Lab: Encapsulation

Problems for in-class lab for the Python OOP Course @SoftUni. Submit your solutions in the SoftUni judge system at https://judge.softuni.bg/Contests/1938

1. Person

Create a class called **Person**. Upon initialization it should receive **name** and **age**. Create **private** properties (cannot be accessed outside the class) called **get_name** and **get_age**. Create two **instance methods**:

Examples

Test Code	Output

`person = Person("George", 32)`
`print(person.get_name())`
`print(person.get_age())`
	`George`
	`20`

2. Email Validator

Create a class called **EmailValidator**. Upon initialization it should receive **min_length** (of the username; example: in **"peter@gmail.com"** **"peter"** is the **username**), **mails** (**list** of the **valid mails**; example: **"gmail"**, **"abv"**), **domains** (**list** of **valid domains**; example: **"com"**, **"net"**). Create **three private methods**:

- **validate_name(name)** - returns whether the name is **greater than or equal to the min_length** (True/False)
- **validate_mail(mail)** - returns whether the **mail is in the possible mails list** (True/False)
- **validate_domain(domain)** - returns whether the **domain is in the possible domains list** (True/False)

Create one **public method**:

- **validate(email)** - using the **three private methods** returns whether the **email is valid** (True/False)

Examples

Test Code	Output

`mails = ["gmail", "softuni"]`
`domains = ["com", "bg"]`
`email_validator = EmailValidator(6, mails, domains)`
`print(email_validator.validate("pe77er@gmail.com"))`
`print(email_validator.validate("georgios@gmail.net"))`
`print(email_validator.validate("stamatito@abv.net"))`
`print(email_validator.validate("abv@softuni.bg"))`
	`True`
	`False`
	`False`
	`False`

3. Mammal

Create a class called **Mammal**. Upon initialization it should receive a **name**, **type** and **sound**. Create **private class attribute** called **kingdom** and set it to be **"animals"**. Create **three more instance methods**:

- **make_sound()** - returns a string in the format **"{name} makes {sound}"**
- **get_kingdom()** - returns the private kingdom attribute
- **info()** - returns a string in the format **"{name} is of type {type}"**

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Follow us:

Examples

Test Code	Output

`mammal = Mammal("Dog", "Domestic", "Bark")`
`print(mammal.make_sound())`
`print(mammal.get_kingdom())`
`print(mammal.info())`
	`Dog makes Bark`
	`animals`
	`Dog is of type Domestic`

4. Account

Create a class called **`Account`**. Upon initialization it should receive an **`id`**, **`balance`** and **`pin`** (all numbers). The **`pin`** and the **`id`** should be **private instance attributes** and the **`balance`** should be **public attribute**. Create **three public instance methods**:

- **`get_id(pin)`** - if the given **`pin`** is correct, return the **`id`**, otherwise return **`"Wrong pin"`**
- **`balance`** - returns the balance
- **`change_pin(old_pin, new_pin)`** - if the old pin is **correct**, **change** it to the new one and return **`"Pin changed"`**, otherwise return **`"Wrong pin"`**

Examples

Test Code	Output

`account = Account(8827312, 100, 3421)`
`print(account.get_id(1111))`
`print(account.get_id(3421))`
`print(account.balance)`
`print(account.change_pin(2212, 4321))`
`print(account.change_pin(3421, 1234))`
	`Wrong pin`
	`8827312`
	`100`
	`Wrong pin`
	`Pin changed`

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Follow us: