

Accueil ► SI - Sciences Informatiques ► SI3 ► Intro POO ► Stuff to do - unevaluated ► Monitoring animal populations

<b>Commencé le</b>	jeudi 26 octobre 2017, 10:06
<b>État</b>	Terminé
<b>Terminé le</b>	jeudi 26 octobre 2017, 11:51
<b>Temps mis</b>	1 heure 44 min
<b>En retard</b>	1 heure 44 min
<b>Points</b>	16,00/16,00
<b>Note</b>	<b>20,00</b> sur 20,00 ( <b>100%</b> )

**Description**

First we'll test your procedural code (in the package `animalmonitoring.v1`). The methods tested are:

- `addSightings`
- `getAnimals`
- `printSightingsOf`
- `getCount`
- `printEndangered`

**Question 1**

Correct

Note de 1,00 sur  
1,00

Tests your addSightings and getAnimals methods.

Paste your v1.AnimalMonitor class into the Answer space.

**For example:**

Test	Result
<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); Set&lt;String&gt; animals = monitor.getAnimals(); System.out.println(animals.size());</pre>	4

Réponse:

```
1 package animalmonitoring.v1;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5 import java.util.List;
6 import java.util.Set;
7 import java.util.HashSet;
8
9 /**
10  * Monitor counts of different types of animal.
11  * Sightings are recorded by spotters.
12  *
13  * @author David J. Barnes and Michael Kölling
14  * @version 2016.02.29 (imperative)
15  */
16 public class AnimalMonitor {
17     // Records of all the sightings of animals.
18     private List<Sighting> sightings = new ArrayList<>();
19 }
```

Vérifier

	Test	Expected	Got	
✓	<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); Set&lt;String&gt; animals = monitor.getAnimals(); System.out.println(animals.size());</pre>	4	4	✓

Passed all tests! ✓

**Correct**

Note pour cet envoi : 1,00/1,00.

**Question 2**

Correct

Note de 1,00 sur  
1,00

Tests printSightingsOf. Paste your v1.AnimalMonitor into the Answer space.

**For example:**

Test	Result
<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); Set&lt;String&gt; animals = monitor.getAnimals(); monitor.printSightingsOf("Arceus");</pre>	<pre>Arceus, count = 0, area = 1, s Arceus, count = 24, area = 2,</pre>

Réponse:

```
1 package animalmonitoring.v1;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5 import java.util.List;
6 import java.util.Set;
7 import java.util.HashSet;
8
9 /**
10  * Monitor counts of different types of animal.
11  * Sightings are recorded by spotters.
12  *
13  * @author David J. Barnes and Michael Kölling
14  * @version 2016.02.29 (imperative)
15  */
16 public class AnimalMonitor {
17     // Records of all the sightings of animals.
18     private List<Sighting> sightings = new ArrayList<>();
19 }
```

Vérifier

	Test	Expected
✓	<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); Set&lt;String&gt; animals = monitor.getAnimals(); monitor.printSightingsOf("Arceus");</pre>	<pre>Arceus, count = 0, area = 1, sp Arceus, count = 24, area = 2, s</pre>

Passed all tests! ✓

**Correct**

Note pour cet envoi : 1,00/1,00.

**Question 3**

Correct

Note de 1,00 sur  
1,00

Copy your `animalmonitoring.v1.AnimalMonitor` class into the Answer box and Check your work.

For example:

Test	Result
<code>System.out.println();</code>	Code looks clean

Réponse:

```
1 package animalmonitoring.v1;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5 import java.util.List;
6 import java.util.Set;
7 import java.util.HashSet;
8
9 /**
10  * Monitor counts of different types of animal.
11  * Sightings are recorded by spotters.
12  *
13  * @author David J. Barnes and Michael Kölling
14  * @version 2016.02.29 (imperative)
15  */
16 public class AnimalMonitor {
17     // Records of all the sightings of animals.
18     private List<Sighting> sightings = new ArrayList<>();
19 }
```

Vérifier

	Test	Expected	Got	
✓	<code>System.out.println();</code>	Code looks clean	Code looks clean	✓

Passed all tests! ✓

**Correct**

Note pour cet envoi : 1,00/1,00.

**Description**

Now we'll test the functional version (`animalmonitoring.v2`).

**Question 4**

Correct

Note de 2,00 sur  
2,00

Checks code for procedural style loops: for, while. Instead of loops, your code should be in a functional style using lambdas, pipelines etc.

Paste your `v2.AnimalMonitor` class into the Answer box.

**For example:**

Test	Result
<code>System.out.println()</code>	No procedural style loops

Réponse:

```
1 package animalmonitoring.v1;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5 import java.util.List;
6 import java.util.Set;
7 import java.util.HashSet;
8
9 /**
10  * Monitor counts of different types of animal.
11  * Sightings are recorded by spotters.
12  *
13  * @author David J. Barnes and Michael Kölling
14  * @version 2016.02.29 (imperative)
15  */
16 public class AnimalMonitor {
17     // Records of all the sightings of animals.
18     private List<Sighting> sightings = new ArrayList<>();
19 }
```

Vérifier

	Test	Expected	Got	
✓	<code>System.out.println()</code>	No procedural style loops	No procedural style loops	✓

Passed all tests! ✓

**Correct**

Note pour cet envoi : 2,00/2,00.

**Question 5**

Correct

Note de 2,00 sur  
2,00

Checks code for functional style lambdas; instead of loops, your code should be in a functional style using lambdas, pipelines etc.

Paste your `v2.AnimalMonitor` class into the Answer box.

Réponse:

```
1 package animalmonitoring.v1;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5 import java.util.List;
6 import java.util.Set;
7 import java.util.HashSet;
8
9 /**
10  * Monitor counts of different types of animal.
11  * Sightings are recorded by spotters.
12  *
13  * @author David J. Barnes and Michael Kölling
14  * @version 2016.02.29 (imperative)
15  */
16 public class AnimalMonitor {
17     // Records of all the sightings of animals.
18     private List<Sighting> sightings = new ArrayList<>();
19 }
```

Vérifier

	Test	Expected	Got	
✓	System.out.println()	Uses functional style lambdas	Uses functional style lambdas	✓

Passed all tests! ✓

**Correct**

Note pour cet envoi : 2,00/2,00.

**Question 6**

Correct

Note de 2,00 sur  
2,00Tests your `addSightings` and `getAnimals` methods.Paste your `v2.AnimalMonitor` class into the Answer space.**For example:**

Test	Result
<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); Set&lt;String&gt; animals = monitor.getAnimals(); System.out.println(animals.size());</pre>	4

Réponse:

```
1 package animalmonitoring.v1;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5 import java.util.List;
6 import java.util.Set;
7 import java.util.HashSet;
8
9 /**
10  * Monitor counts of different types of animal.
11  * Sightings are recorded by spotters.
12  *
13  * @author David J. Barnes and Michael Kölling
14  * @version 2016.02.29 (imperative)
15  */
16 public class AnimalMonitor {
17     // Records of all the sightings of animals.
18     private List<Sighting> sightings = new ArrayList<>();
19 }
```

Vérifier

	Test	Expected	Got	
✓	<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); Set&lt;String&gt; animals = monitor.getAnimals(); System.out.println(animals.size());</pre>	4	4	✓

Passed all tests! ✓

**Correct**

Note pour cet envoi : 2,00/2,00.

**Question 7**

Correct

Note de 2,00 sur  
2,00

Tests printSightingsOf. Paste your v2.AnimalMonitor into the Answer space.

**For example:**

Test	Result
<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); Set&lt;String&gt; animals = monitor.getAnimals(); monitor.printSightingsOf("Greninja");</pre>	<pre>Greninja, count = 10, area = 1 Greninja, count = 2, area = 1, Greninja, count = 16, area = 1 Greninja, count = 0, area = 2,</pre>

Réponse:

```
1 package animalmonitoring.v1;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5 import java.util.List;
6 import java.util.Set;
7 import java.util.HashSet;
8
9 /**
10  * Monitor counts of different types of animal.
11  * Sightings are recorded by spotters.
12  *
13  * @author David J. Barnes and Michael Kölling
14  * @version 2016.02.29 (imperative)
15  */
16 public class AnimalMonitor {
17     // Records of all the sightings of animals.
18     private List<Sighting> sightings = new ArrayList<>();
19 }
```

Vérifier

	Test	Expected
✓	<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); Set&lt;String&gt; animals = monitor.getAnimals(); monitor.printSightingsOf("Greninja");</pre>	<pre>Greninja, count = 10, area = 1, Greninja, count = 2, area = 1, Greninja, count = 16, area = 1, Greninja, count = 0, area = 2,</pre>
✓	<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); Set&lt;String&gt; animals = monitor.getAnimals(); monitor.printSightingsOf("Sander");</pre>	

Passed all tests! ✓

**Correct**

Note pour cet envoi : 2,00/2,00.



**Question 8**

Correct

Note de 2,00 sur  
2,00Tests `getCount`. Paste your `v2.AnimalMonitor` into the Answer space.**For example:**

Test	Result
<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); System.out.println(monitor.getCount("Greninja"));</pre>	28

Réponse:

```
1 package animalmonitoring.v1;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5 import java.util.List;
6 import java.util.Set;
7 import java.util.HashSet;
8
9 /**
10  * Monitor counts of different types of animal.
11  * Sightings are recorded by spotters.
12  *
13  * @author David J. Barnes and Michael Kölling
14  * @version 2016.02.29 (imperative)
15  */
16 public class AnimalMonitor {
17     // Records of all the sightings of animals.
18     private List<Sighting> sightings = new ArrayList<>();
19 }
```

Vérifier

	Test	Expected	Got	
✓	<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); System.out.println(monitor.getCount("Greninja"));</pre>	28	28	✓
✓	<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); System.out.println(monitor.getCount("Mew", 3, 1));</pre>	30	30	✓

Passed all tests! ✓

**Correct**

Note pour cet envoi : 2,00/2,00.

**Question 9**

Correct

Note de 2,00 sur  
2,00

Tests printEndangered. Paste your v2 .AnimalMonitor into the Answer space.

**For example:**

Test	Result
<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); Set&lt;String&gt; animals = monitor.getAnimals(); monitor.printEndangered(new ArrayList&lt;String&gt;(animals), 25);</pre>	<pre>Pikachu is endangered. Arceus is endangered.</pre>

Réponse:

```
1 package animalmonitoring.v1;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5 import java.util.List;
6 import java.util.Set;
7 import java.util.HashSet;
8
9 /**
10  * Monitor counts of different types of animal.
11  * Sightings are recorded by spotters.
12  *
13  * @author David J. Barnes and Michael Kölling
14  * @version 2016.02.29 (imperative)
15  */
16 public class AnimalMonitor {
17     // Records of all the sightings of animals.
18     private List<Sighting> sightings = new ArrayList<>();
19 }
```

Vérifier

	Test	Expected	Got
✓	<pre>AnimalMonitor monitor = new AnimalMonitor(); String sightingsFile = System.getProperty("java.class.path")     + "../../../src/animalmonitoring/v1/sightings.csv"; monitor.addSightings(sightingsFile); Set&lt;String&gt; animals = monitor.getAnimals(); monitor.printEndangered(new ArrayList&lt;String&gt;(animals), 25);</pre>	<pre>Pikachu is endangered. Arceus is endangered.</pre>	<pre>Pikachu Arceus</pre>

Passed all tests! ✓

**Correct**

Note pour cet envoi : 2,00/2,00.

**Question 10**

Correct

Note de 1,00 sur  
1,00Copy your `animalmonitoring.v2.AnimalMonitor` class into the Answer box and Check your work.**For example:**

Test	Result
<code>System.out.println();</code>	Code looks clean

Réponse:

```
1 package animalmonitoring.v1;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5 import java.util.List;
6 import java.util.Set;
7 import java.util.HashSet;
8
9 /**
10  * Monitor counts of different types of animal.
11  * Sightings are recorded by spotters.
12  *
13  * @author David J. Barnes and Michael Kölling
14  * @author Florian S
15  * @version 2016.02.29 (imperative)
16  */
17 public class AnimalMonitor {
18     // Records of all the sightings of animals.
19 }
```

Vérifier

	Test	Expected	Got	
✓	<code>System.out.println();</code>	Code looks clean	Code looks clean	✓

Passed all tests! ✓

**Correct**

Note pour cet envoi : 1,00/1,00.