

MITx: 6.00.1x Introduction to Computer Science and Programming Using Python

■Bookmarks

Week 7 > Problem Set 7 > Part 2D - SluggishAdopter

■ Bookmark

Overview

Entrance Survey

- Week 1
- Week 2
- Week 3
- Week 4
- Quiz
- Week 5
- Week 6
- ▼ Week 7

Lecture 13 - Trees -Time 51:54 Lecture Sequence

Wrap up - Time 33:39

Problem Set 7

Problem Set due Aug 04, 2016 at 23:30 UTC

Sandbox

The Sluggish Adopter

(10 points possible)

The final type of adopter will be the SluggishAdopter, and it will extend the base Adopter class. A SluggishAdopter really dislikes travelling. The further away the adoption center is linearly, the less likely they will want to visit it. Since we are not sure the specific mood the SluggishAdopter will be in on a given day, we will assign their score with a random modifier depending on distance as a guess.

The SluggishAdopter varies from the regular Adopter because a SluggishAdopter really dislikes travelling. The further away the adoption center is linearly, the less likely they will want to visit it. Since we are not sure the specific mood the SluggishAdopter will be in on a given day, we will assign their score with a random modifier depending on distance as a guess. The SluggishAdopter is a subclass of the Adopter class, and should inherit from it and only it. The SluggishAdopter is __init__ method should look like the following:

__init__(self, name, desired_species, location)

All of the inputs are the same as the Adopter class, *except* that location is a **tuple of floats** of the (x, y) coordinates, similar to the AdoptionCenter 's location variable. The range for the coordinates are -5.0 to 5.0.

For this adopter, you will have to write an additional class method called <code>get_linear_distance(to_location)</code>, which will calculate the linear distance between two points, $(x_1,y_1),(x_2,y_2)$. You will want to calculate the distance by using the following formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

This will be used calculate the linear distance between the SluggishAdopter, and the AdoptionCenter.

The SluggishAdopter 's scoring method also differs from the Adopter 's scoring method. You should override the method so that a score calculated on a SluggishAdopter will return a value that is:

- $1*the_number_of_desired_species$, if the distance is less than 1
 - $random_value_between_0.7_and_0.9*the_number_of_desired_speci$, if the distance is less than 3 but greater than or equal to 1

Part 2D - SluggishAdopter | Problem Set 7 | 6.00.1x Courseware | edX , if the distance is less than 5 but greater than or equal to 3 $random_value_between_0.1_and_0.5*the_number_of_desired_speci$, if the distance is greater than or equal to five. Hints on how to generate random numbers! The scoring method should take only one argument, the AdoptionCenter instance to calculate the score from. Hint: remember AdoptionCenter 's get_location method! Below, please write your implementation of the SluggishAdopter class, including its __init__ method, its _get_linear_distance(to_location) method, and its get_score(adoption_center) method. 1 # Enter your code for the SluggishAdopter class here

© All Rights Reserved



© edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc.

You have used 0 of 30 submissions

Unanswered

















