

## Report

### Grayling Class

- Grayling Class is the abstract class for Grayling1, Grayling2, Grayling3.
- All attributes are in Grayling Class because all Grayling forms has the same attributes.
- Since all attributes are needed to use in derived classes, attributes set as “protected”.
- print(), givebirth(), aging(), functions defined virtual because every form has different behaviour. But these functions should can be used for other forms with same name so these functions also set as virtual.
- Methods of this class are needed to use in main function so they are set as “public”
- There is only default constructor defined in Grayling Class which creates dead Grayling.
- print() function of Grayling Class include mutual things to print for all grayling forms. Such as age, name, gender, mpi, mp, mutate at.
- aging() function of Grayling Class increase age and mp rate because these operations are the same for all grayling forms.
- Since every grayling form has different behaviours in givebirth() function, Grayling::givebirth() is an empty function.

### Grayling1 Class

- Grayling1 Class is derived “public” mode since function needs to reach every attributes.
- Grayling1 Class has no attributes.
- print() function of Grayling1 Class prints only form name and call base print function.
- aging() function of Grayling1 Class calls base aging function first(to increase age and mp). After necessary increasing done, the function checks if fish should die.
  - It checks first age. Age should be smaller than 4 because Grayling2 form dies at 4. There is no possibility to die before mutation and graylings can mutate only once.
  - After checking age, the function checks mutant rate, if mutation rate is greater than 100 fish should die.
  - If the fish should not die, the function checks if fish should mutate. If the fish is not mutant and “mp” is greater or equal then “mpi” mutation operations operated (print, mutatedTo upgrade, mutant becomes true, mp becomes 0, and mpi becomes 40(since Grayling2)).
  - If the fish only needs aging, print operation operated.
  - Also function first checks if the fish is dead. If it is dead necessary print operation operated.
- givebirth() function determines all possible conditions of not being able to give birth and prints them. If the fish is able to give birth, necessary message is printed.

## Grayling2 Class

-Grayling2 Class is derived "public" mode since function needs to reach every attributes.

-Grayling2 Class has no attributes.

-print() function of Grayling2 Class prints only form name and call base print function.

-aging() function of Grayling2 Class calls base aging function first(to increase age and mp). After necessary increasing done, the function checks if fish should die.

It checks first age. Age should be smaller than 3 because Grayling3 form dies at 3. There is no possibility to die before mutation and graylings can mutate only once.

After checking age, the function checks mutant rate, if mutation rate is greater than 100 fish should die.

If the fish should not die, the function checks if fish should mutate. If the fish is not mutant and "mp" is greater or equal then "mpi" mutation operations operated (print, mutatedTo upgrade, mutant becomes true, mp becomes 0, and mpi becomes 50(since Grayling3)).

If the fish only needs aging, print operation operated.

Also function first checks if the fish is dead. If it is dead necessary print operation operated.

-givebirth() function determines all possible conditions of not being able to give birth and prints them. If the fish is able to give birth, necessary message is printed.

## Grayling3 Class

-Grayling3 Class is derived "public" mode since function needs to reach every attributes.

-Grayling3 Class has no attributes.

-print() function of Grayling1 Class prints only form name and call base print function.

-aging() function of Grayling3 Class calls base aging function first(to increase age and mp). After necessary increasing done, the function checks if fish should die.

It checks first age. Age should be smaller than 3.

After checking age, the function checks mutant rate, if mutation rate is greater than 100 fish should die.

If the fish should not die, then fish only needs aging and print operation operated.

Also function first checks if the fish is dead. If it is dead necessary print operation operated.

-givebirth() function determines all possible conditions of not being able to give birth and prints them. If the fish is able to give birth, necessary message is printed.

## Constructors

----Default constructors create dead fish. Grayling1, Grayling2, Grayling3 default constructors calls directly base constructor.

---Copy constructors of Grayling1, Grayling2, Grayling3 passes object, name and gender to base copy constructor. Base copy constructor directly copy other attributes.

---Constructors passes gender name and other specific class attributes (mpi and mutateAt) to base Constructor. Base constructor directly assign necessary attributes.