## Sample exam 3

The INFDEV team

## 1 Question 1

Given the following block of code, fill in the stack, heap, and PC with all the steps taken by the program at runtime.

- Points: 4 (50% of total).
- Grading: one point per correctly filled-in execution step.
- Associated learning objective: abstraction.

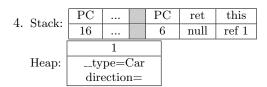
```
interface MovableObject {
   2
   3
      class Car : MovableObject {
       private float direction = 3.14;
   5
       public Car() {
       static public void move(Car car,float direction) {
   8
          car.direction = direction;
   9
       }
  10
  11
      class Particle : MovableObject {
  12
       private float direction = 0;
       public Particle() {
  13
       }
  14
\sim 15
  16
     MovableObject mo = new Car();
  17
     Car.move(mo,1);
```

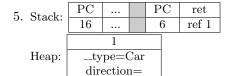


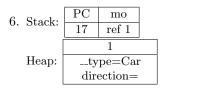
- 2. Stack: PC 16
- 3. Stack: PC 16

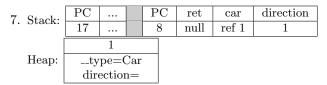
  Heap: 1

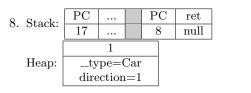
  --type=Car direction=

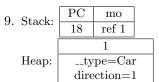












## 2 Question 2

Given the following block of code, fill in the declarations, class definitions, and PC with all steps taken by the compiler while type checking.

- Points: 4 (50% of total).
- Grading: one point per correctly filled-in type checking step.
- $\bullet$  Associated learning objective:  $type\ checking.$

```
interface IntList {
  bool isEmpty();
  int getValue();
}
class IntNode : IntList {
  private int value;
  private IntList tail;
```

ಲ

```
public IntNode(int value,IntList tail) {
9
       this.value = value;
10
       this.tail = tail;
11
     }
     public bool isEmpty() {
12
13
       return false;
14
15
     public int getValue() {
       return this.value;
16
17
     }
18
   class IntEmpty : IntList {
19
     public IntEmpty() {
20
21
     }
22
     public bool isEmpty() {
23
       return true;
24
     }
25
     public int getValue() {
26
       return 0;
27
     }
28
   IntList list = new IntNode(5, new IntEmpty());
```

1. Declarations:  $\begin{array}{|c|c|c|}\hline PC \\ \hline 1 \\ \hline \end{array}$ 

4

2. Declarations:  $\frac{PC}{5}$ 

3. Declarations: PC 18

Classes:  $\begin{array}{|c|c|c|c|} \hline & IntList & IntNode \\ \hline & & IntNode = (IntNode \times int \times IntList) \rightarrow IntNode \\ \hline & getValue = IntList \rightarrow int \\ isEmpty = IntList \rightarrow bool \\ \hline & tail = IntList \\ \hline & value = int \\ \hline \end{array}$ 

28

	IntEmpty	IntList	IntNode
Classes:		$getValue=IntList \rightarrow int$ $isEmpty=IntList \rightarrow bool$	$IntNode = (IntNode \times int \times IntList) \rightarrow IntNode$
	$\begin{split} & \text{IntEmpty=IntEmpty} \rightarrow \text{IntEmpty} \\ & \text{getValue=IntList} \rightarrow \text{int} \\ & \text{isEmpty=IntList} \rightarrow \text{bool} \end{split}$		$getValue=IntList \rightarrow int$
			$isEmpty=IntList \rightarrow bool$
			tail=IntList
			value=int

5. Declarations:

30

	IntEmpty	IntList	IntNode
Classes:		$getValue=IntList \rightarrow int \\ isEmpty=IntList \rightarrow bool$	$\begin{split} IntNode = & (IntNode \times int \times IntList) \rightarrow IntNode \\ & getValue = IntList \rightarrow int \\ & isEmpty = IntList \rightarrow bool \\ & tail = IntList \\ & value = int \end{split}$

6. Declarations:

PC	list
31	IntList

Classes:

IntEmpty	IntList	IntNode
$\begin{split} & IntEmpty{=}IntEmpty \rightarrow IntEmpty \\ & getValue{=}IntList \rightarrow int \\ & isEmpty{=}IntList \rightarrow bool \end{split}$	$getValue=IntList \rightarrow int$ $isEmpty=IntList \rightarrow bool$	$\begin{split} & IntNode = (IntNode \times int \times IntList) \rightarrow IntNode \\ & getValue = IntList \rightarrow int \\ & isEmpty = IntList \rightarrow bool \\ & tail = IntList \\ & value = int \end{split}$