@startuml

abstract class Activity {

- string \_name

- string \_description

# int \_duration

+ Activity(name: string, description: string)

+ StartActivity(): void

+ EndActivity(): void

# Spinner(seconds: int): void

# Countdown(seconds: int): void

+ {abstract} PerformActivity(): void

}

class BreathingActivity {

+ BreathingActivity()

+ PerformActivity(): void

}

class ReflectionActivity {

- List<string> \_prompts

- List<string> \_questions

+ ReflectionActivity()

- GetRandomPrompt(): string

- GetRandomQuestion(): string

+ PerformActivity(): void

}

class ListingActivity {

- List<string> \_prompts

+ ListingActivity()

- GetRandomPrompt(): string

+ PerformActivity(): void

}

class Program {

+ {static} Main(args: string[]): void

}

Hide empty members

BreathingActivity <|-- Activity

ReflectionActivity <|-- Activity

ListingActivity <|-- Activity

' Program depends on Activity and its subclasses

Program ..> Activity : "creates and uses"

Program ..> BreathingActivity

Program ..> ReflectionActivity

Program ..> ListingActivity

' Notes/Comments for clarity

note "Abstract class: Defines common behavior\nand an abstract method to be implemented\nby concrete activity types." as N1

Activity .. N1

note "Manages available prompts/questions\nand ensures no repetition until all used." as N2

ReflectionActivity .. N2

ListingActivity .. N2

note "Entry point of the application,\nmanages activity selection." as N3

Program .. N3

@enduml