**Derive class BreathingActivity**

Attribute

None

Constructor

BreathingActivityInfo()

Behavior

PromptBreathIn(): string

PromptBreathOut(): string

**Mindfulness Program**

**Base class Activity**

Attributes

\_activityName: string

\_description: string

\_duration: int

Constructor

Activity()

Activity(activityName: string, description: string, duration: int)

Behavior

DisplayStartMessage() : void

DisplayEndMessage(): void

Spinner(): string

CountDown(): int

RunActivity(): int

**Derive class RefelctingActivity**

Attribute

List<reflections>(): string

List<questions>(): string

Costructor

ReflectingActivityInfo()

Behavior

GetRandomRefection(): string

GetRandomQueation(): string

DisplayPrompt(): string

DisplayQuestion(): string

**Derive class ListingActivity**

Attribute

List<list>() : string

Constructor

ListingActivityInfo()

Behavior

GetRamdomQuestion(): string

GetUserInput(): string

In this diagram you will see how the base class Activity has all the basic common member variables that the other 3 derived classes will need as well. The derived classes will then get those variables and behaviors and use them when needed. Each will also have whatever they need individually to make the program work. For example, in the breathing activity class there is a prompt for the user to breath IN and another to breath OUT. The countdown will be borrowed from the base class.

The reflection Activity will have a list of questions the user will be prompt to reflect on and those behaviors are individual to that class, etc.