

Problemas de PRED – Control de datos y Tipos

1. Dada la siguiente acción, donde A es una variable global visible en P:

```
acción P (X:^entero)
X^ := X^+1;
A^:= A^+1;
facción
```

Suponiendo que se produce la llamada P(A), donde A:^entero, y que justo antes de la llamada se había ejecutado la instrucción $A^:=2$, ¿cuanto valdrá A después de la ejecución de P si el paso de parámetros se produce por valor? ¿y si se produce por valor-resultado? ¿y si se produce por referencia?

2. Supongamos que tenemos el siguiente fragmento de programa:

```
var p,q : ^entero;
/* p y q son punteros o referencias a enteros */
new(p);
q := p;
p^ := 1; /* asignamos 1 al campo apuntado por p */
q^ := p^ + 1;
escribir(p^);
free(p);
... /* en este punto iría un número indeterminado de
    líneas de código que no usan ni a p ni a q */
escribir(q^)
```

¿Cual será el valor escrito por la instrucción escribir(p^)? ¿y por la instrucción escribir(q^)?

3. Supongamos que tenemos definida una acción con la siguiente cabecera:

```
acción P(ent/sal x, y : entero)
<cuerpo de P>
facción
```

y supongamos que sabemos que P funciona bien si el paso de parámetros se produce por valor/resultado, pero no si se realiza por referencia. Lamentablemente, en la implementación de la que disponemos el compilador implementa el paso de parámetros por referencia. ¿Cómo podríamos modificar fácilmente P para que funcionara bien en este caso?

4. Supongamos que tenemos las siguientes declaraciones:

```
class C
  x: entero; /* x es un campo o atributo de C */
  acción P(y: entero)
  ...
  facción
fclass
```

```

class C1 subclase de C
    x1: entero; /* x1 es un campo o atributo de C1 */
    acción P1(y:entero)
    ...
    facción
fclase

class C2 subclase de C1
    x2: entero; /* x2 es un campo o atributo de C2 */
fclase

class C3 subclase de C
    acción P(y:entero)
    ...
    facción
fclase

```

¿Hay algo erróneo en estas declaraciones?. Si es así, suprimidlo. Supongamos, además, que tenemos declaradas las variables

```
a: C; a1: C1; a2: C2; a3: C3
```

y que v, v1, v2, v3 son, respectivamente, valores de las clases C, C1, C2 y C3. ¿Alguna de las asignaciones de la siguiente serie contiene un error de tipos?

```

a1:= v2;
a:= v3;
a2:= v;
a3:= a;
a2:= v2;
a:= a3

```

Supongamos ahora que después de dicha serie de asignaciones (una vez suprimidas las asignaciones incorrectas) tenemos las siguientes instrucciones:

```

a1.x1:= 3;
a1.x2:= 5;
a2.x2:= 4;
a2.x:= 7;
a2.P(7);
a3.P(7);
a.P(7);

```

¿Contiene alguna un error de tipos? Si las dos últimas son correctas, en cada caso ¿a qué acción P se llamaría? ¿a la de la clase C? o ¿a la de C3?.

5. Supongamos que las declaraciones de clase del ejercicio anterior definen como `public` a las acciones P, P1 y al atributo x1; como `protected` al atributo x; y como `private` al atributo x2. Supongamos también que todas las declaraciones de esas clases están en un mismo

package y que la acción principal (`main`) está en otro package distinto que usa al anterior. Por último, supongamos que las reglas de visibilidad que tenemos son las de Java. En este contexto:

- a.) Si las instrucciones que aparecen en el ejercicio anterior estuvieran en la acción principal, ¿cuales serían erróneas?
- b.) ¿Puede contener el cuerpo de P1 las asignaciones siguientes?

```
this.x:= 3;  
this.x1:= 5;  
this.x2:= 6;
```

- c.) Y ¿cuales de esas asignaciones podrían estar en el cuerpo de de la acción P de la clase C? ¿Y en la acción P de la clase C3?
- d.) ¿Puede contener el cuerpo de P1 las llamadas siguientes?

```
this.P(5);  
this.P1(7);
```

- e.) ¿Las puede contener el cuerpo de la acción P de la clase C? ¿Y la acción P de la clase C3?