Certain methods and attributes (namely activityName and .timer) will be overwritten in the child classes. I suppose you could say I extend those methods and attributes.

Breathing Activity Class

- -breathingActivityName:String
- -breathInMessage:String
- -breathOutMessage: String
- -countdownCount: int
- +breasthingResults: string
- +BreatingActivity(): BreathingActivity
- +BreathingTimer():Void
- -DisplayBreathCountdown(): void
- -DisplayBreathInMessage(): void
- -DisplayBreathOutMessage(): void
- -RecordBreathingResults(): void

UML Diagram

Zach Newby | June 2, 2023

Activity Class

- -activityName:String
- -activityDurationSeconds:int
- -currentTime:DateTime
- -futureTime:DateTime
- -startMessage:String
- -endMessage: String
- -spinner:String
- -timerMessage: string
- +results: string
- +Activity(): Activity
- +SetActivityDuration(): Void
- +Timer():Void
- +GetName(): String
- -DisplayTimerMessage():void
- -animateSpinner(): void
- -DisplayStartMessage():Void
- Display End Message (int: seconds): Void
- +RecordResultsToFile(): .txt file
- +AskIfRecordResults(): / \id

Reflection Activity Class

- -refletionName: String
- -promptList: List<String>
- -questionList: List<String>
- -answerList: List<string>
- +ReflectionActivity(): ReflectionActivity
- +ReflectionTimer(): void
- -DisplayPrompt(): void
- -DisplayQuestion(): void
- -RecordAnswer(): void
- +GetAnswerList(): String

The menu will ba handled in the program class via while loop, users will select an activity and a new instance of class will be created. The results and answers of the user will be automatically recorded, when the user ends the program they will be asked if they want to save the results to a file, if they do the results will be outputed to a .txt file.

ListingActivity Class

-listingName: String
-promptList: List<String>
-answerList: List<String>

-answerCount: int

- +ListingActivity(): ListingActivity
- +ListingTimer(): Void
- -DisplayPrompt(): void
- -recordAnswer(): void
- -DisplayIAnswerCount():Void
- +GetAnswerList(): String
- -increaseAnswerCount(): void