**Inheritance - Design Activity**

* Class Diagram in Excel spreadsheet

Class Descriptions

Program

**Attributes**

None

**Behaviors**

Main(): void main functionality of program including handling user input and

DisplayMenu(): void display menu

Base Class:

MindfullnessActivity

**Attributes** (that each child class will inherit):

\_name: string the name of the activity which will be displayed in certain places.

\_description: string description of the activity.

\_duration: TimeSpan stores the users input for how long they choose to do the activity.

\_startMessage: string stores the start message for the activity indicating to the user when to start

\_endMessage: string stores the end message for the activity to display when the user finishes the activity

**Behaviors**:

MindfullnessActivity(name, duration): construct the class with inputs from user passed in from Main()

DoForDuration(CallBack): void will call the activity to run on repeat unless the duration is over

DisplayStartMessage(): void will display the start message for the specific activity before the user begins the activity

DisplayEndMessage(): void will display the end message after the activity is complete

PauseAnimation(): void will display the animation, especially during wait times given to the user to read what has been written, before moving to the next part of the program

Wait(): void method for any times in the program meant to pause, often will be used together with PauseAnimation(): and may not be necessary if the built-in function involved in creating the Wait() method is simple enough.

Child classes:

MindfullnessActivity: Breathing

**Attributes**

None

**Behaviors**

Breathing(): constructor which will call base constructor to construct the class and prepare it for use

Display(): void display the prompts for the user to breath in and out (very simple and will be called repeatedly during the DoForDuration() method until duration ends

MindfullnessActivity: Reflection

**Attributes**

\_prompts: list<string> store prompts to be first displayed to the user

\_followUpPrompts: list<string> store prompts of questions for follow up on their initial response to help continue their train of thought and reflect deeper

**Behaviors**

Reflection(): call base class constructor and prepare the class.

Display(): void display the prompts for the user to respond to and after they have responded, display a follow up prompt (very simple and will be called repeatedly during the DoForDuration() method until duration ends

MindfullnessActivity: Listing

**Attributes**

\_prompt: list <string> store prompts for the user to help them write a list based on a specific topic

**Behaviors**

Listing(): call base class constructor and prepare the class.

Countdown(): void Give the user time to think about the printed prompts before then encouraging them to begin listing

Display(): void displays the prompts, and later will display the number of items listed also

CountItemsListed(): int counts the number of items they listed which will be displayed at the end in the end message.

Constructors behavior

Each constructor will accept any relevant input from the user and initialize attributes so that the program is ready to run without complications as the user moves through it.

Interaction

Program class will contain user interaction and for each class instantiated will ask how long they want to do it for. As the Breathing class is created, it also inherits everything in the MindfullnessActivity class.

Everything will run from the Program class with a relatively simple layout as the program has multiple activities it can facilitate, and each one does not extend deeply. Main() in Program will start off and display the menu. The user will choose an activity, then enter how long they want to participate for. Then, DoForDuration() will activate the chosen activity and do so on repeat until the chosen duration is over. When the activity ends, DisplayEndMessage() will be called and display any relevant information to the user and wait before terminating.

After many of the displays and user inputs, there will be a wait time in which the program pauses and displays an animation to show the user that it is waiting, and still running.