



OCTOBER 2022

C++ PROJECT



IT'S THE CIRCLE OF CODE

DARIUSZ.BRZEZINSKI@CS.PUT.POZNAN.PL

GENERAL DESCRIPTION

The aim of the project is to implement an application that keeps track of animals in a zoo.

CLASS HIERARCHY

1. Each student is tasked with defining their own **hierarchy of animals**;
2. The animal hierarchy should consist of at least **7 classes** connected by relations of inheritance;
3. Each modeled class should have 1 or 2 fields, so that the "leaf" classes of the hierarchy have **at least 3 fields**;
4. Each leaf class should have at least **one numeric and one character field**;
5. Classes that are not at the bottom of the hierarchy should be **abstract**;
6. **No parameterless constructors**; constructor parameters should mirror the classes' fields;
7. Each class should have a **toString()** method that provides text information about a given animal.
8. Before implementing the project, students are asked to prepare a **UML class diagram** of the modeled hierarchy by **October 18, 2022, 23:59 CET**.

REQUIREMENTS

The project should include a generic (template) class called Zoo that will be a collection of animals. Adding an animal to the Zoo should be performed by the "+=" operator, whereas removing one should be done by the "-=" operator.

Students are asked to create a console application that does the following:

1. Creates a user-defined number of random animals and adds them to a zoo.
2. Lists all the animals currently in the zoo (please use the implemented `toString()` method).
3. There should also be a separate class with the method `breed()` that for each animal in the zoo:
 - with a user-defined probability (`pBreed`) crosses the animal with a random animal that is also in the zoo, and adds a new animal (of one of the parents' types) to the zoo; the field values of the offspring should be a combination of the values of the parents;
 - with a user-defined probability (`pExtinct`) removes the animal from the zoo.
4. The program should run `breed()` in a loop until the zoo becomes or reaches 100 animals.
5. After each iteration, the zoo should list all the animals.
6. Each class should be divided into two files: `*.cpp` and `*.h` (or `*.hpp`). The `*.h` files should contain declarations, whereas the `*.cpp` files should host the implementations.

DEADLINE

Please upload the code of your projects to the course website by **November 15, 2021, 23:59 CET**.