B2 - Reboost

B-RBT-200

Backend

PHP - Object Oriented Programming

1.1

Backend

repository name:	RBT_back_d03_$ACADEMICYEAR
repository rights:	anthony1.palasse@epitech.eu fabien-luc1.lallemand@epitech.eu kevin.cazal@epitech.eu nicolas1.ah-leung@epitech.eu william.grondin@epitech.eu

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

+ Introduction

Today we will continue to go deeper into OOP, you will keep using all the notions from yesterday and

You will also discover a few new things:

- static keyword
- namespace

Except specified otherwise, all messages must be followed by a newline.

+ Task 01

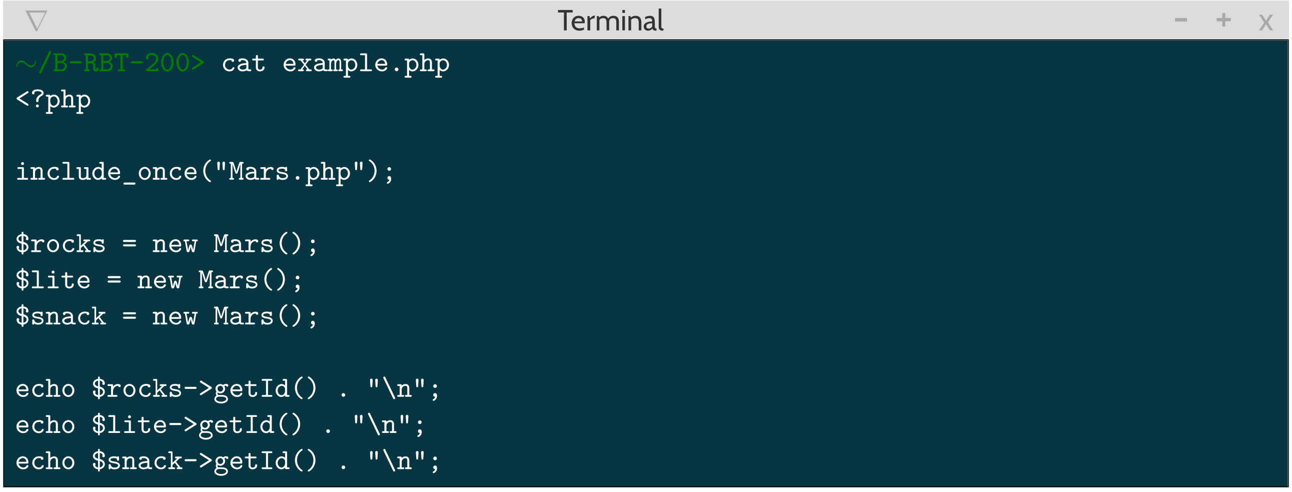
Turn in: task_01/Mars.php
Restrictions: None.

Create a new class `Mars`: it will have an attribute id for which you will create a getter (getId) but no setter.

You must create your class so that the first instance of `Mars` has an id of 0, the second an id of 1…

Static attributes

Example :

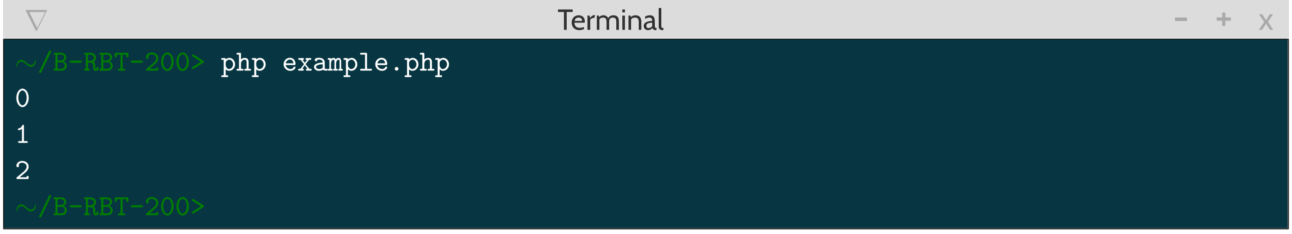


```
~/B-RBT-200> cat example.php
<?php

include_once("Mars.php");

$rocks = new Mars();
$lite = new Mars();
$snack = new Mars();

echo $rocks->getId() . "\n";
echo $lite->getId() . "\n";
echo $snack->getId() . "\n";
```



```
~/B-RBT-200> php example.php
0
1
2
~/B-RBT-200>
```

+ Task 02

Turn in: task_02/Astronaut.php
Restrictions: None.

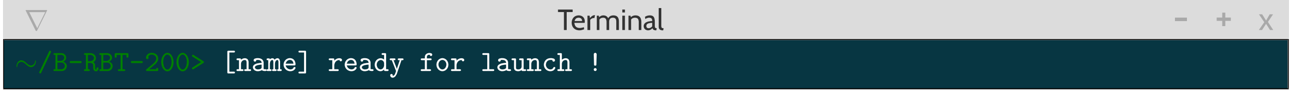
Create a new class `Astronaut` with the following attributes:

- name: a string describing the name of the Astronaut
- snacks: an integer describing the number of snacks our Astronaut possesses
- destination: will be used to store his destination later
- id: an integer describing the id of the Astronaut

The name must be passed during the creation of the Astronaut, it is mandatory, his snack will be initialized to 0 and his destination to null.

The id must be unique and be incremented for each new Astronaut being created (i.e the first Astronaut of our program must have id 0, then the second one will have id 1 etc…)

Moreover, every Astronaut being created will display:

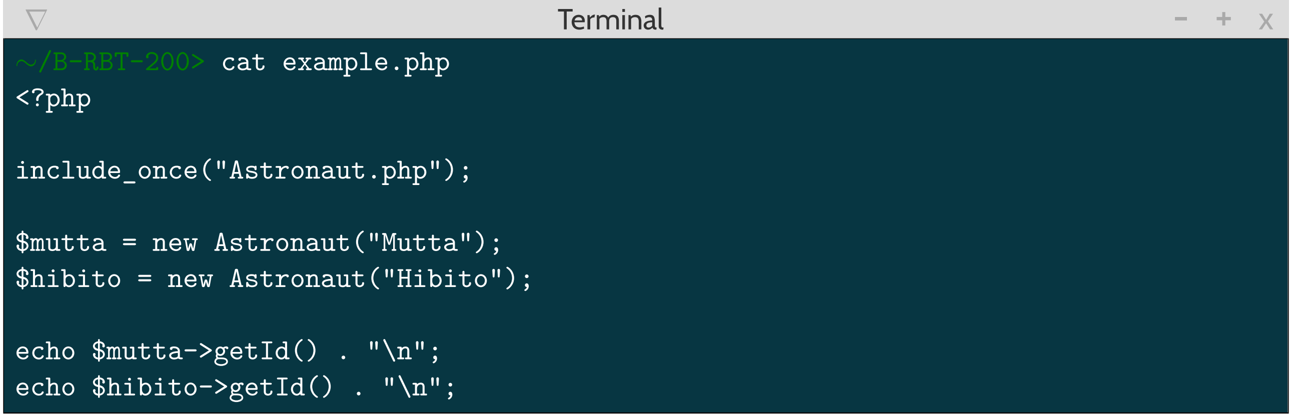


```
~/B-RBT-200> [name] ready for launch !
```

Where `[name]` is the name of the Astronaut.

You must also add a getter for id (`getId`) but no setter, and a getter for destination (`getDestination`) without setter.

Example :

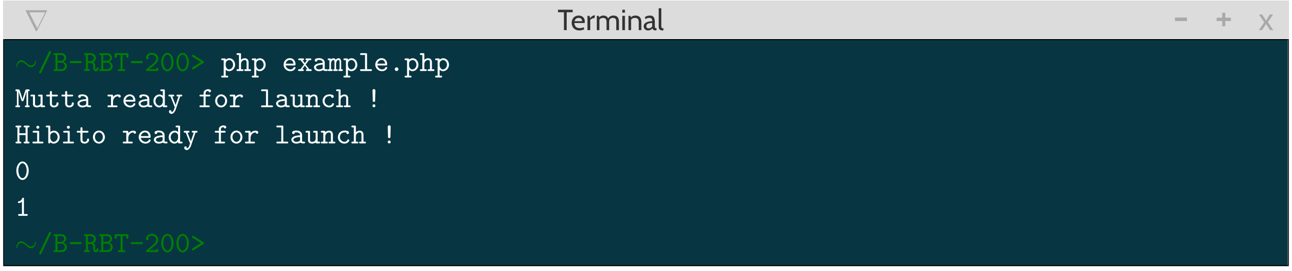


```
~/B-RBT-200> cat example.php
<?php

include_once("Astronaut.php");

$mutta = new Astronaut("Mutta");
$hibito = new Astronaut("Hibito");

echo $mutta->getId() . "\n";
echo $hibito->getId() . "\n";
```



```
~/B-RBT-200> php example.php
Mutta ready for launch !
Hibito ready for launch !
0
1
~/B-RBT-200>
```

+ Task 03

Turn in: task_03/Mars.php
Restrictions: None.

Use your class `Mars` from the first task without changing it.

You must create another class `Mars` representing the planet.

To differentiate the two Mars, you'll put the first one (the one from Task 1) in a namespace called `chocolate` and the second one in a namespace called `planet`.

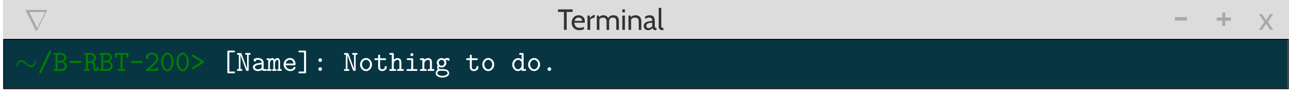
> You should turn in only one file Mars.php, all your code must be in this file.

You will also add a private attribute `size` to this new `Mars` class, with a getter and a setter (getSize and set-Size).

It must be possible to specify the size of the planet when creating one thanks to its constructor, but it shouldn't be mandatory.

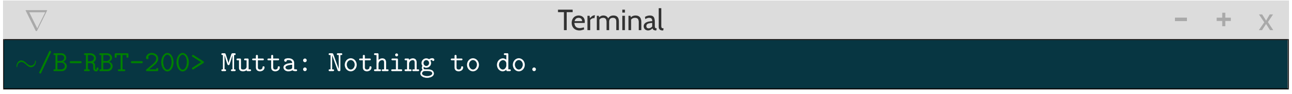
+ Task 04

Turn in: task_04/Astronaut.php
task_04/Mars.php
Restrictions: In this Task every time we ask for `[Name]` to be displayed it should be replaced by the name of the Astronaut for example:



```
~/B-RBT-200> [Name]: Nothing to do.
```

Will in the end becomes:



```
~/B-RBT-200> Mutta: Nothing to do.
```

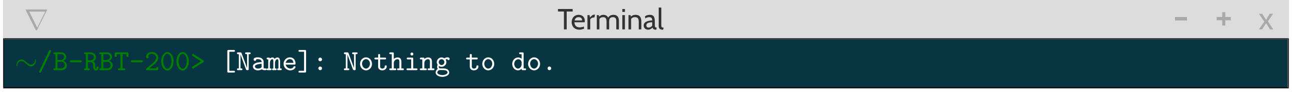
If our Astronaut is named `Mutta`.

It is time for our Astronaut to start working!

If it is not already done, add a getter for his snacks (`getSnacks`), no setter should be created.

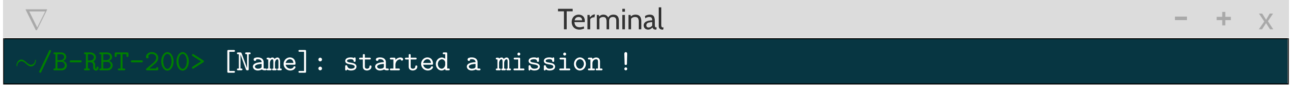
Create a new method `doActions` taking an optional parameter.

If no parameter is given the method simply display:



```
~/B-RBT-200> [Name]: Nothing to do.
```

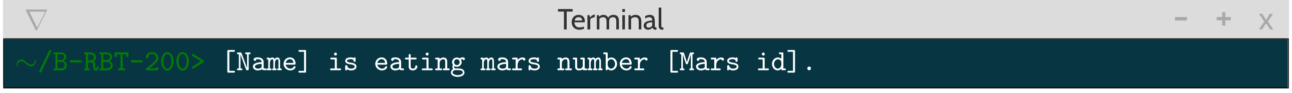
If the parameter is a `planet::Mars` you will display:



```
~/B-RBT-200> [Name]: started a mission !
```

Because some people can't guess obvious things (Obivously I won't say who…), you will also need to store our Astronaut's new destination to his attribute destination (see Task 2).

On the contrary if a `chocolate::Mars` is given you will display:

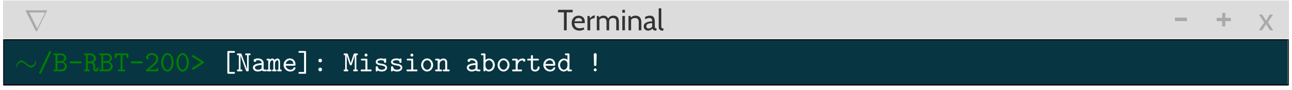


```
~/B-RBT-200> [Name] is eating mars number [Mars id].
```

Where [Mars id] should obviously be replaced by the Mars id.

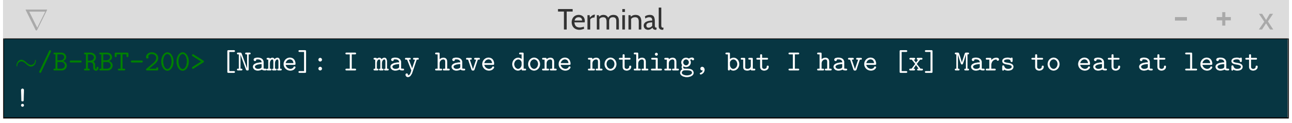
Moreover, if our Astronaut received a snack, you will also need to increment his attribute `snacks` by one.

Finally, during its destruction, if a destination has been selected it will display:



```
~/B-RBT-200> [Name]: Mission aborted !
```

If no destination was selected it will display:



```
~/B-RBT-200> [Name]: I may have done nothing, but I have [x] Mars to eat at least
!
```

Where [x] should be replaced by the number of snacks possessed by our Astronaut.