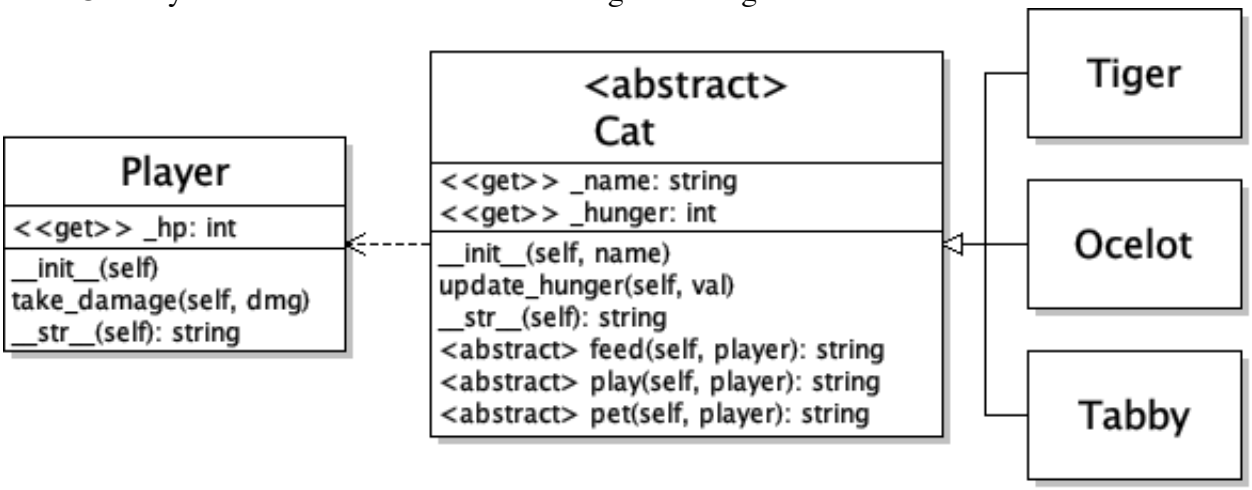
CECS 277 – Lab 9 – Abstract Class

Virtual Cat

Create an interactive pet game where the user has life points and interacts with different types of cats. Create your classes based on the following class diagram:



Cat Class (cat.py) –

1. __init__(self, name) – set the cat's name and assign a default starting value for the hunger value.
2. name and hunger properties – use decorators to get (not set) the values of _name and _hunger.
3. update_hunger(self, val) – value will be a positive or negative value that is added to the cat's _hunger attribute. Make sure that the value of _hunger never leaves the range of 1-10.
4. __str__(self) – return a string with the cat's name and then the cat's hunger level as a bar graph that clearly shows whether the cat is hungry or full.
5. Create the 3 abstract method stubs for the feed, play, and pet functions.

Tabby, Ocelot, Tiger Classes (tabby.py, ocelot.py, tiger.py) –

1. feed(self, player) – use if statements to build a string representing feeding the cat based on the state of the cat's hunger level (ie. if the cat is already full, it probably won't want to eat, but if it's starving, then it'll probably gobble down all of the food).
2. play(self, player) - use if statements to build a string representing playing with the cat based on the state of the cat's hunger level (ie. if the cat is satisfied, it will be more likely to play, but if it is starving, then it'll probably be mad that you're not feeding it).
3. pet(self, player) - use if statements to build a string representing petting the cat based on the state of the cat's hunger level (ie. if the cat is satisfied, it might enjoy being petted, but if it's starving, then it'll be annoyed that you're trying to pet it).
4. Note: some of these interactions should do physical damage to the user. If the cat is hungry or playful, it might scratch or bite the player, and a tiger is going to do more damage to the player than a housecat.
5. Note: each of the interactions should also modify the hunger level by different amounts by calling update_hunger. Feeding the cat should cause the cat to be more full, playing with the cat will make them more hungry. The amounts will differ based on the cat's

hunger level (ex. if they are full, they're less likely to eat the whole bowl of food, so it should only fill them up a little, as opposed to when they are starving, they'll eat all the food and fill up a lot). Tailor each interaction so it makes sense and add some creativity. It shouldn't always be the same amount for each interaction, and it shouldn't be the same amount for each type of cat.

Player Class (player.py) –

1. __init__(self) – set the player's default starting hit points. Give them enough that they'll survive living with a tabby cat for a while, but maybe only a few with a tiger.
2. hp property – use decorators to get (not set) the value of the player's _hp attribute.
3. take_damage(self, dmg) – subtract the dmg from the player's hp. Reset the player's hp to 0 if it goes negative.
4. __str__(self) – return a string with the player's hp.

Main (main.py) –

1. interact_cat(cat, player) – display the cat interaction menu, get the user's input, then use that value to call either feed, play, or pet and display the resulting string from that interaction.
2. main – have the user choose a cat and name it, construct a cat of that type with that name, then repeatedly call interact_cat until the player runs out of hp, at which point the game is over and the program ends.

Example Output:

```
Cat Selection:
1. Tabby Cat
2. Ocelot
3. Tiger
Enter choice: 3
Name your kitty: Fang

You have 25 hp.
Fang:
Starving          Full
|+ + + + + - - - - - |
Cat Menu:
1. Feed your cat
2. Play with your cat
3. Pet your cat
Enter choice: 1
Fang is pretty hungry and
accidentally bites you when it
takes the steak from your hand.

You have 23 hp.
Fang:
Starving          Full
|+ + + + + + + + + + |
Cat Menu:
1. Feed your cat
2. Play with your cat
3. Pet your cat
Enter choice: 2
Fang is so full, when your throw
the ball, it lays there sleepily in
the sun.

You have 23 hp.
Fang:
Starving          Full
|+ + + + + + + + + - |
Cat Menu:
1. Feed your cat
2. Play with your cat
3. Pet your cat
Enter choice: 3
Fang is incredibly full and purrs
happily as they drift off to sleep.

You have 23 hp.
Fang:
Starving          Full
|+ + + + + + + + + - - |
Cat Menu:
1. Feed your cat
2. Play with your cat
3. Pet your cat
Enter choice: 2
Fang jumps and plays with the
soccer ball you threw, then
accidentally tackles you when it
comes running back.
```

```
You have 21 hp.
Fang:
Starving          Full
|+ + + + - - - - - - |
Cat Menu:
1. Feed your cat
2. Play with your cat
3. Pet your cat
Enter choice: 2
Fang sniffs the basketball you have
and then decides that you might be
delicious. Fang bites you for 3
damage.

You have 18 hp.
Fang:
Starving          Full
|+ + - - - - - - - - |
Cat Menu:
1. Feed your cat
2. Play with your cat
3. Pet your cat
Enter choice: 2
Fang is starving, they don't want
to play right now.  Fang stalks
you, chases you down, tackles you,
and takes a large chunk out of your
arm for 8 damage.

You have 10 hp.
Fang:
Starving          Full
|+ - - - - - - - - - |
Cat Menu:
1. Feed your cat
2. Play with your cat
3. Pet your cat
Enter choice: 1
Fang is so hungry that when you set
down the steak, Fang mistakes you
for food and bites you for 5
damage.
```

```
You have 5 hp.
Fang:
Starving          Full
|+ + + + + - - - - - |
Cat Menu:
1. Feed your cat
2. Play with your cat
3. Pet your cat
Enter choice: 3
Fang happily allows you to pet
them.

You have 5 hp.
Fang:
Starving          Full
|+ + + + + - - - - - |
Cat Menu:
1. Feed your cat
2. Play with your cat
3. Pet your cat
Enter choice: 2
Fang sniffs the basketball you have
and then decides that you might be
delicious. Fang bites you for 3
damage.

You have 2 hp.
Fang:
Starving          Full
|+ + - - - - - - - - |
Cat Menu:
1. Feed your cat
2. Play with your cat
3. Pet your cat
Enter choice: 1
Fang is pretty hungry and
accidentally bites you when it
takes the steak from your hand.

Your cat killed you...
```

Notes:

1. You should have 6 different files: cat.py, tabby.py, ocelot.py, tiger.py, player.py, main.py
2. Check all user input using the get_int_range function in the check_input module.
3. Do not create any extra methods, attributes, functions, parameters, etc.
4. Please do not create any global variables or use the attributes globally (ie. do not access any of them using the underscore). Use the properties to access attributes.
5. Use docstrings to document each of the classes, their attributes, and each of their methods. See the lecture notes for examples.
6. Place your names, date, and a brief description of the program in a comment block at the top of your program. Place brief comments throughout your code.

7. Thoroughly test your program before submitting:
    a. Make sure that each interaction increases or decreases the cat's hunger level appropriately (ie. feeding should make them more full, playing should make them hungry).
    b. Make sure that the appropriate response is returned for each type of interaction.
    c. Make sure that the appropriate response is returned for the type of cat they chose.
    d. Make sure that the cat's bar graph clearly shows whether the cat is full or hungry.
    e. Make sure user input is valid.
    f. Make sure that the game ends when the user dies.