

Class Documentation

The Skincare class represents taking care of your skin. It has a money variable and variables denoting numerical comparisons of five traits of one's skin. These attributes can be increased or decreased accordingly depending on different actions taken.

The class variable `progress` represents how well you have done in trying to improve your skin. Since you are trying to improve your (hypothetical) skin in this file, your progress is a 10 (out of 10), as it feels cruel to dictate one's progress by fiscal constraints (or when one's money runs out).

The data variable `money` represents how much money one has available to buy skincare products, \$300. This reduces whenever a skincare product is bought. There are five other data variables, all of which are integers from 1 (worst) to 10 (best). These are: `acne` (representing how much acne someone has), `glow` (how nice someone's skin glows), `hydration` (how hydrated someone's skin is), `wrinkles` (how wrinkly someone's skin is), and `texture` (how smooth someone's skin is). These were all randomized using the `randint()` method from the imported `random` module.

There is a corresponding getter and setter for all six of the data variables above. The `get_money()` method returns how much money the user has. The `get_acne()` method returns the number representing how bad someone's acne is, `acne`. The `get_glow()` method returns the number representing how nice someone's skin is, `glow`. The `get_hydration()` method returns the number representing how hydrated someone's skin is, `hydration`. The `get_wrinkles()` method returns the number representing how wrinkly someone's skin is, `wrinkles`. The `get_texture()` method returns the number representing how bad someone's acne is, `texture`. The `set_money()` method has the argument `price` so it can subtract `price` from `money`, and then setting `money` equal to the aforementioned difference. The `set_acne()`, `set_glow()`, `set_hydration()`, `set_wrinkles()`, and `set_texture()` add 1 to the corresponding value of `acne`, `glow`, `hydration`, `wrinkles`, and `texture` respectively. This makes the value be closer to 10, the maximum and goal. These getters and setters all take in `self` as an argument so they can refer to the necessary variables from the constructor.

There is also the `voucher()` method, which gives a free skincare product. It has a 25% chance of occurring due to the imported `randint()` function. If this 25% possibility is happening, a random number is chosen from 1 to 9, telling which skincare product has been given for free. There are 9 skincare products in the `main()` function so, depending on what skincare product has been chosen, the corresponding stats will be increased by calling the corresponding setters depending on the stat, such as `set_glow()` for `glow` to represent the gains from the free skincare product. This takes `self` as an argument to be able to use the setters. There is also

the `breakout()` method, which causes a breakout resulting in a reduction in one of the stats. It has a 10% chance of occurring due to the imported `randint()` function. If this 10% possibility hasis happening, a random number is chosen from 1 to 5. If 1 is chosen, then acne will be reduced by one. If 2 is chosen, then glow will be reduced by one. If 3 is chosen, then hydration will be reduced by one. If 4 is chosen, then hydration will be reduced by one. If 5 is chosen, then texture will be reduced by one.

Demo Program Documentation

An object of the `Skin` class is created. It goes through a while loop, hoping for one to attain 10 in the five main stats before running out of money. In the loop, it asks the user to type in which skincare product they want to buy. If they type it incorrectly, they are prompted to retype it in. However, if they do type it correctly, the price of the skincare product is subtracted from their total money, and the benefits of the skincare product that was just brought are reaped. After a skincare product is bought, there is a possibility of a breakout happening and the possibility of receiving a free voucher for a skincare product. The loop ends if the user runs out of money, in which they will be told this, or if they have gotten all the five main stats to 10, resulting in a small congratulations.

To run the demo program, one can just run the file `Skincare.py` in Windows Powershell or the terminal of an IDE such as Visual Studio.