Seminar 6
week 6 (4-8 November 2024)

A. **Discussion of the implementation for the lab assignment A3.** Regarding the View part we discuss how it would be possible to call many times the execution of the same example.

B. Exercise 1

Given the following four classes in Java:
class A {...} class B extends A {...} class C extends A {...}

```
class Amain{

A  method1(ArrayList<...> list) {  if list.isEmpty() return null;
                          else return list.get(1);}

void method2(ArrayList<...>  list) {  list.add(null);}

void method3(){
ArrayList<A> listA=new ArrayList<A>(); listA.add(new A());
ArrayList<B> listB = new ArrayList<B>(); listB.add(new B());
ArrayList<C> listC = new ArrayList<C>(); listC.add(new C());
this.method1(listA); this.method1(listB); this.method1(listC);
this.method2(listA); this.method2(listB); this.method2(listC);
}
}
```

Compute the most specific signatures for the class Amain methods (method1 and method2) such that the entire program is correct.
If it is not possible to find the types justify your answer.
Discuss line by line the correctness of the above program.

C. Exercise 2

Given the following four classes in Java:

```
class A {...}  class B extends A {...}  class C extends B {...}

class Amain{

A  method1(ArrayList<...> list) {  if list.isEmpty() return null;
                          else return list.get(1);}

void method2(ArrayList<...>  list, C elem) {  list.add(elem);}

void method3(C elem){
ArrayList<A> listA=new ArrayList<A>(); listA.add(new A());
ArrayList<B> listB = new ArrayList<B>(); listB.add(new B());
ArrayList<C> listC = new ArrayList<C>(); listC.add(new C());
```

```
this.method1(listA); this.method1(listB); this.method1(listC);
this.method2(listA,elem); this.method2(listB,elem); this.method2(listC,elem);
}
}
```

Compute the most specific signatures for the class Amain methods (method1 and method2) such that the entire program is correct.
If it is not possible to find the types justify your answer.
Discuss line by line the correctness of the above program.

B. **Discussion of the following IO classess usage:** FileReader, FileWriter, BufferedReader, BufferedWriter, StreamTokenizer, Scanner and PrintStream. Some code templates of using these classes are given below:

- **FileReader class example:**

```
try(FileReader fileReader = new FileReader("c:\\data\\text.txt")){
int data = fileReader.read();
     while(data != -1) { // read a char
       System.out.print((char) data));
       data = fileReader.read();
     }
}
```

- **FileWriter class example:**

```
try(FileWriter fileWriter = new FileWriter("data\\filewriter.txt",true)){
        //true –appends, false or nothing-overwrites
fileWriter.write("data 1");
fileWriter.write("data 2");
fileWriter.write("data 3");
}
```

- **BufferedReader class example:**

```
Reader reader = new FileReader("data.bin");
try(BufferedReader bufferedReader =new BufferedReader(reader)){
   String line = bufferedReader.readLine();
   while(line != null) {
      //do something with line

      line = bufferedReader.readLine();
   }
}
```

or

```
br=new BufferedReader(new FileReader(numefis));
      String linie;
      while((linie=br.readLine())!=null){
         String[] elems=linie.split("[|]");
         if (elems.length<2){
            System.err.println("Linie invalida "+linie);
            continue;}
```

```
        //do something with the line
    }
```

- **BuferredWriter class example:**

```
FileWriter output = new FileWriter("data.bin");
try(BufferedWriter bufferedWriter = new BufferedWriter(output)){
   for(i=0;i<100;i++){
          bufferedWriter.write("Hello World");
          bufferedWriter.newLine();
          if(i%5==0)
             bufferedWriter.flush();
   }
}
```

- **StreamTokenizer class example:**

```
Reader reader = new FileReader("data.bin");
try(StreamTokenizer streamTokenizer = new StreamTokenizer(reader)){

   while(streamTokenizer.nextToken() != StreamTokenizer.TT_EOF){

      if(streamTokenizer.ttype == StreamTokenizer.TT_WORD) {
         System.out.println(streamTokenizer.sval);
      } else if(streamTokenizer.ttype == StreamTokenizer.TT_NUMBER) {
         System.out.println(streamTokenizer.nval);
      } else if(streamTokenizer.ttype == StreamTokenizer.TT_EOL) {
         System.out.println();
      }
   }
}
```

- **PrintWriter class example:**

```
        FileWriter  writer      = new FileWriter("report.txt");
        PrintWriter printWriter = new PrintWriter(writer);
        printWriter.print(true);
        printWriter.print((int) 123);
        printWriter.print((float) 123.456);
        intVar i=200;
        printWriter.printf("Text + data: %d", intVar);
        printWriter.close();
```

- **Scanner class examples:**

```
Scanner sc = new Scanner(new File("myNumbers"));
    while (sc.hasNextLong()) {
       long aLong = sc.nextLong();
    }
```