|  |
| --- |
| Class Activity |
| String \_startMessage  String \_finishMessage  String \_timeInput  Int \_intTimeInput |
| DisplayMessage()  GetTimeInput()  Pause()  Countdown() |

|  |
| --- |
| Class ReflectingActivity : Activity |
| List \_prompts  List \_questions  String \_reflectDesription |
| GetDescription()  GetPrompt()  DisplayQuestions()  CalculateQuestions() |

|  |
| --- |
| Class BreathingActivity : Activity |
| String \_inhale  String \_exhale  Int \_inhaleTime  Int \_exhaleTime  String \_breathDescription |
| GetDescription()  CalculateBreaths()  DisplayBreaths() |

|  |
| --- |
| Class ListingActivity : Activity |
| List \_prompts  String \_userItem  List \_userItems  Int \_userListCount  String \_listDescription |
| GetDescription()  GetPrompt  AppendItem()  DisplayList() |

|  |
| --- |
| Class Menu |
| string \_menuOptions  string \_menuChoice |
| DisplayMenu()  GetUserChoice() |

Upon starting the program, class Menu will use DisplayMenu() to give the option to run activity 1, 2, or 3, or quit. GetUserChoice() will run the selected activity or quit the program.

Activity 1, class BreathingActivity, will base:DisplayMessage() to greet the user, followed by GetDescription() to describe the activity. base:GetTimeInput will receive how long the user wishes to perform the activity. It will use CalculateBreaths() to determine how long each inhale and exhale should be, given the user’s input time. DisplayBreath will display an inhale and exhale message, paired with the calculated breath time. The countdown display will be made using base:Countdown(). After the activity completes, an ending message will be displayed via base:DisplayMessage(). Class Menu will then run again.

Activity 2, class ReflectingActivity, will base:DisplayMessage() to greet the user, followed by GetDescription() to describe the activity. base:GetTimeInput will receive how long the user wishes to perform the activity. It will select a random prompt from a list using GetPrompt(). The prompt will be displayed and CalculateQuestions() will calculate how much time to wait between displaying follow-up questions. DisplayQuestions() will then write questions to the terminal. After the activity completes, an ending message will be displayed via base:DisplayMessage(). Class Menu will then run again.

Activity 3, class ListingActivity, will base:DisplayMessage() to greet the user, followed by GetDescription() to describe the activity. base:GetTimeInput will receive how long the user wishes to perform the activity. It will select a random prompt from the prompts list using GetPrompt(). The user will be invited to respond to the prompt with as many words as possible, each appended to a new list using AppendItem(). After the amount of time base:Countdown() receives expires, the users inputs will be displayed using DisplayList. After the activity completes, an ending message will be displayed via base:DisplayMessage(). Class Menu will then run again.