ‘\n’

<patient>: <name> ‘:’ <speed-multiplier> ‘:’ <health-problem> ‘:’ <node> ‘:’ <time>

<name>: [A-Z][a-zA-Z]+

<speed-multiplier>: <int>

<health-problem>: [‘CARDIAC’ | ‘PNEUMONIA’ | ‘HIP-PAIN’ | ‘NEUROTIC’]

<time>: [0-2][0-9] ‘.’ [0-2][0-9] ‘.’ [0-2][0-9]

Theme Hospital Lite

7. Input Specification

INPUT: <int> ‘\n’ [<node> ‘ ’ <link> ‘ ’ <node> ‘\n’]+ <int> ‘\n’ [<patient> ‘\n’]+ <int> ‘\n’ [<infodesk/treatment> ‘\n’]+ ‘\n’ <int> [<doctor> ‘\n’]+ ‘\n’

<infodesk/treatment>: <node> ‘:’ <service-time>

<service-time>: <int>

<doctor>: <name> ‘:’ <speed-multiplier> ‘:’ <service-time>

Theme Hospital Lite

8. Output Specification

Output:

Which doctors are you going to use and in which rooms they should begin in + when the last patient leaves the hospital (reaches his/her exit ENTRANCE node).

The hospital opens at 08:00:00.

The hospital closes at 18:00:00.

[<doctor-start> ‘\n’]+ <finishing-time>

<doctor-start>: <name> ‘:’ <node>

<finishing-time>: <time>

Theme Hospital Lite

8. Output Specification

Example output:

```
DoktorJaroslav:GP-1
DoktorkaJaroslava:GP-2
14.05.26
```

Theme Hospital Lite

9. Task summary

- You are given
  - A hospital
  - Patients
  - Pool of doctors
- You have to come up with a solution that consists of
  - Which doctors are you going to use for a day
  - Which rooms they will start in (at 8:00:00)
- Such that
  - All patients get treated and exit the hospital by the hospital’s closing time (18:00:00)
- Try to come up with the best solution
  - That is to minimize number of doctors you need + finishing the day as fast as possible
  - You can score bonus points if you come up with one of the best solutions! (up to +10)

Assignment 7

Send me an email

- Email: jakub.gemrot@gmail.com
- Subject: **Programming II – 2016 – Assignment 07**
- Zip up the whole project and send it
- You WILL NOT find the assignment in CoDex!
- Deadline: **30.9.2016 23:59**

Questions?

I sense a soul in search of answers…

- In case of doubts about the assignment or some other problems don’t hesitate to contact me!
  - Jakub Gemrot
    - gemrot@gamedev.cuni.cz