Student ID: **ID-No.:**

Systems and Software Engineering
Examination WS 2012/2013

Institut für Technik der Informationsverarbeitung – ITIV
Prof. Dr.-Ing. Müller-Glaser ▪ Dipl.-Ing. Stephanie Friederich

Systems and Software Engineering
Date: 05.04.2013
Name:
Matriculation ID:

Lecture Hall:
Seat No.:

Prerequisites for the examination

Aids

- Allowed aids for the examination are writing utensils and a single sheet of A4 paper with self- and hand-written notes. Writing may be on both sides of the paper. The use of own concept paper is not allowed.
- Use only indelible ink - use of pencils and red ink is prohibited.
- Other material than that mentioned above, is strictly forbidden. This includes any type of communication to other people.

Duration of the examination

120 minutes

Examination documents

The examination comprises 202 pages (including title page). Answers may be given in English or German. A mix of language within a single (sub)-task is not allowed. In your solution mark clearly which part of the task you are solving. Do not write on the backside of the solution sheets. If additional paper is needed ask the examination supervisor.
You will not be allowed to hand in your examination and leave the lecture hall in the last 30 minutes of the examination.
At the end of the examination: Stay at your seat and put all sheets into the envelope. Only sheets in the envelope will be corrected. We will collect the examination.

		Page	*Points*	*Result*
Task 1	General Questions	2	11	
Task 2	House of Quality	4	8	
Task 3	Petri Nets	6	8	
Task 4	State Charts	8	8	
Task 5	Scheduling	11	17	
Task 6	Reliability	14	8	
Task 7	UML Diagrams	15	18	
Task 8	Other diagrams	19	10	
			89	

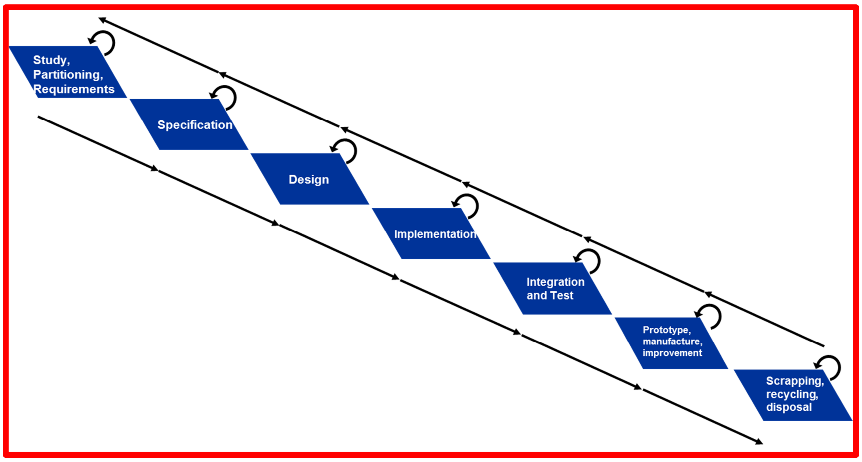
1 General Questions

1.a Lifecycle models

/3

Briefly describe the Waterfall Lifecycle model.

In what ways is the Waterfall Lifecycle model different to other Lifecycle models?



Alternative:

The waterfall lifecycle model includes phases: system engineering, requirements analysis, design, testing, construction, installation, and maintenance.

Its distinguishing feature is that work on each phase is supposed to be complete before moving onward. Backward moves are only allowed to the previous phase.

1.b V-model

/2

Name the four sub models of the V-Model.

PM – Project Management

QA – Quality Assurance

CM – Configuration Management

SD – System Development

1.c Reliability

/1

f(t) is the failure function (probability density function) of a system. How can the Reliability R(t) be computed from f?

$$R(t) = 1 - \int_0^t f(t)dt \qquad R(t) = \frac{1}{\lambda} f(t) \quad wenn\ \lambda\ const$$

Multiple choice

You get plus 0.5 points for a right answer and minus 0.5 points for a wrong answer. In total you cannot get a negative number of points for each subtask. More than one answer could be true.

1.d More than Moore

/1.5

What does 'More than Moore' means?

true	false	
☐	☑	The number of transistors on integrated circuits more than doubles every two years.
☑	☐	Diverse Technologies and Functions are implemented on a Single Chip or a system-in-Package.
☐	☑	Moore machine which is extended by concurrency.

1.e Watchdog timer

/1.5

Which statements could be made for software watchdog timer and hardware watchdog timer?

true	false	
☐	☑	It is not possible to implement both, hardware and software timer in one system.
☑	☐	Timer in software runs much slower than the system clock.
☑	☐	Time resolution of the hardware timer has the same resolution as the system clock.

1.f Reliability

/2

The reliability function of a system is defined as $R(t) = R_1(t) + R_2(t) - R_1(t)*R_2(t)$. What statements could be made about the two elements E1 and E2?

true	false	
☐	☑	Elements E1 and E2 are serially connected.
☑	☐	Elements E1 and E2 are connected in parallel.
☐	☑	The failure rates of E1 and E2 are considered as being constant.
☑	☐	Elements E1 and E2 are stochastically independent.

2 House of Quality

Your task is to do the analysis of a new digital camera. Thus, you decide to determine important design characteristics and critical system components of the new device. For this task you do the Quality Function Deployment and create a “House of Quality”. Implement the quality deployment in Figure 1.

From a public-opinion poll you know that the most important things which people want in a camera are “Image Quality” (100%), “Usability” (90%), “Portability” (60%) and “Movies” (40%). The technical design characteristics you want to evaluate are “CCD quality” (sensor for recording the images), “Size/Weight”, “Display quality” and “Battery quality”.

Table 1 shows a feature comparison between your old camera model and the actual camera of your main competitor.

Feature	Own previous model	Competitors model
Resolution	8.0 Mio Pixel	6.0 Mio Pixel
LCD size	1.5 inch	2.0 inch
Optical zoom	3x	3x
Weight	160g	130g
Size	88mm x 60.5mm x 32.7mm	86,0mm x 53,0mm x 20,7mm
Shutter speeds	15s – 1/2000s	1/8s - 1/2000s
Lens focal width	39 mm – 117 mm	36mm - 108 mm
Battery charging time	6h	8h
Video	640x200, 30 frames/s	320x200, 30 frames/s
Connectors	USB 2.0	USB 3.0, Video
Accessories	Software, Cables, AC adapter kit, bag	Cables, AC adapter kit

Table 1: Digital camera model comparison

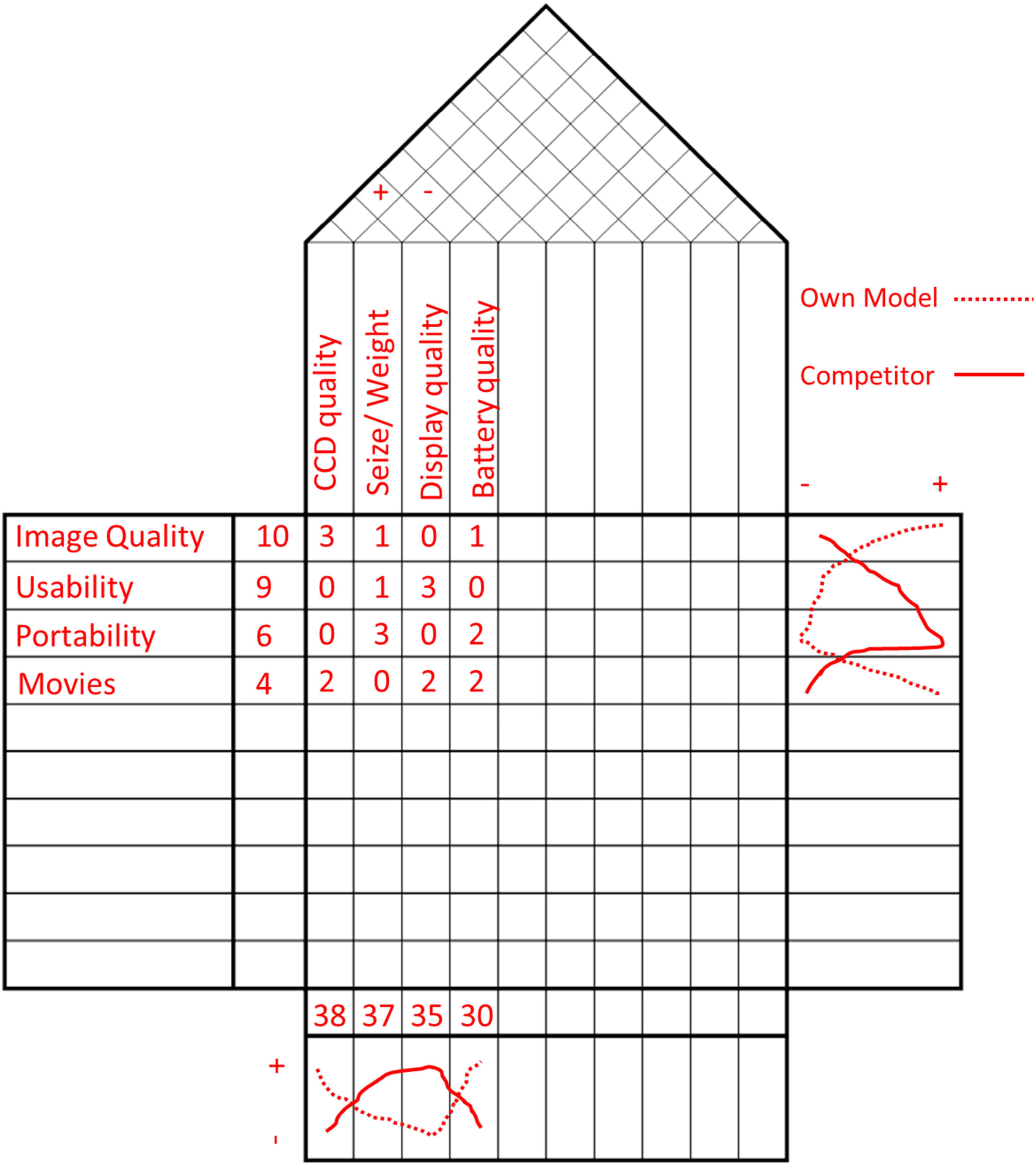


Figure 1: House of quality

3 Petri nets

A petri net is formally described by $N = (P, T, F, M_0)$.

P = {p_1, p_2, p_3, p_4, p_5}

T = {t_1, t_2, t_3, t_4}

$F = W^+ \cup W^-$

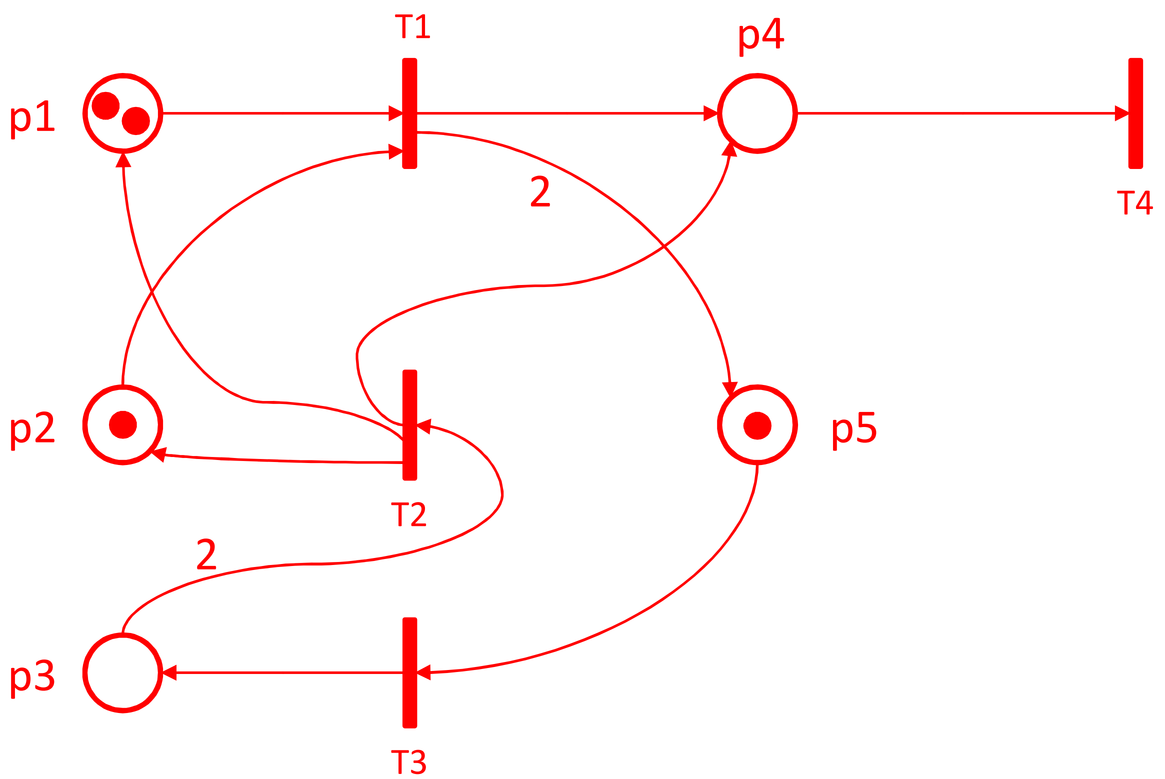
$$W^+ = \begin{bmatrix} & t_1 & t_2 & t_3 & t_4 \\ p_1 & 1 & 0 & 0 & 0 \\ p_2 & 1 & 0 & 0 & 0 \\ p_3 & 0 & 2 & 0 & 0 \\ p_4 & 0 & 0 & 0 & 1 \\ p_5 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad W^- = \begin{bmatrix} & t_1 & t_2 & t_3 & t_4 \\ p_1 & 0 & 1 & 0 & 0 \\ p_2 & 0 & 1 & 0 & 0 \\ p_3 & 0 & 0 & 1 & 0 \\ p_4 & 1 & 1 & 0 & 0 \\ p_5 & 2 & 0 & 0 & 0 \end{bmatrix}$$

$M_0[2 \quad 1 \quad 0 \quad 0 \quad 1]$

3.a Graphical representation

/3

Draw the graphical representation of the described net.



3.b Analysis of the net

/2

Is the formally described Petri net conservative? Explain your answer.

No it is not conservative.

Transition T4 for example has only one input but no output; hence it is possible to decrease the number of tokens.

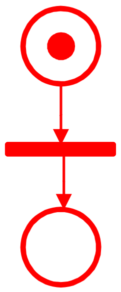
Transition T1 has more outputs than inputs.

3.c Reversibility

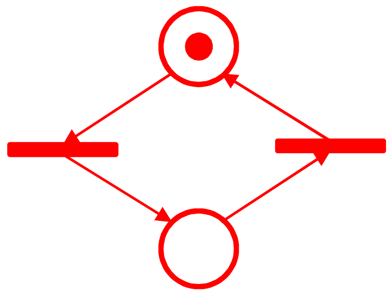
/3

Draw two Petri nets, one which is reversible and the other one that is not reversible.

Not reversible:



Reversible:



4 State charts

Figure 2 shows a state chart of an mp3-Player with battery and shock control.

Functional description:

The player could only perform a function if a battery voltage from at least 3 Volt is measured. During the operation a hard disk is accessed. In case a shock is detected, the hard disk read heads are parked. Track number, title, time and artist could be displayed. For operation the buttons start, stop, play, pause and mode could be used.

The simulation input values are given in Figure 3.

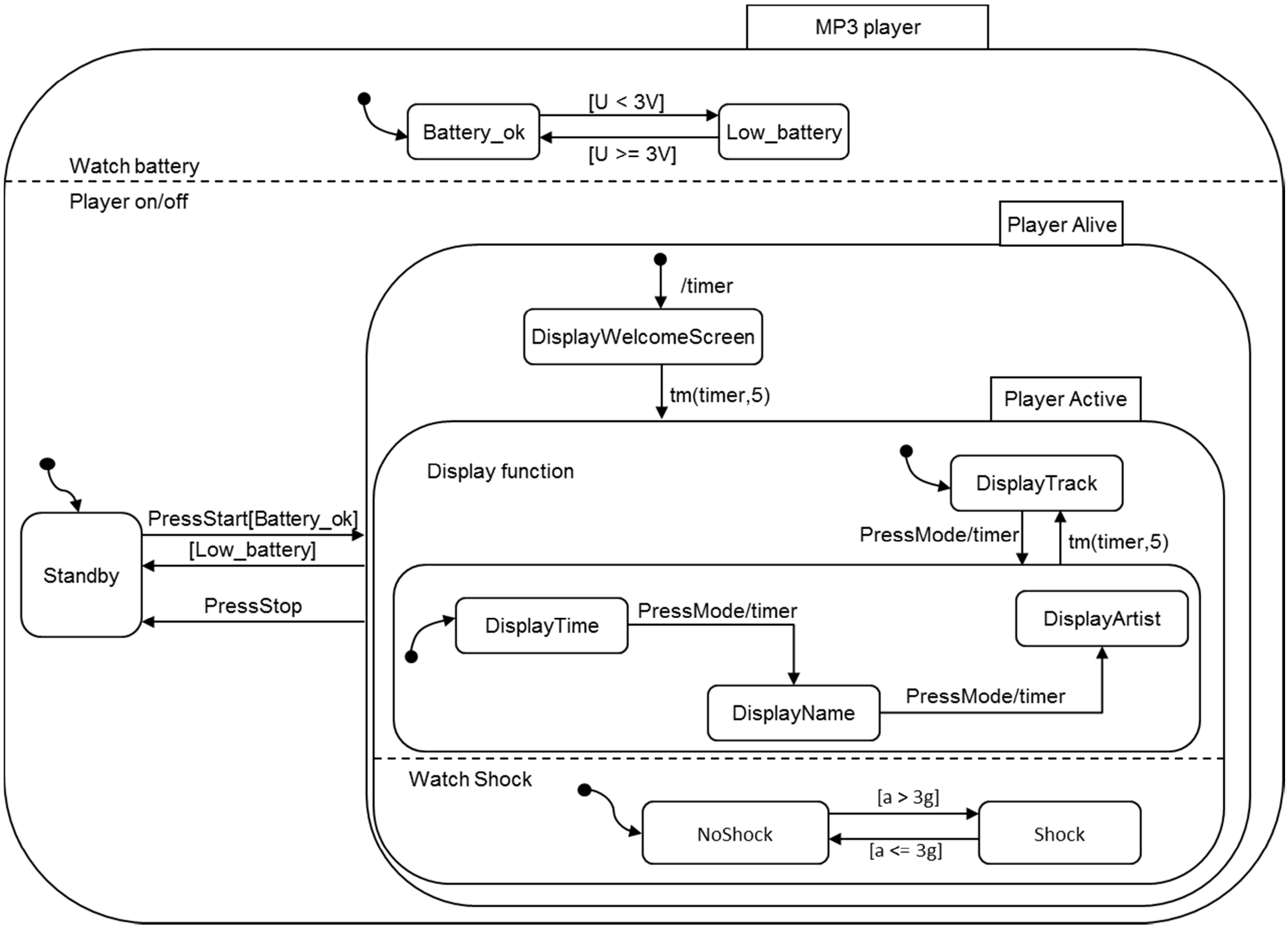


Figure 2: State chart of an mp3-player

4.a Basic states

/2

How many states are necessary if you transform the state chart in Figure 2 into a state chart containing basic states only? Write down the solution steps!

= 2*(1+1+2*(1+3) = 20

= WatchBattery * (Standby + DisplayWelcome + WatchShock * (DisplayFunction)

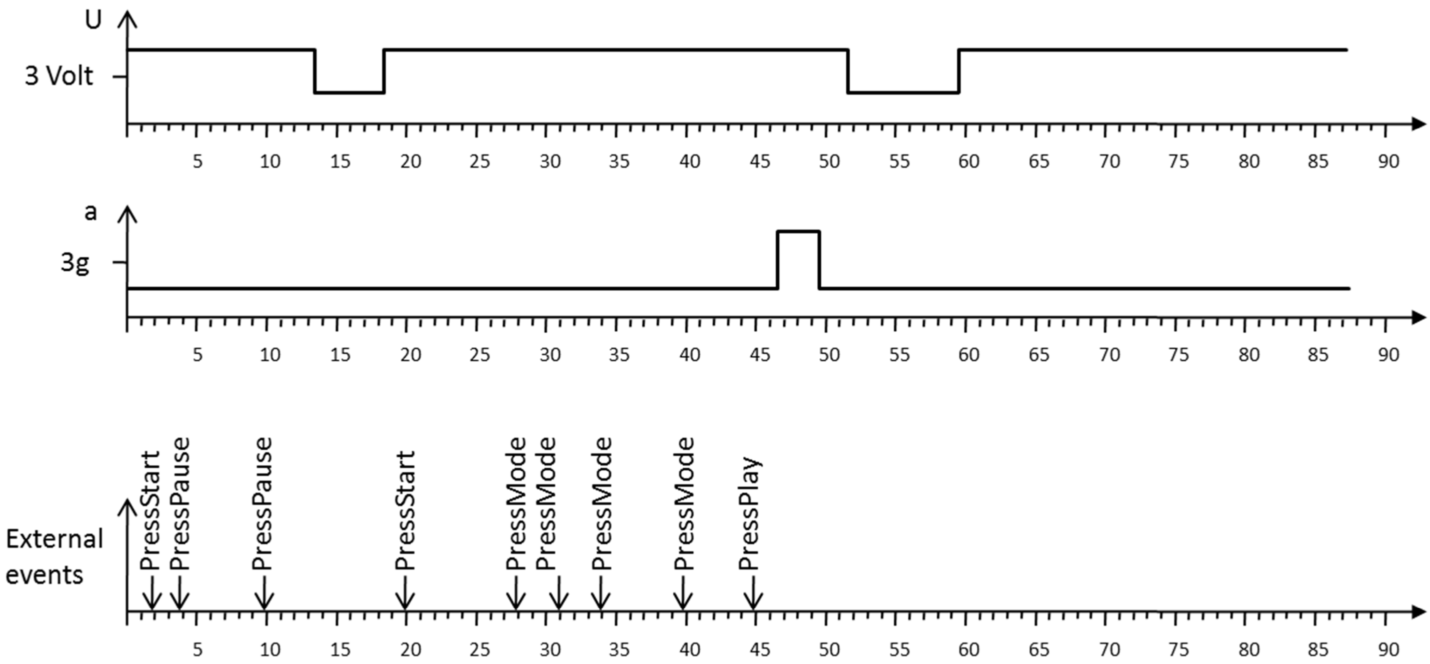


Figure 3: Simulation Inputs

4.b State chart analysis

/6

Complete Table 2, filling in the sequence of active basic states of Figure 2 considering the simulation input values are given in Figure 3. Only changes must be considered.

Simulation step	Active basic states
0	Battery_ok; Standby
2	Battery_ok; DisplayWelcomeScreen
7	Battery_ok; DisplayTrack; NoShock
14	Low_Battery; DisplayTrack; NoShock
15	Low_Battery; Standby
19	Battery_ok; Standby
20	Battery_ok; DisplayWelcomeScreen
25	Battery_ok; DisplayTrack; NoShock
28	Battery_ok; DisplayTime; NoShock
31	Battery_ok; DisplayName; NoShock
34	Battery_ok; DisplayArtist; NoShock
39	Battery_ok; DisplayTrack; NoShock
40	Battery_ok; DisplayTime; NoShock
45	Battery_ok; DisplayTrack; NoShock
47	Battery_ok; DisplayTrack; Shock
50	Battery_ok; DisplayTrack; NoShock
52	Low_Battery; DisplayTrack; NoShock
53	Low_Battery; Standby
60	Battery_ok; Standby

Table 2: Active Basic States

5 Scheduling

5.a Scheduling-strategies for tasks

/9

Five tasks with different priority should be executed on one processor. Table 3 shows the features of these tasks.

Task	Processing Time	Priority (0 is highest)	Arrival time	Deadline
A	17	2	T + 0 ms	T + 160 ms
B	33	3	T + 7 ms	T + 120 ms
C	49	1	T + 11 ms	T + 110 ms
D	25	0	T + 18 ms	T + 70 ms
E	31	4	T + 9 ms	T + 95 ms

Table 3: Tasks

You are at a certain point in time T, all tasks request processor-time at this point in time. Plot the processing of the given tasks into the following diagram (see Figure 4 next page) under consideration of the scheduling methods named below.

Round Robin (time slice 15ms) (Task Queue)

TDMA (time slice 10ms) (Cycle A-B-C-D-E, A starts at T)

Priority Scheduling

Deadline Scheduling

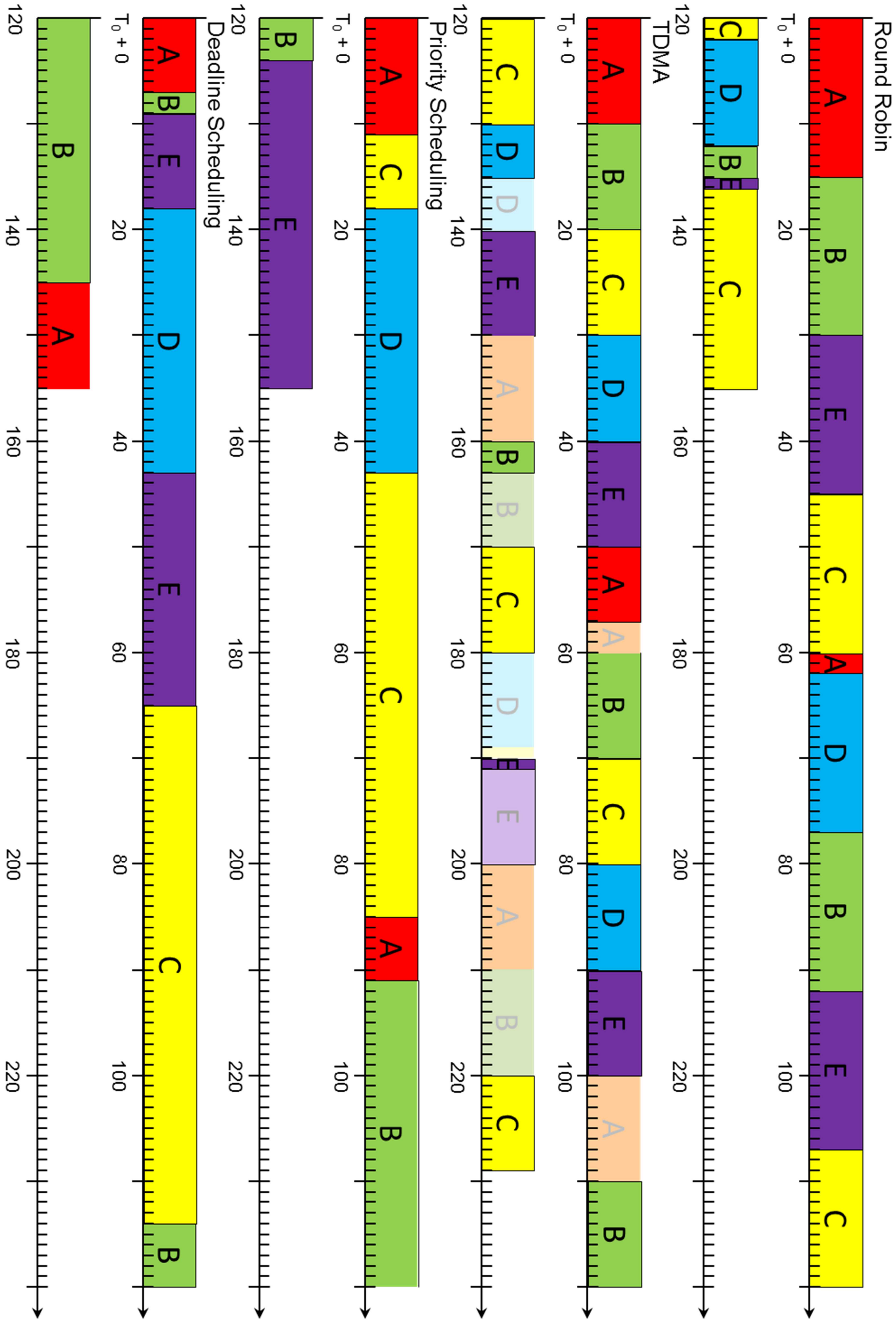


Figure 4: Scheduling strategies

5.b Response time

/4

Calculate the maximal and the average response time of the Round Robin and TDMA scheduling methods.

$$T_{\mathrm{Re}\,s,Max}(RR) = T_{\mathrm{Re}\,s,A} = 144\,\mathrm{ms}$$

$$T_{\mathrm{Re}\,s,Avg}(TDMA) = \frac{T_{\mathrm{Re}\,s,A} + T_{\mathrm{Re}\,s,B} + T_{\mathrm{Re}\,s,C} + T_{\mathrm{Re}\,s,D} + T_{\mathrm{Re}\,s,E}}{5}$$

$$= \frac{(62-0)+(135-7)+(155-11)+(132-18)+(136-9)}{5} = \frac{62+128+144+114+127}{5} = \frac{575}{5} = 115\,\mathrm{ms}$$

$$T_{\mathrm{Re}\,s,Max}(\mathrm{TDMA}) = T_{\mathrm{Re}\,s,A} = 218\,\mathrm{ms}$$

$$T_{\mathrm{Re}\,s,Avg}(TDMA) = \frac{T_{\mathrm{Re}\,s,A} + T_{\mathrm{Re}\,s,B} + T_{\mathrm{Re}\,s,C} + T_{\mathrm{Re}\,s,D} + T_{\mathrm{Re}\,s,E}}{5}$$

$$= \frac{(57-0)+(163-7)+(229-11)+(135-18)+(191-9)}{5} = \frac{57+156+218+117+182}{5} = \frac{730}{5} = 146\,\mathrm{ms}$$

5.c Scheduling strategy

/1

Which scheduling strategy has to be implemented if you want to obtain a minimal average response time. Take into account all scheduling strategies which have been discussed in the lecture and assume that all tasks arrive at the same time.

Shortest Job First

5.d Calculation of CPU usage

/3

Give a formula for the CPU-usage for Round-Robin scheduling as the ratio between the time used for actual task processing and the overall time (task processing plus time for switches). Assume that there are always tasks ready to be executed.

Which CPU-usage is obtained if the time slice leans toward zero?

$Number\ of\ time\ slices = \frac{t}{q}$ $\qquad$ $Total\ time\ for\ task\ switches = \frac{t}{q}s$

$$Total\ processing\ time = t + \frac{t}{q}s$$

$$CPU\ usage = \frac{t}{t + \frac{t}{q}s} = \frac{1}{1 + \frac{s}{q}}$$

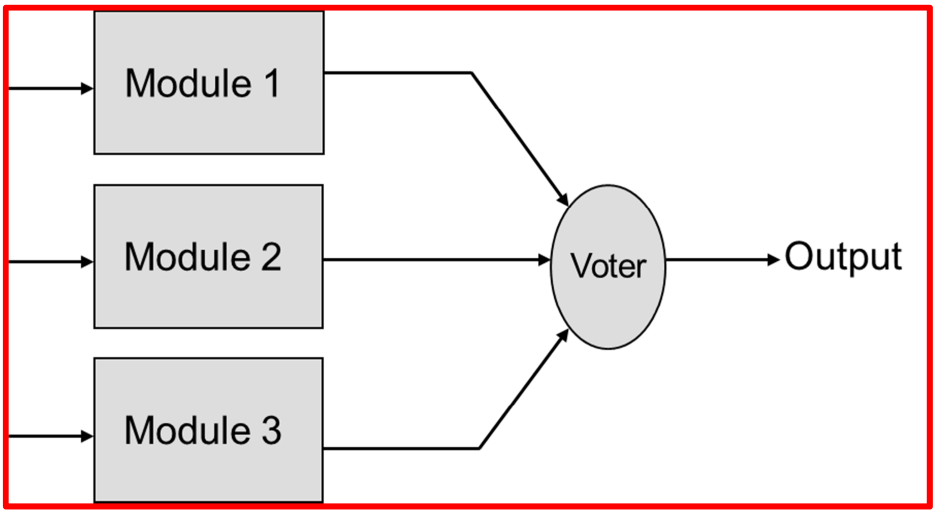
If q → 0 than the CPU usage → 0.

6 Reliability

6.a TMR

/3

Sketch a diagram of a TMR system and explain the functionality of the system.



TMR: Three systems perform a process and that result is processed by a majority-voting system to produce a single output. If any one of the three systems fails, the other two systems can correct and mask the fault. If the voter fails then the complete system will fail.

6.b Terms

/3

Explain the three terms error, failure and fault.

Error: The occurrence of an incorrect value in some unit of information within a system

Failure: Undue deviation from functional features and features of performance of a system

Fault: a physical defect, imperfection, or flaw that occurs in hardware or software

6.c Mean time to failure

/2

Give the general equation for the MTTF and determine the MTTF for a system with a constant failure rate of 0.0002.

$$MTTF = \int_0^\infty R(t)dt = \int_0^\infty e^{-\lambda t}dt = \left[-\frac{1}{\lambda}e^{-\lambda t}\right]_0^\infty$$

$$MTTF_{\lambda=constant} = \frac{1}{\lambda} = \frac{1}{0.0002} = 5000$$

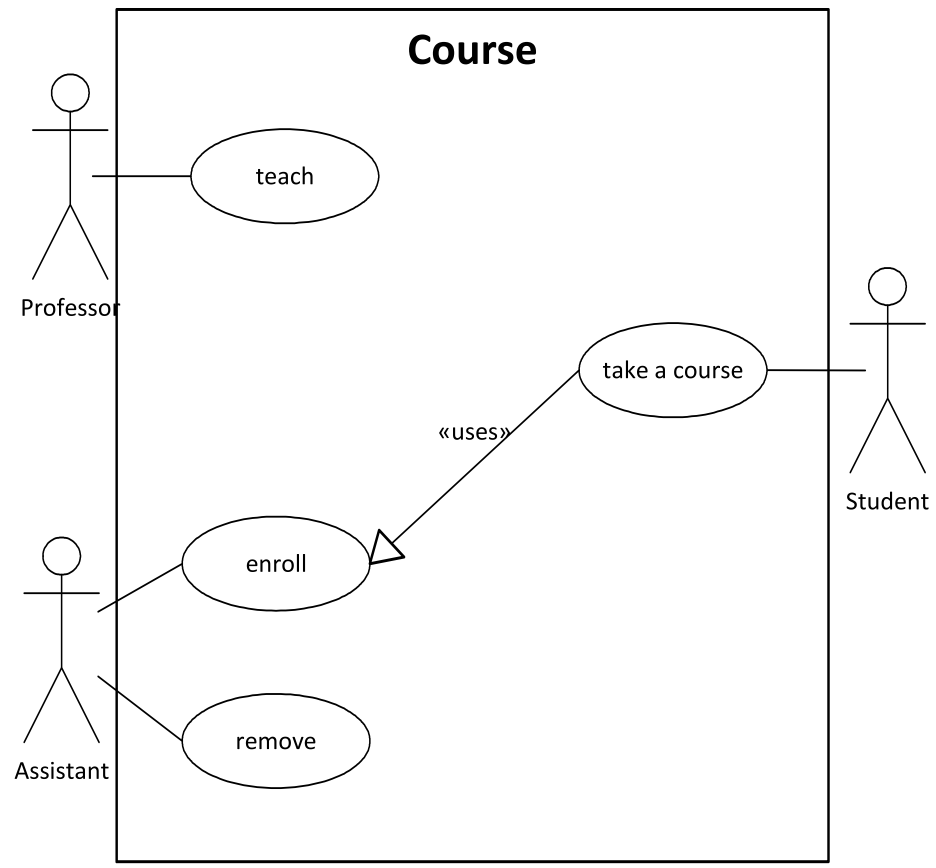
7 UML Diagrams

7.a Use case diagram

/4

Draw a use case diagram for university courses.

In particular, assume that courses are taught by professors, while an assistant can enroll or remove students from a course. Students take a course, provided they are enrolled in it.



7.b Class diagram

/5

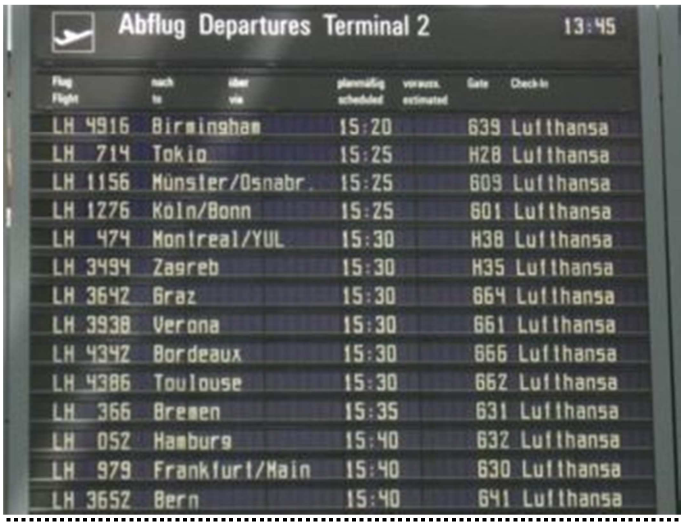
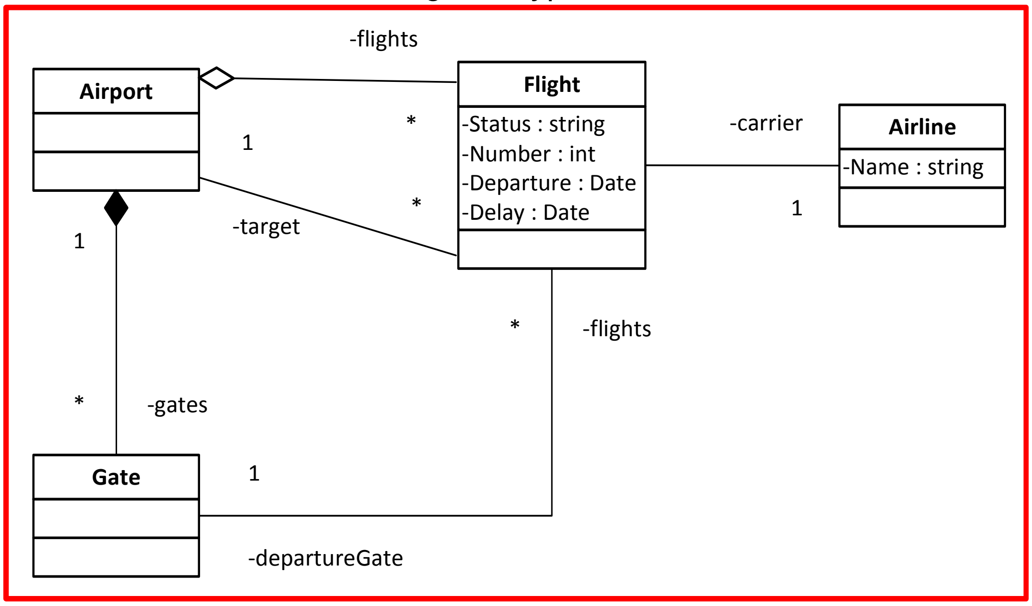


Figure 5: Airport display

You are working for a company designing a display system for an airport similar to the one in the picture above. Its task is to show information on departing flights, their status, departure times and boarding gates. Design a class model that is capable of storing all the information needed by the display system. Adhere to the following constraints:

- One airport can provide any number of flights.
- A flight has a scheduled departure time but can be delayed.
- Each flight has a target Airport and a flight number.
- Each flight can have the status “not boarding”, “boarding”, “last call” or “departed”.
- Each gate of an Airport can be used to board several flights.
- Each flight is served by one Airline.

Design a class diagram of the described airport display system. Add associations, multiplicities and attributes with fitting data types.



7.c Sequence diagram

/5

Convert the UML Communication diagram in Figure 6 into a UML Sequence Diagram using as much information as possible.

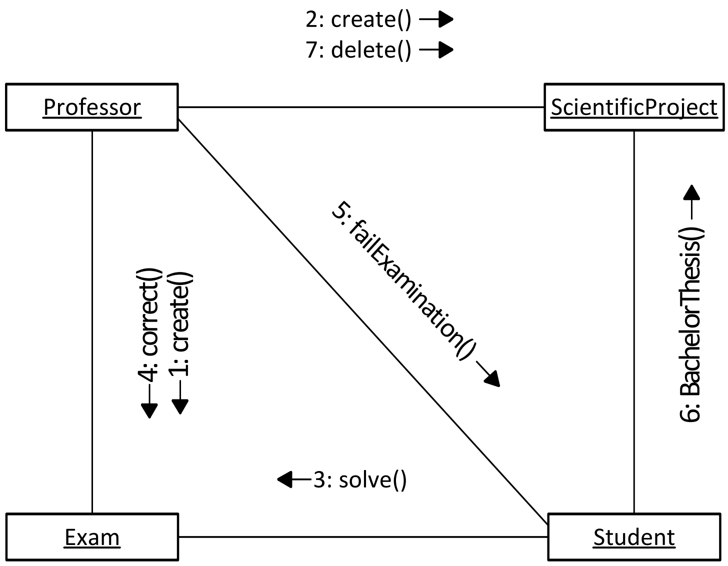
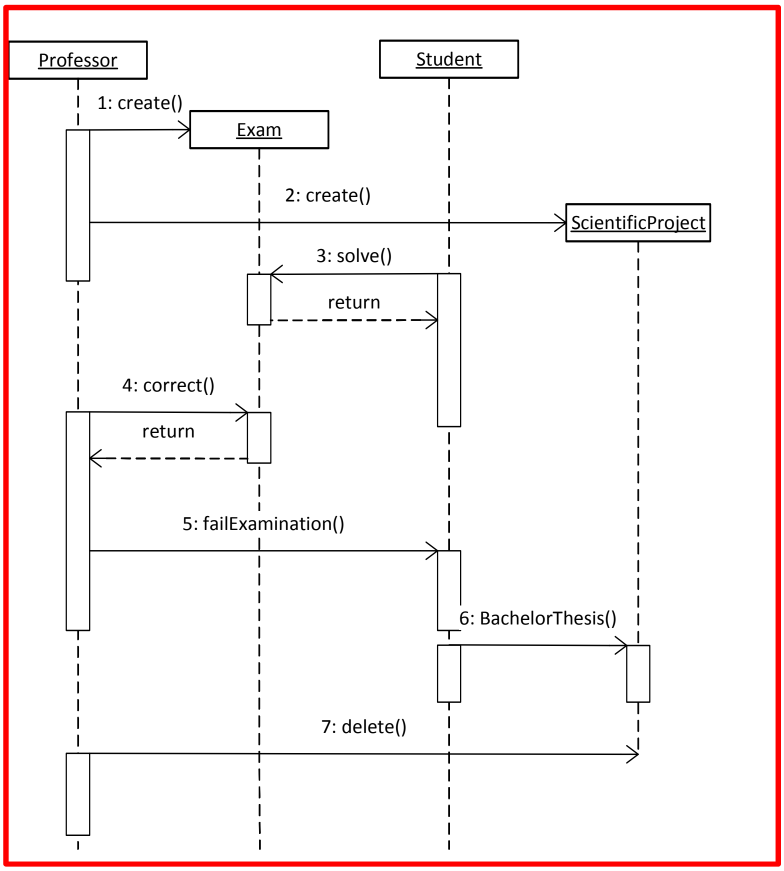


Figure 6: UML Communication diagram



7.d Visibility

/4

Given the UML Class Diagram in Figure 7 draw an UML Communication Diagram with all possible messages (operation calls).

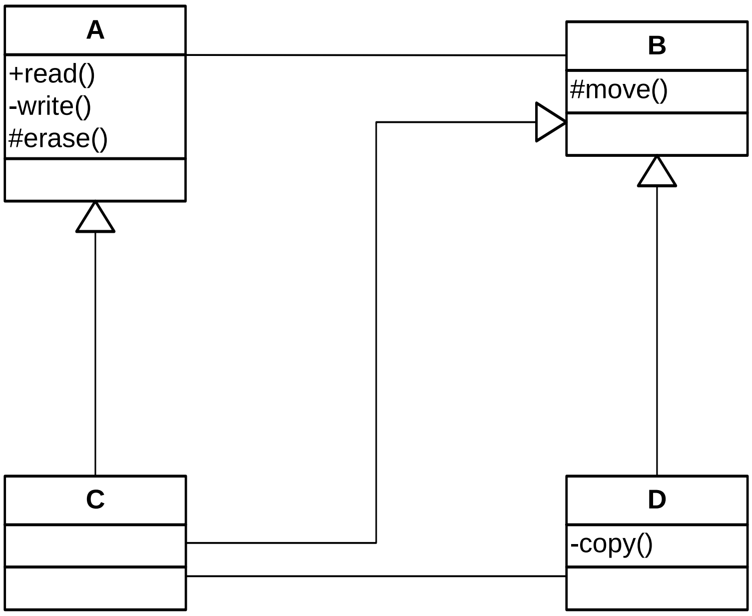
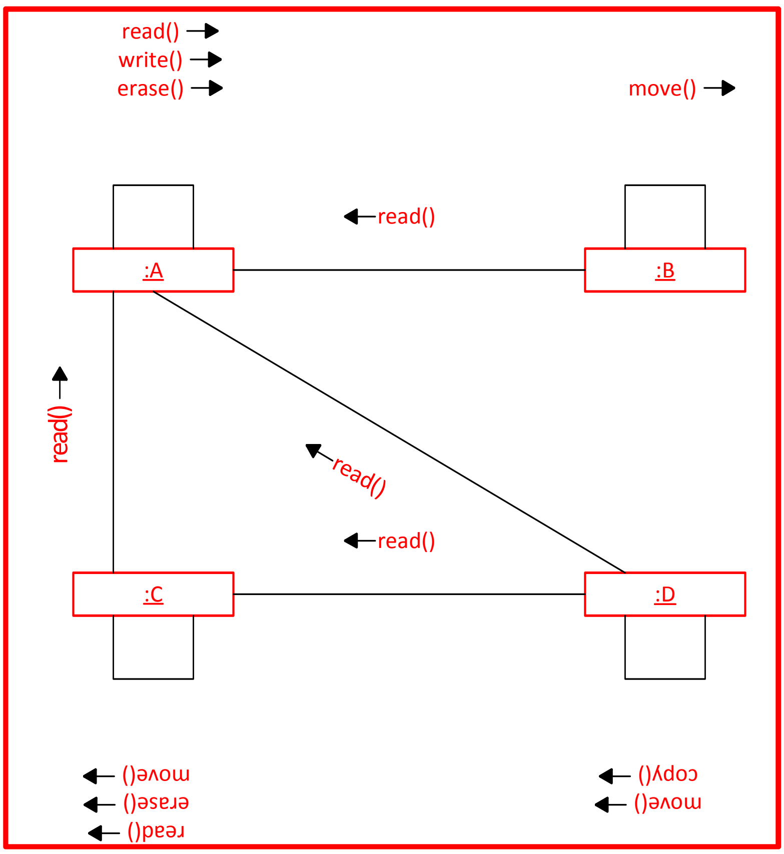


Figure 7: UML Class diagram



8 Other diagrams

8.a Backus Naur Form

/2

Write down the following EBNF expressions in BNF.

expression ::= term { ('Plus' | 'Minus') term }

term ::= factor { ('Mult' | 'Div') factor }

factor::= 'Number' | 'Minus' 'Number'

expression ::= expression 'Plus' term | expression 'Minus' term | term

term ::= term 'Mult' factor | term 'Div' factor | factor

factor::= 'Number' | 'Minus' 'Number'

8.b Extended Backus Naur Form

/4

Design an EBNF expression which only defines floating point numbers. Examples for producible expressions are shown below.

1.234
+50000
-9.3021
+1.432E4
0
1E10
-4932E-13

D := '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

S := '-' | '+'

F := (S)? (D)+ ('.' (D)+)? ('E' (S)? (D)+)?

8.c Nassi-Schneiderman diagram

/4

Analyze the following program written in C++/Java by drawing a Nassi-Shneiderman diagram.

```
void func(int x[], int y) {
     for (int i = 0; i < y-1; i++) {
          for (int j = 0; j < y-i-1; j++) {
               if ( x[j] > x[j+1] ) {
                    int z = x[j];
                    x[j] = x[j+1];
                    x[j+1] = z;
               }
          }
     }
}
```