

The deadline for this exercise sheet is **Monday, 02.07.2018, 08:00.**

Contents

Property Summary **2**

Function Summary **3**

DISCLAIMER: These are all just suggestions and not necessarily a complete or the best approach to a solution. It just offers hints, general approaches and ideas.

Property Summary

To give you an overview, a **BunnyFriend** can have following properties:

- `name` a name (handed to the initialization method)
- `_age` an age in days (initialized with 0)
- `_fullness` stomach fullness (initialized with 2)
- `_played` how many times it's been played with (initialized with 0)
- `_played_today` if it's been played with today (initialized with `False`)
- `_mode` the mode ('happy', 'angry', 'grown_happy' or 'grown_angry')
- `_looks` a dictionary containing all 4 bunny illustrations corresponding to the modes
- `_interactable` if the user can interact with the bunny (initialized with `True`). Make a function 'interactable' returning this value as well and mask it as a property.

Pay attention to which attributes are supposed to be private and which aren't.

Function Summary

Following functions (or similar) should be implemented into the `BunnyFriend` class:

- `__init__` Initializes the `BunnyFriend` with all its properties.
- `_play` Will increase the `_played` variable by 1 and also set `_played.today` to `True`. Also makes the `BunnyFriend` go into happy mode.
- `_feed` If the stomach is already full (3 meals), will return `False`. Will otherwise increase the `_fullness` variable by 1 and return `True`.
- `interact` Will check if the bunny is interactable and return `False` if it isn't. Will ask the user to enter, which interaction should be performed. If the user chooses playing or feeding, a corresponding message is displayed. Then the program will be paused for a second, the corresponding function is called and another message is displayed after the function was executed. If the user tried to feed the `BunnyFriend`, but the `_feed` function returned `False`, the message will state that the `BunnyFriend` was not hungry. If the user chose neither feeding nor playing, a message will be printed that the `BunnyFriend` will be ignored for now. Will return `True` at the end of the function.
- `_grow` Is called when the `BunnyFriend` grows up (if it was fed appropriately for 10 days). The user will be notified that it is growing up, and after a pause of two seconds the grown-up `BunnyFriend` will be revealed. If the user has played with the `BunnyFriend` for at least 15 times during the 10 days, it will go into `_grown_happy` mode. If not, it will go into `_grown_angry` mode. In both cases, a fitting message is displayed and in both cases, the `BunnyFriend` will not be interactable anymore, as it is now off to its own adventures.
- `pass_day` finishes the day of the `BunnyFriend`. It will update the stomach fullness and age of the `BunnyFriend`. Each day, the fullness is reduced by 1. If the fullness becomes 0, the `BunnyFriend` runs away as it was left to starve and is not interactable anymore. If it becomes 10 days old, it will grow. Otherwise, the mode of the bunny will be updated (to 'sad', if it was not played with during the day), the `_played_today` variable will be reset.
- `__str__` prints the `BunnyFriend` status as described in the main task.

Pay attention to which functions are supposed to be private and which aren't.