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

Systems and Software Engineering
Examination ST 2013

Institut für Technik der Informationsverarbeitung – ITIV
Prof. Dr.-Ing. Müller-Glaser

Systems and Software Engineering
Date: 13.09.2013
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 21 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.

1 General Questions

1.a Software test

/2

Name the different phases of the software test process, discussed in the lecture.

Unit testing

Module testing

(Sub-system testing)

System testing

Acceptance testing

/1

1.b BNF

Explain shortly, the reason for using the Backus-Naur-Form.

The BNF is a formal way to describe a language.

/3

1.c UML

Name eight different UML 2.0 diagrams and classify them in Table 1.

Structural UML diagrams	Behavioral UML diagrams
Class diagram	Sequence diagram
Object diagram	Communication / Collaboration diagram
Component diagram	State diagram
Deployment diagram	Activity diagram
Composite structure diagram	Timing diagram
Package diagram	Interaction (overview) diagram
Profile diagram	Use Case diagram

Table 1: UML diagrams overview

/2

1.d Hunger Life Cycle Model

What is the difference between the Sortie Mission and the Life Mission?

Sortie mission:

Scenario that describe how the system will be used during operational phase, capturing the reasons the system has for existing

Life mission:

Address the non-operational life cycle aspects of the system, resulting in scenarios for each life-cycle phase and some that cross life-cycle phases

/2

1.e V-model

Name the four sub models of the V-Model ‘97.

PM – Project Management

QA – Quality Assurance

CM – Configuration Management

SD – System Development

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.f Real time systems

/1.5

Select the correct answers:

true	false	
☐	☑	Hard real-time systems need faster hardware than soft real-time systems.
☑	☐	The response time of a computing process in a real-time system is the sum of waiting-, input-, processing- and output-time.
☐	☑	In the context of a real-time operating system, the “first come first served” - strategy minimizes the average response time when tasks arrive at the same time.

1.g 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 processor clock.
☑	☐	Time resolution of the hardware timer has in best case the same resolution as the system clock.

2 Quality function deployment

2.a House of Quality

/2

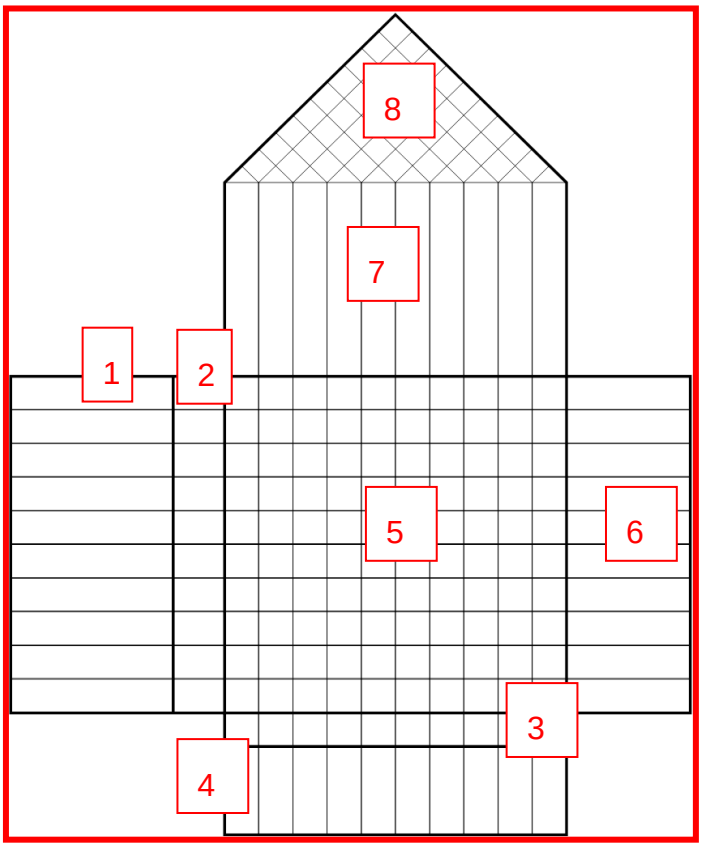
What is the primary function of the House of Quality tool?

The House of Quality tool provides a fast way to pull together the customer requirements and product features. It systematically flows down the requirements to lower levels of design, parts, manufacturing and production.

2.b House of Quality

/2

Draw the shape of a House of Quality with all its components.



/ 2

2.c House of Quality

Name all elements in the House of Quality and point out the location in the figure you drew in 2.b.

1. Customer Requirements / What?
2. Weighting of Customer Requirements
3. Prioritization
4. Technical Benchmarking
5. Dependencies
6. Comparison with competitor
7. Quality Features / How
8. Mutual Influence of Quality Features

/1

2.d House of Quality

Which gradations are commonly used in the relationship entries?

- Influence of Quality Features: Gradations are strong positive, positive, negative, and strong negative or neutral.
  Or Gradations are positive and negative or neutral.
- Dependencies: 0-3 where 0 means no dependency and 3 means a strong dependency.

3 Petri nets

/4

3.a Petri net characteristics

Which of the following characteristics (a-d) of petri nets are mutually exclusive? Give a short explanation for each.

a) Alive and reversible

b) Alive and dead marking

c) Reversible and not alive

d) Reversible and not safe

a) Mutually exclusive? Yes ☐ No ☑

Explanation:

It is possible that a reversible petri net is not alive, if any transition is dead at any point. But reversibility indicated that the initial marking can be reached from any other marking and this means that transitions which are live at the beginning are live the whole time.

b) Mutually exclusive? Yes ☑ No ☐

Explanation:

If there exist a dead marking, than there are all transitions dead at this point. But the condition for a live petri net is that all transitions are live.

c) Mutually exclusive? Yes ☐ No ☑

Explanation:

As in a) explained it is possible to have a petri net where one transition is not alive at the beginning and yet the net is reversible. (Not alive don't only apply dead markings, but also dead transitions)

d) Mutually exclusive? Yes ☐ No ☑

Explanation:

Not save means that the amount of tokens in a system increases, but the net can still be reversible if tokens are destroyed.

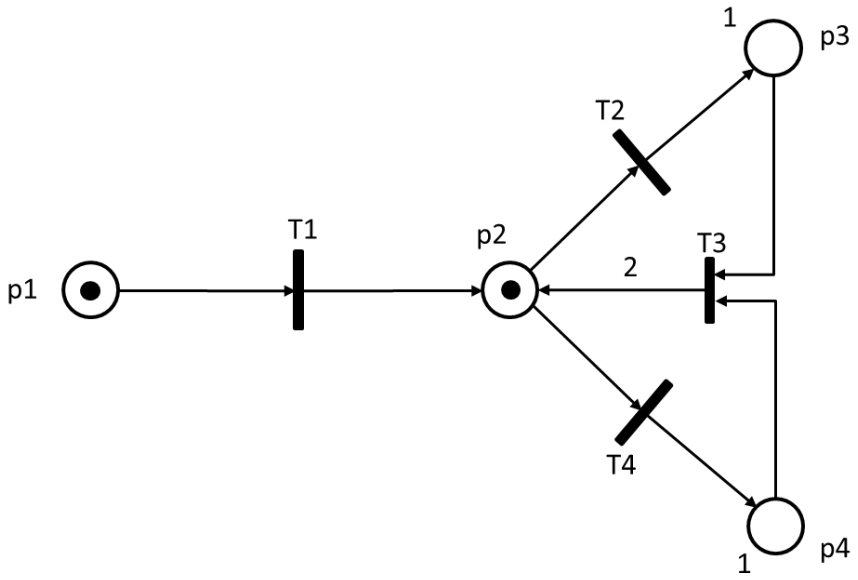


Figure 1: Petri net

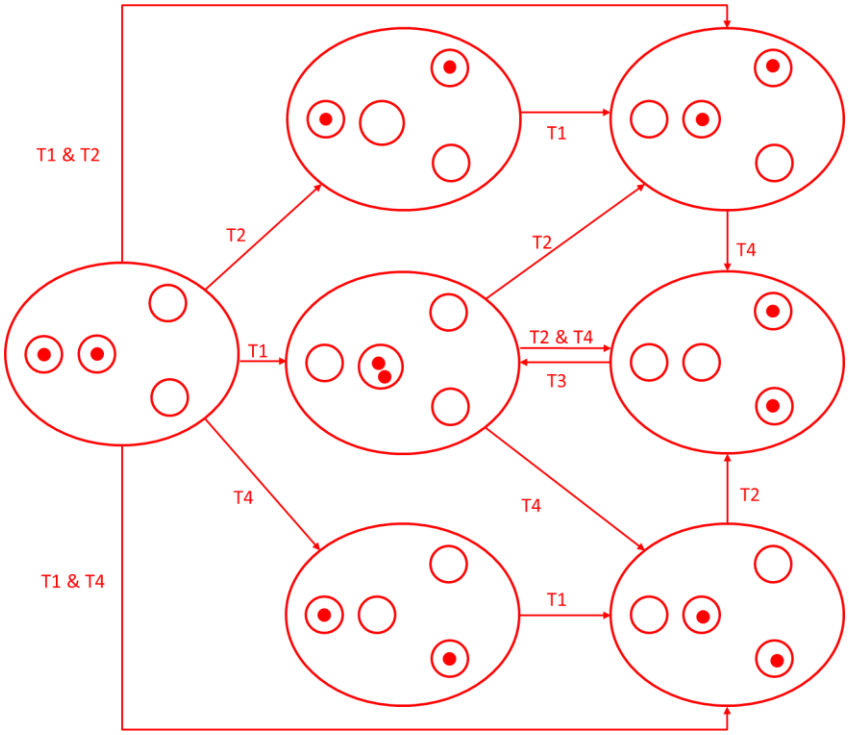
/5

3.b Petri net characteristics

Is the petri net shown in Figure 1 safe? Draw the transition graph of the petri net and explain your answer.

Yes.

The amount of tokens in the system doesn't increase; the maximum amount of tokens in one place is 2. Hence the petri net is 2-safe.



/2

3.c Petri net characteristics

Is the petri net shown in Figure 1 alive? Explain your answer.

No.

See transition graph answer 3.b. There is no deadlock, but transition T1 can only fire once, afterwards this transition is dead. Hence the petri net is not alive, because not all transitions are live

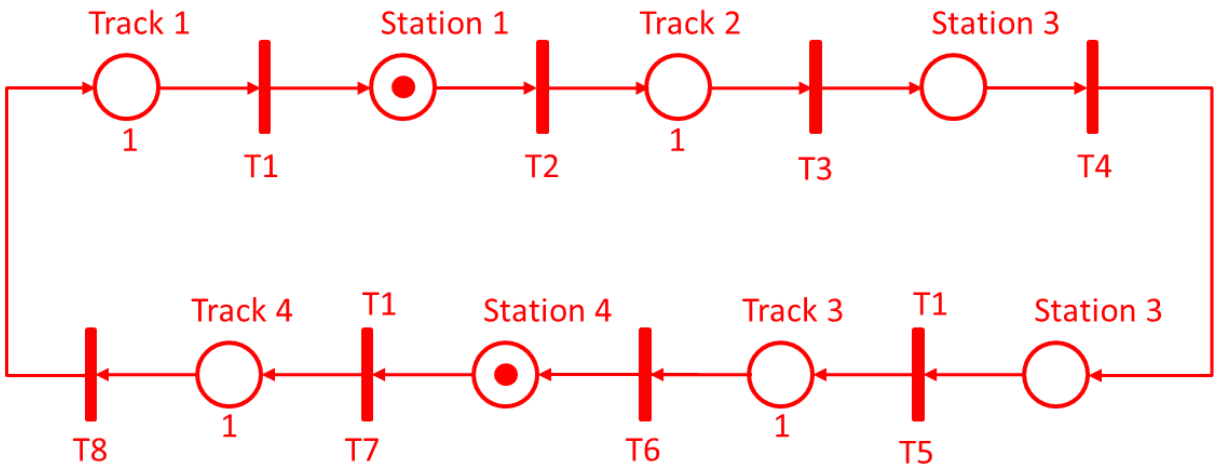
/5

3.d Design a petri net

Consider a circular railroad system with 4 tracks and 2 trains (represented by simple tokens). No two trains should be on the same track at the same time. The 4th track is connected to the 1st track. Between two tracks there is a station located.

Model a Petri net with simple tokens, so that both trains can use the tracks and travel conflict-free through the entire railroad system.

Point out the capacity of each place and the initial tokens to start from. Use *station 1* to *4* and *track 1* to *4* as labels for the specific parts of the net.



4 State charts

/2

4.a History connector

How is history connector “H” treated in a state chart diagram?

Hierarchical state.

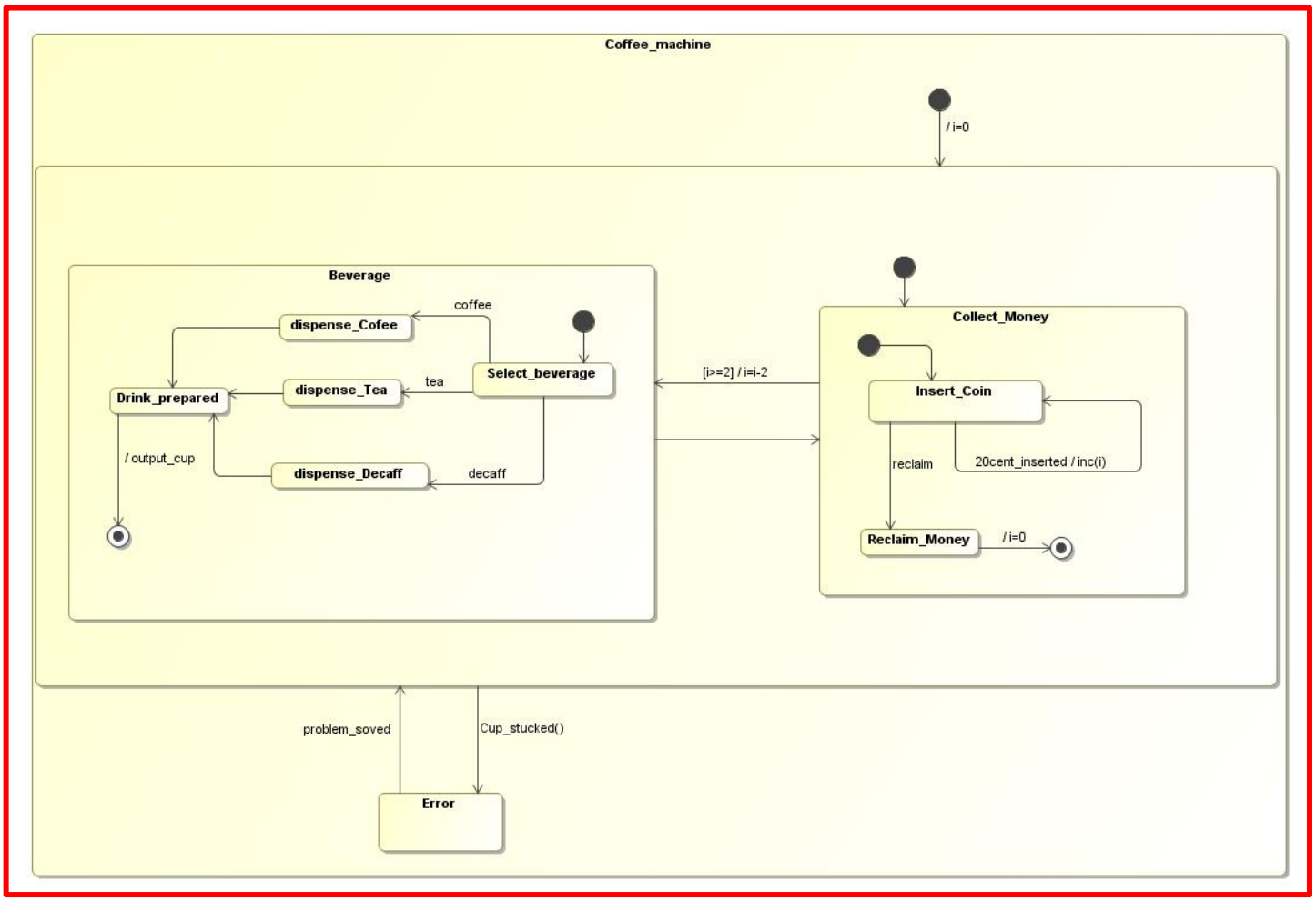
At first entry it will enter the default state, afterwards the encircled H as the entry point means to start in the state within that level which was exited before.

(Each level can have its own history mechanism.)

4.b Design a state chart

/8

Design a state chart for a simple coffee dispensing machine. The machine has two phases: in a first phase the money for the desired drink is collected. The money that can be inserted is 20 cent coins. Each beverage costs 40 cents. During the collection phase money can always be reclaimed. If at least 40 cents are stored in the machine, the user can choose between coffee, decaffeinated coffee and tea. After the drink was prepared the machine outputs the cup and returns to the collection phase. Pay attention to the following malfunction of the machine which may occur at any time: an error could occur and prevent the machine from behaving correctly. After such a malfunction is repaired, the machine automatically returns to the collecting phase.



5 Scheduling

/8

5.a Task scheduling

Four tasks with different priority should be executed on one processor. The following table shows the features of these tasks.

Task	Processing Time	Priority (0 is highest)	Arrival time	Deadline
A	32	2	T + 0 ms	T + 100 ms
B	19	0	T + 7 ms	T + 125 ms
C	45	1	T + 14 ms	T + 90 ms
D	27	3	T + 19 ms	T + 110 ms

Table 2: Scheduling

You are at a certain point in time T, all tasks request processor-time from this point in time. Plot the processing of the given tasks into the following diagrams (see next page Figure 2) under consideration of the scheduling methods named below.

- Round Robin (time slice 10ms) (Task Queue)
- TDMA (time slice 15ms) (Cycle A-B-C-D, A starts at T)
- Priority Scheduling
- Deadline Scheduling

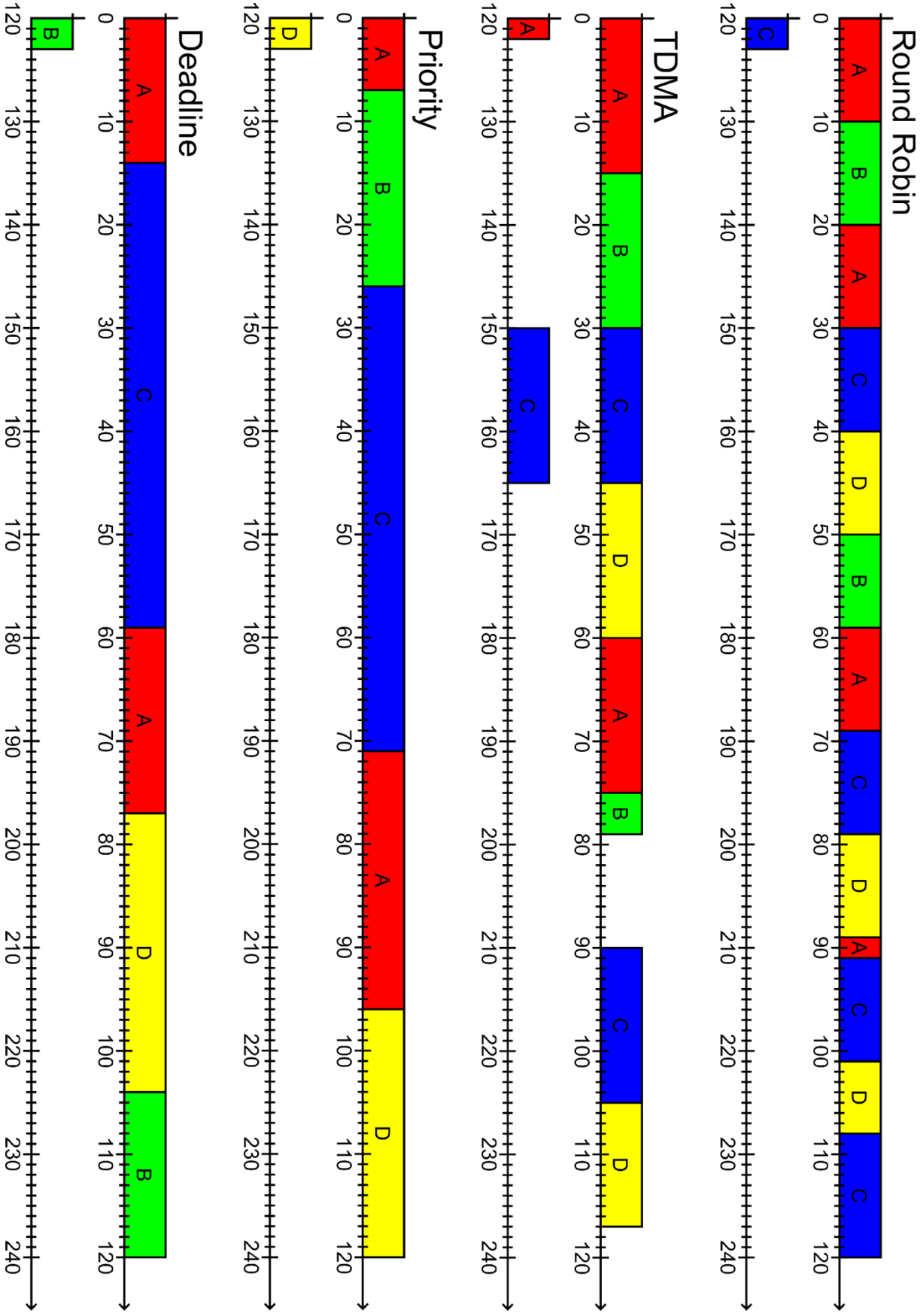


Figure 2: Scheduling

/5

5.b Response time

Calculate the maximal and the average response time of every method.

$$T_{\text{Res},Max}(\text{RR}) = T_{\text{Res},C} = 109\,\text{ms}$$

$$T_{\text{Res},Avg}(\text{RR}) = \frac{T_{\text{Res},A} + T_{\text{Res},B} + T_{\text{Res},C} + T_{\text{Res},D}}{4}$$

$$= \frac{(91-0) + (59-7) + (123-14) + (108-19)}{4} = \frac{91 + 52 + 109 + 89}{4} = 85{,}25\,\text{ms}$$

$$T_{\text{Res},Max}(\text{TDMA}) = T_{\text{Res},C} = 151\,\text{ms}$$

$$T_{\text{Res},Avg}(\text{TDMA}) = \frac{T_{\text{Res},A} + T_{\text{Res},B} + T_{\text{Res},C} + T_{\text{Res},D}}{4}$$

$$= \frac{(122-0) + (79-7) + (165-14) + (117-19)}{4} = \frac{122 + 72 + 151 + 98}{4} = 110{,}75\,\text{ms}$$

$$T_{\text{Res},Max}(\text{Priority}) = T_{\text{Res},D} = 104\,\text{ms}$$

$$T_{\text{Res},Avg}(\text{Priority}) = \frac{T_{\text{Res},A} + T_{\text{Res},B} + T_{\text{Res},C} + T_{\text{Res},D}}{4}$$

$$= \frac{(96-0) + (26-7) + (71-14) + (123-19)}{4} = \frac{96 + 19 + 57 + 104}{4} = 69\,\text{ms}$$

$$T_{\text{Res},Max}(\text{Deadline}) = T_{\text{Res},B} = 116\,\text{ms}$$

$$T_{\text{Res},Avg}(\text{Deadline}) = \frac{T_{\text{Res},A} + T_{\text{Res},B} + T_{\text{Res},C} + T_{\text{Res},D}}{4}$$

$$= \frac{(77-0) + (123-7) + (59-14) + (104-19)}{4} = \frac{77 + 116 + 45 + 85}{4} = 80{,}75\,\text{ms}$$

6 Reliability

Figure 3 shows part of a switchgear. The shown setup allows switching from the main power transformer T1 to the reserve power transformer T2.

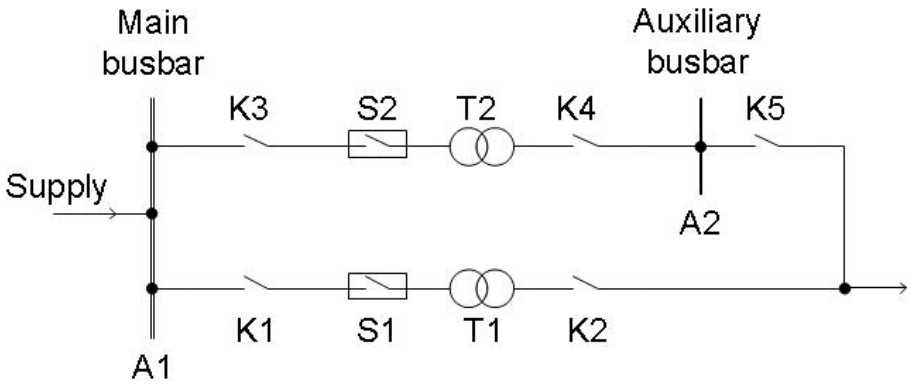


Figure 3: Part of a switchgear

Component type	Amount	Code	λ [10^{-6} / h]
Busbar	2	A1, A2	1,0
Power transformer	2	T1, T2	5,0
Disconnector	5	K1-K5	3,0
Power switch	2	S1, S2	10,0

Table 3: Switchgear components and failure rates

/1

6.a MTTF

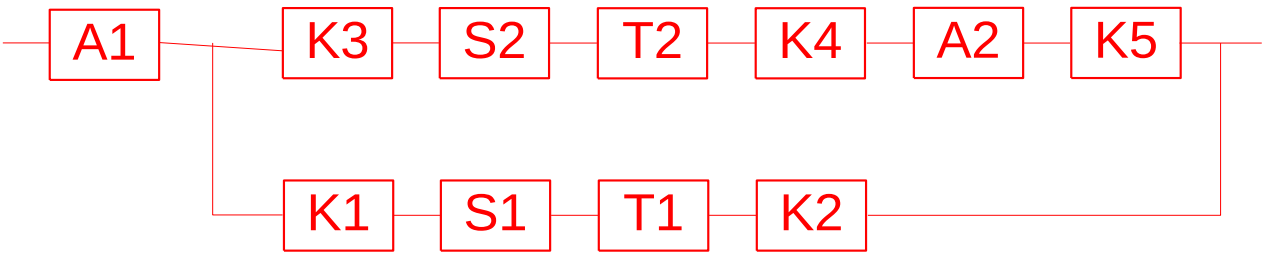
Calculate the MTTF for a single power transformer.

λ = const -> MTTF = 1/λ = 200000h = $2*10^5$h

/3

6.b Reliability block diagram

Draw a reliability block diagram for the whole system shown in Figure 3.



/6

6.c Reliability and MTTF

Determine the reliability R(t) and the MTTF of the electronic switch shown in Figure 3. The answer should compromise the intermediate steps of your calculation.

$$\lambda_{A1} = 1*10^{-6}/h$$

$$\lambda_{E1} = \lambda_{K3} + \lambda_{S2} + \lambda_{T2} + \lambda_{K4} + \lambda_{A2} + \lambda_{K5} = 25*10^{-6}/h$$

$$\lambda_{E2} = \lambda_{K1} + \lambda_{S1} + \lambda_{T1} + \lambda_{K2} = 21*10^{-6}/h$$

$$R_{E3}(t) = R_{E1}(t) + R_{E2}(t) - R_{E1}(t) * R_{E2}(t)$$

$$= e^{-\lambda_{E1}t} + e^{-\lambda_{E2}t} - e^{-(\lambda_{E1}+\lambda_{E2})t}$$

$$R(t) = R_{A1}(t) * R_{E3}(t) = e^{-(\lambda_{E1}+\lambda_{A1})t} + e^{-(\lambda_{E2}+\lambda_{A1})t} - e^{-(\lambda_{E1}+\lambda_{E2}+\lambda_{A1})t}$$

$$= e^{-t*26.10^{-6}/h} + e^{-t*22.10^{-6}/h} - e^{-t*47.10^{-6}/h}$$

$$MTTF = \int_0^\infty R(t)dt$$

$$= \frac{1}{\lambda_{E1}+\lambda_{A1}} + \frac{1}{\lambda_{E2}+\lambda_{A1}} - \frac{1}{\lambda_{E1}+\lambda_{E2}+\lambda_{A1}} = \left(\frac{1}{26} + \frac{1}{22} - \frac{1}{47}\right) * 10^6 h$$

7 UML diagrams

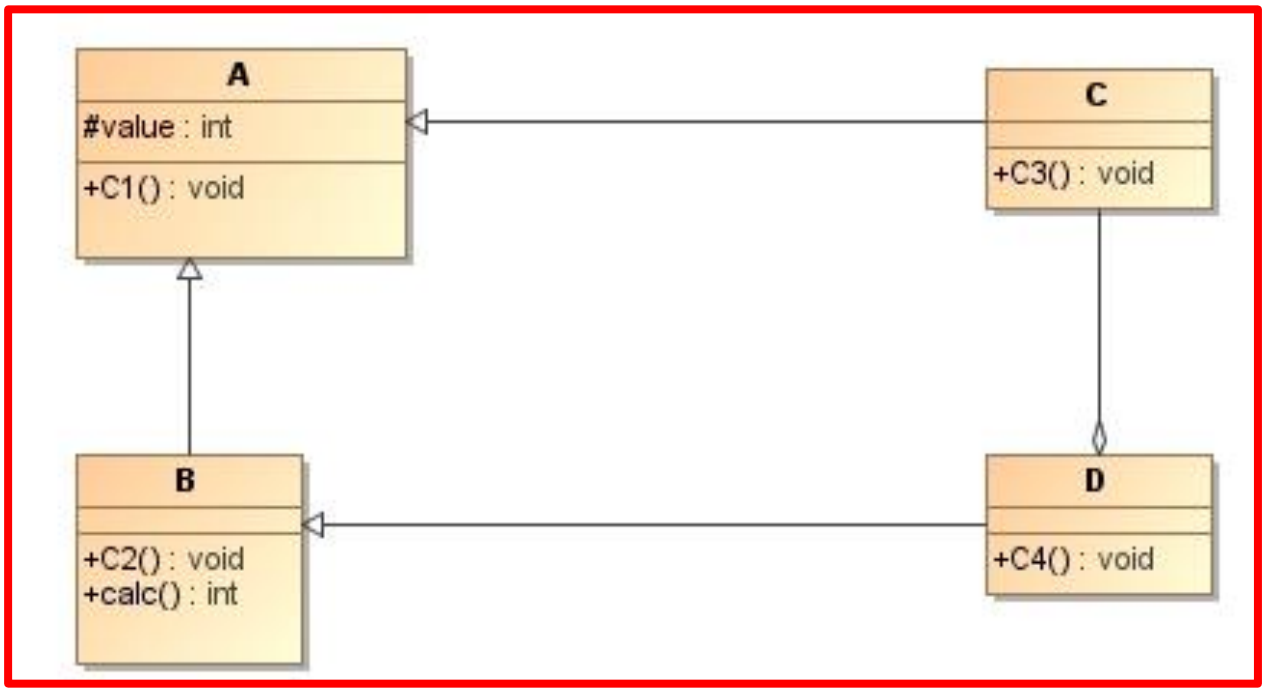
```
class A {
protected:
  int value;
public:
  void C1()  {
    cout << "C1()" << endl;
  }
};
class B : public A {
public:
  void C2()  {
    cout << "C2()" << endl;
  }
  int calc() {
    return value * 10;
  }
};
class C : public A {
public:
  void C3()  {
    cout << "C3()" << endl;
  }
};
class D : public B {
private:
  C c2;
public:
  void C4()  {
    cout << "C4()" << endl;
  }
};

int main () {
  D d;
  return 0;
}
```

Figure 4: C++ software code

/5

7.a Class diagram

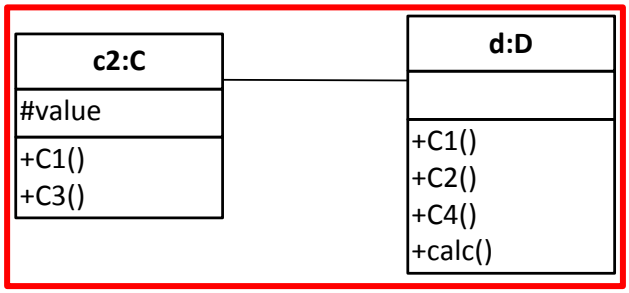
Draw the corresponding class diagram of the C++ software code given in Figure 4. The class diagram should include as many information as possible like classes, attributes, operations and associations.



/4

7.b Object diagram

Draw the corresponding object diagram of the C++ software code given in Figure 4 with all possible information.



/6

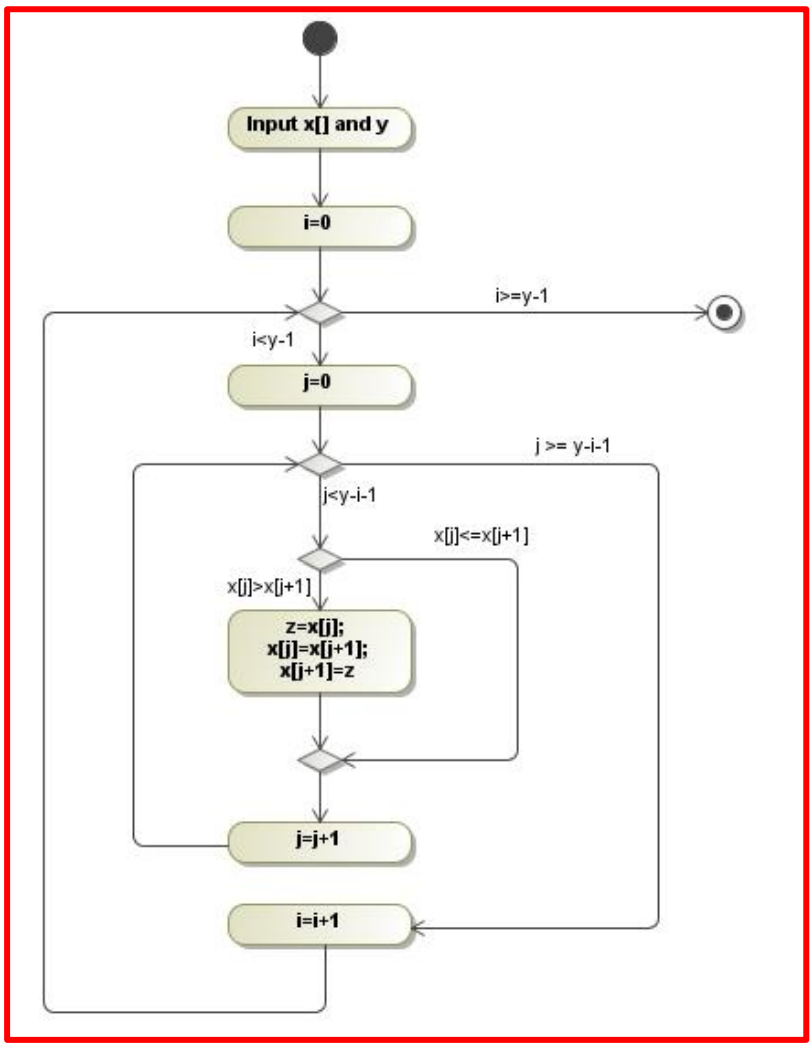
7.c UML activity diagram

a) Analyze the program given below in Figure 5 by drawing an UML activity diagram.

b) Which task does the function *func()* fulfill?

```
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;
      }
    }
  }
}
```

Figure 5: C++ software code



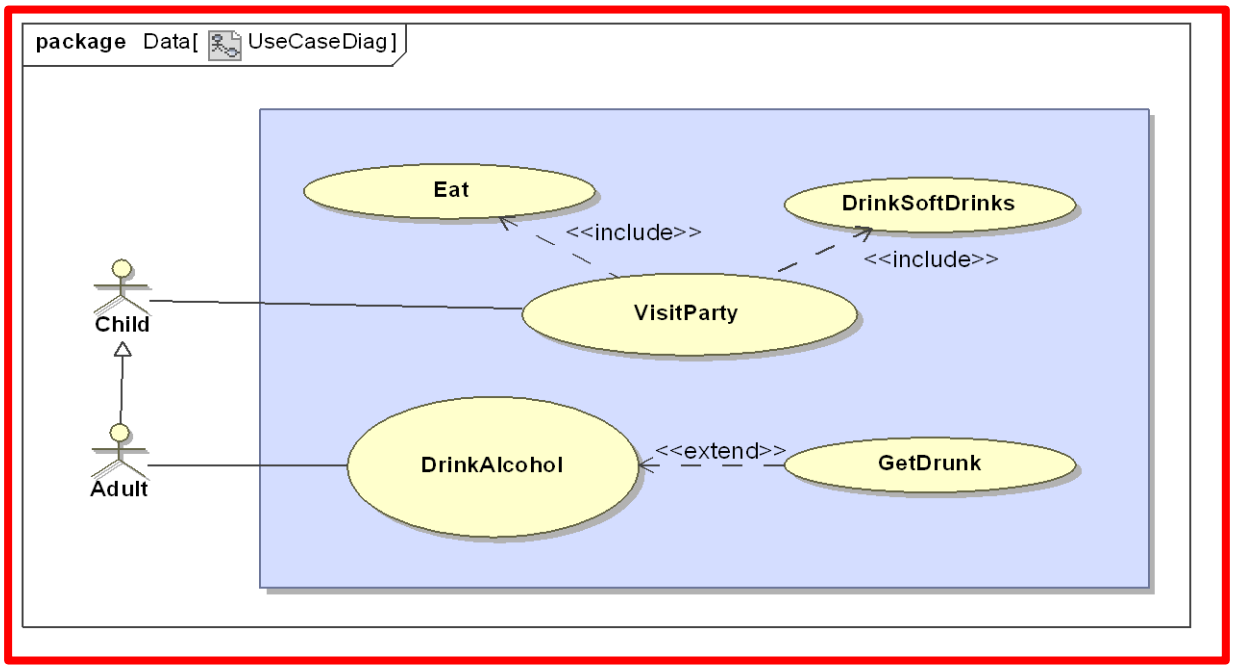
Sort numbers in an array y[] using Bubblesort algorithm or sorting numbers

/4

7.d UML use case diagram

Develop a use case diagram based on the following description of visiting a party.

"A child can visit a party. Eating and drinking soft drinks are the things a child does on a party. On a party, an adult can do everything a child can. In addition to a child, the adult can drink alcohol. Drinking alcohol may result in getting drunk, but it doesn't need to."



8 Nassi Shneiderman diagrams

/4

8.a Nassi Shneiderman diagram

Build a Nassi-Shneiderman diagram from the pseudo code program stated below.

```
float doSomething(float x, int r) {
  if (x == 0) return 1;
  float v = 1;
  float w = 1;

  x *= -x;
  if (r<1) {
    r=10;
  }
  int i=1;
  do{

    w = w*x;

    w = w/(2*i-1);

    w = w/(2*i);
    v += w;
    ++i;
  } while ( i<=r );
  return v;
}
```