INF3110

Problems week 5.10 - 9.10, 2009

Problem 1

In the following Java-like program we use redefinition (overriding) of a virtual method change. Note that the class A is defined locally to Circle, and that the subclass B is defined locally to the method fromAtoB. The call of the method super.change() implies call of the method as it is defined in the superclass, not as redefined in B.
The program execution is started by execution of the main method in the class Program.

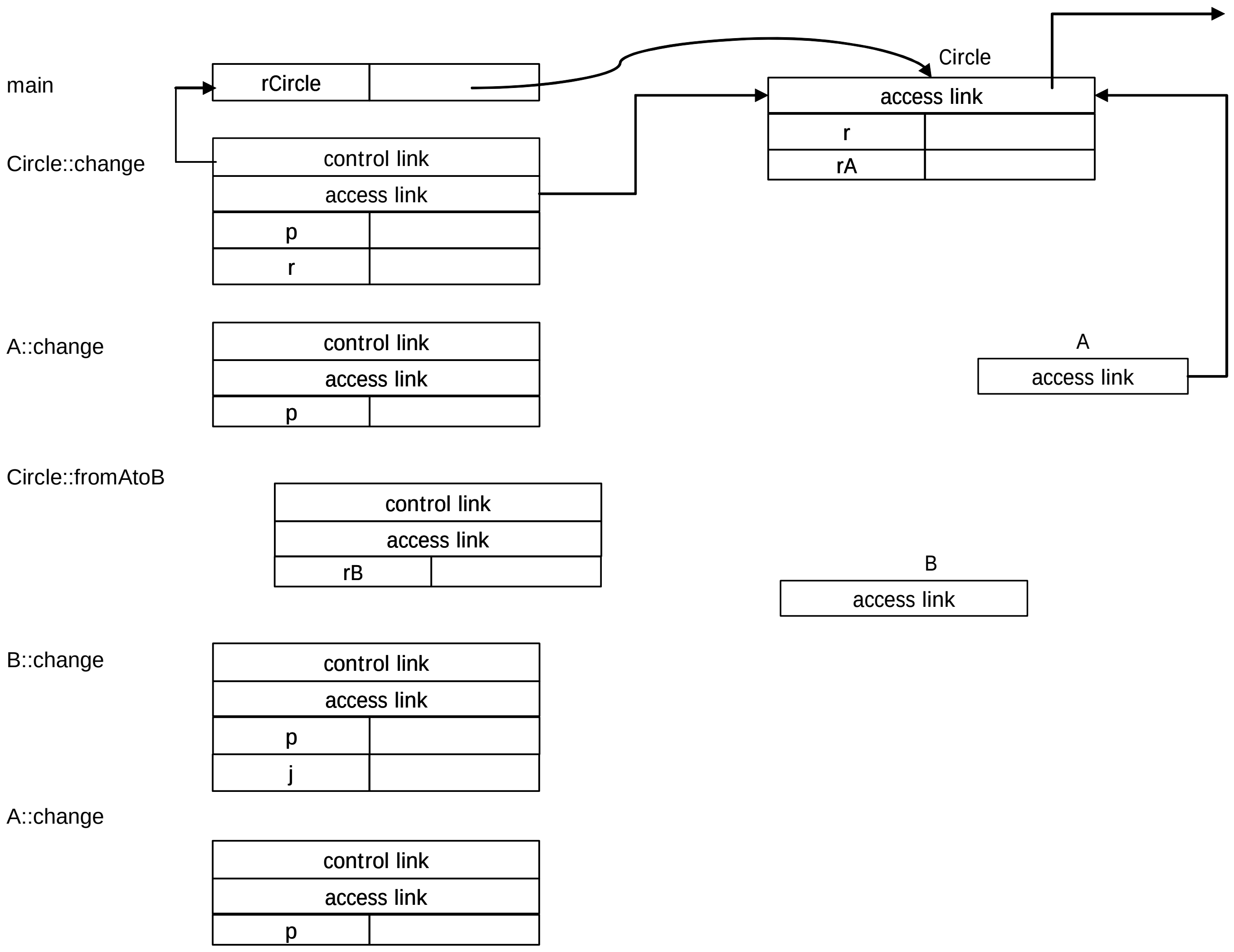
```
class Program {
  public static void main(String[] args) {
    Circle rCircle= new Circle(); rCircle.change(2);
  }
}

class Circle {
  int r;
  Circle() {r = 1;}
  class A {
    void change(int p){r = r + p;}
  }
  A rA = new A();
  void change(int p){
    int r = p;
    rA.change(p);
    fromAtoB();
    rA.change(p);     //        (*)
  }
  void fromAtoB(){
    class B extends A {
      void change(int p){
        int j = r;
        super.change(p);
        r = r + j;
      }
    }
    B rB = new B();
    rA = rB;
  }
}
```

We assume that the language has static scoping. Put in values on parameters, references, and variables, access links and control links in the activation blocks and in objects as they are when the activation record for the call rA.change(p) in the line marked with (*) is on top of the stack. Put in values and links as they are just before the change method exits. The figure below contains activation blocks that may be on the stack (which means that they will not all be there). Some of the links and references are put in already. Assume that there has been no garbage collection.

At the bottom of the stack we have as usual the activation block for main.

The access link for an object denotes the instance that represents the textually enclosing block (that is object for an enclosing class or activation block for an enclosing method). The classes A and B are so simple that A and B objects only consist of an access link.



Problem 2

The following procedure Jensen has three by-name parameters:

```
begin
  integer procedure Jensen(x, i, n);
    name x, i, n; integer x, i, n;
  begin
    integer index, sum;
    for index := 1 step 1 until n do
      begin
        i := index;
        sum := sum + x;
     end;
    Jensen := sum;
  end Jensen;
  integer ix, res1, res2, res3; integer array a(1:5);
  a(1) := 7; a(2) := -1; a(3) := 11; a(4) := 8; a(5) := 4;

  res1 := Jensen(ix*ix, ix, 10);
  res2 := Jensen(a(ix), ix, 5);
  res3 := Jensen(if Rem(a(ix),2)<>0 then 1 else 0, ix, 5);
end
```

What are evaluated by the three calls of Jensen? Rem(i,j) gives the remainder of i divided by j, e.g. Rem(5,2)=1 and Rem(-5,2)= -1. The sign of the result is the same as the sign of the dividend (i), independently of the sign of the divisor (j). '<>' means 'not equal'. Integer variables have default value 0.

Problem 3

The following program has a function as parameter.

```
{ int i, j;
  void a(){
    int k; ...
  };
  void b(void function f){
    int i, j, k;
    void c(){
      ... k = ...
     };
    f();
    b(c);
    ...
  }
  main(){
    b(a);
  }
}
```

Draw the run-time stack as it is when the f is called within b the second time.

Problem 4
Exercise 7.12 in Mitchell

Problem 5
Can the L-value of a variable be accessed only when its name is visible (i.e. within scope)? If YES, why, and if NO, why and how?

Problem 6
Parameters to procedures are often used in order to parameterize the computation, so that procedures called with different actual values perform different computations.

In which cases will a parameterless procedure not perform the same computation every time it is called?

Problem 7
By-reference and by-value-result have in most cases the same effect. Consider this small example:

```
int x;
void p(int i) {
  i=i+1;
  x=x+1;
};
x=1;
p(x);
```

Will the call p(x) have the same or different effect when the parameter i is by-reference and by-value-result?

Problem 8
It was indicated at the lecture that functions as parameters and name parameters are similar in that the actual parameters have to maintain their environment. Indicate a way in which some of the properties of name parameters can be achieved by means of functions as parameters. Which property cannot be achieved in this way.

Problem 9
a) Java does not have call-by-reference parameters, while C# has. How would you in Java get the effect of p(a), where a is a variable and the formal parameter is a call-by-reference parameter?

b) Java does not have call by value result parameters. How would you in Java get the effect of p(a), where a is a variable and the formal parameter is a call-by-value-result parameter.

c) What about p(a,b), where both are call-by-value-result parameters?