

DevOps assignments

C++

Implement an equestrian display. Below you will find the pre-requisites, implementation and the expected output.

Setup:

- a) There are 3 herds of different animals: horses, zebras and zebroids (zebra-horse hybrid). All have the attribute 'name' and 'age'. Zebras and zebroids have the attribute of 'hasStripes', horses and zebroids have the attribute of 'rideableAnimal'.
- b) The Circus has:
 - i) 4 horses (name/age: Hanni/7, Holger/3, Hennes/2, Hans/11)
 - ii) 3 zebras (name/age: Alex/3, Anke/8, Ann/5) and
 - iii) 2 zebroids (name/age: Otto/2, Olga/2).
- c) The show should show the following:
 - 1) All animals lexicographically ordered by name.
 - 2) All animals ordered by their age (youngest to oldest).
 - 3) First, animals with stripes that are rideable animals, then those with stripes (not rideable), then rideable animals, then any remaining animals.
 - 4) First all horses, then all zebras followed by all zebroids, then any remaining animals.

Implementation:

Implement the classes 'animal', 'horse', 'zebra' and 'zebroid' with the given attributes (a).

Write a program that initializes all the animals in the show (b) and that will list for each of the four stages of the show (c, 1-4) the names of the animals. Use an STL container to store the animals for the show and implement a sorting methods for each stage.

Output:

Stage 1: Alex, Anke, Ann, Hanni, Hans, ...

Stage 2: Hennes, Olga, Otto, Alex, Holger, ...

Stage 3: Olga, Otto, Alex, ...

Stage 4: Hanni, Hans, ...

Deliverable:

Submit an archive of the entire project containing the .h/.cpp files

Python

You are running thousands of tests with ctest in parallel (ctest -R [reg_ex] -j `nproc` --output-on-failure) on a Jenkins node but execution is hanging. From the last line of the ctest output

```
('1392/1395 Test #93: test_feature_x ..... Passed 496.63 sec')
```

you know that 3 tests are still running but you don't know which ones. You don't have access to the Jenkins nodes to check on running processes so you have to parse the entire output of ctest to find the tests that have started their execution

```
('    Start 1645: test_feature_y')
```

but not finished (see above). As everything got stuck, you aborted the job and can't see the ctest summary of failing tests ('1378/1395 Test #1096: ctest_feature_z***Failed 5.11 sec').

Write a script (python3) that parses the output of CTest from a file passed to the script. It should print two summaries: the first summary should list all finished but failed tests, the second should list all started but not finished tests.

Deliverable:

Submit the python script.

Material:

ctestOutput.txt

Groovy

You are running ctests in your scripted pipeline. After a successful execution you want to parse the full ctest output and print the 10 slowest test with their execution time. The possible output looks like the following:

```
1/9 Test #3: foo ..... Passed 38.55 sec
2/9 Test #4: foo2 ..... ***Timeout 100.04 sec
3/9 Test #5: foo3 ..... ***Failed 55.05 sec
```

Write the following two functions for your pipeline script:

```
List getSlowestTests(String log) {
```

```
}  
void showSlowestTests(List slowest) {  
  
}
```

to run them like this:

```
List slowest = getSlowestTests(log)  
showSlowestTests(slowest)
```

Deliverable:

Submit a groovy script with the code snippet of the two functions.