Programming Language Design 2024
Modules and Object-Oriented Programming

Hans Hüttel

4 March 2024

1. *(20 minutes)* Find a programming language that allows *nested modules*, and show an example of a declaration of a nested module in this language.

2. *(25 minutes)* Consider this Java program:

```
        class A {
        void f () {System.out.println("A");}
        void g () {this.f ();}
}
      class B extends A {
        void f () {System.out.println("B");}
}
      public class Foo {
        public static void  main (String [] args) {
B b = new B();
b.g (); }
}
```

   What does this program output, and why? Use the pointer model to justify your answer.

   If we change the second line to

```
      private void f () {System.out.println("A");}
```

   what will happen then, and why?

3. *(25 minutes)* We can think of the new C constructor in Java as a method that is invoked at the level of the *class* C as opposed to at the level of a single instance of it. This might be a useful idea.

   In Java, we cannot create a function that returns the list of all objects of a given class. Sometimes it would be nice to be able to do this, though. Suggest an extension of Java that would make this possible by drawing an analogy from the example with new.

   Could this have other uses as well? And are there any disadvantages?

4. *(25 minutes)* In Smalltalk there are no control structures! Biconditionals and while loops do not exist, and nor does assignment. But it is still possible to write useful programs. How would Smalltalk programmers express the following simple piece of code which we can easily write in imperative languages such as C or Java? Why and how is the Smalltalk approach different?

```
i=7;
sum=484000;
while i > 0 do
{ i = i -1;
  if (i mod 2) = 0
  then sum = sum + i
  else sum = sum + 17}
```