

SCALE FOR PROJECT PISCINE OCAML (/PROJECTS/PISCINE-OCAML) / DAY 07 (/PROJECTS/PISCINE-OCAML-DAY-07)

Introduction

For the good of this evaluation, we ask you to:

- Stay mannerly, polite, respectful and constructive during this evaluation. The trust between you and the 42 community depends on it.
- Bring out to the graded student (or team) any mistake she or he might did.
- Accept that there might be differences of interpretation of the subject or the rules between you and the graded student (or team). Stay open minded and grade as honestly as possible.

Guidelines

- You must grade only what is present and the graded student's (or team) repository.
- You must stop grading at the first failed exercise, but you are encouraged to continue testing and discussing the following exercises.

Attachments

📎 Subject (<https://cdn.intra.42.fr/pdf/pdf/5497/d07.en.pdf>)

Preview!!!

Preliminaries

This section is dedicated to setup the evaluation and to test the prerequisites. It doesn't rewards points, but if something is wrong at this step or at any point of the evaluation, the grade is 0, and an appropriate flag might be checked if needed.

Respect of the rules

- The graded student (or team) work is present on her or his repository.
- The graded student (or team) is able to explain her or his work at any time of the evaluation.
- The general rules and the possible day-specific rules are respected at any time of the evaluation.

✓ Yes

✗ No

OCaml piscine D07

- For each exercise, you must compile the exercise using `ocamlpt` and run the generated executable. If the compilation fails or warns, or an exception is thrown at runtime, the exercise is failed. - Remember to check function names, types, behaviours and outputs.

Ex00, Do What I do. Hold tight and pretend it's a plan!

Test the people class. It should contain :

- a name attribute of type string (mutable or not, the subject was open)
- an hp attribute of type int initialized to 100.
- a `to_String` method that returns a read-friendly string.
- a `talk` method that should write "I'm [NAME]! Do you know the Doctor?" on the standard output.
- a `die` method that writes "Aaaarghh!" on the standard output.
- an initializer that prints at least the name and the type of the object. (i.e. "Class people instanciated as Amelia_Pond").

ALL tests MUST be implemented by the student. Again, if something is missing, the feature won't be graded.

Rate it from 0 (failed) through 5 (excellent)



Ex01, The Name Of The Doctor!

Test the doctor class. It should contain :

- a name attribute of type string
- an age attribute of type int
- a sidekick attribute that is a an object of the people class.
- an hp attribute of type int initialized to 100.
- a to_String method that returns a read-friendly string.
- a talk method that should write "Hi! I'm the Doctor!" on the standard output.
- an initializer that prints at least the name and the type of the object. (i.e. "Class doctor instanciased as The_Tenth").
- a travel_in_time method that do nothing to doctor age (Duh!) and draw a TARDIS on the standard output.
- an use_sonic_screwdriver method that prints on the standard output "Whiiiiwhiiwhiii Whiiiiwhiiwhiii Whiiiiwhiiwhiii".
- a private method regenerate that set the doctor hp to 100.

Except for the hp attribute, which MUST be mutable, the other attributes can be mutable or not. The subject was open.

ALL tests MUST be implemented by the student. Again, if something is missing, the feature won't be graded.



Rate it from 0 (failed) through 5 (excellent)

Ex02, You are a good Daaaaaalek!

Test the dalek class. It should contain :

- a name attribute of type string (mutable or not), generated randomly according to the pattern DalekXXX so DalekSec DalekSup are good. DalekTaMere is wrong.
- an hp attribute of type int initialized to 100.
- a shield attribute of type boolean, mutable, initialized to true and change it's value each time the exterminate method is used.
- a to_String method that returns a read-friendly string.
- a talk method that should write on of the 4 sentences on the standard output.
- an exterminate method that takes an instance of type people as parameter and kills it instantly.
- a die method that prints "Emergency Temporal Shift!" on the standard output.

ALL tests MUST be implemented by the student. Again, if something is missing, the feature won't be graded.



Rate it from 0 (failed) through 5 (excellent)

Ex03, The Day of The Doctor!

Check the army parameterized class :

- if the class is not a parameterized class... Well sorry for you but it's a big fail.
- the class should contains an attribute (can be mutable or not) of type 'a list composed of objects from one of the previous classes (doctor, people or dalek.)
- It should have an add method allowing to add an object to the list (front or back are ok).
- It should have a delete method that removes an object from the list (front or back are ok).

☒ Yes

☐ No

Ex04, The Time War

Check the gallifrey class. It should have :

- a dalek list attribute.
- a doctor list attribute.
- a people list attribute.
- a do_time_war method that initiate the most epic battle of all space and time.
- any other method or enemy types to improve the Time War dramatically if you find them usefull. Remember the student must EXPLAIN any choice he made.

(Remember: the subject was open, the attributes can be mutable or not.)

☒ Yes

☐ No

Bonus

In the 4th exercise, if the student was not so lazy and provided a fuckload of features to his Gallifrey class, grant him with some bonus points. Nonetheless, if added methods or class don't belong to the Doctor Who Universe, don't give him bonus points, he didn't earn it because he's not fun.

Rate it from 0 (failed) through 5 (excellent)



Ratings

Don't forget to check the flag corresponding to the defense

✓ Ok

📄 Empty work

📄 Incomplete work

💬 No author file

💀 Invalid compilation

📖 Norme

📖 Cheat

💥 Crash

🚫 Forbidden function

Conclusion

Leave a comment on this evaluation

General term of use of the site
(<https://signin.intra.42.fr/legal/terms/6>)

Privacy policy
(<https://signin.intra.42.fr/legal/terms/5>)

Legal notices
(<https://signin.intra.42.fr/legal/terms/3>)

Declaration on the use of cookies
(<https://signin.intra.42.fr/legal/terms/2>)

Terms of use for video surveillance
(<https://signin.intra.42.fr/legal/terms/1>)

Terms of use for the site
(<https://signin.intra.42.fr/legal/terms/4>)