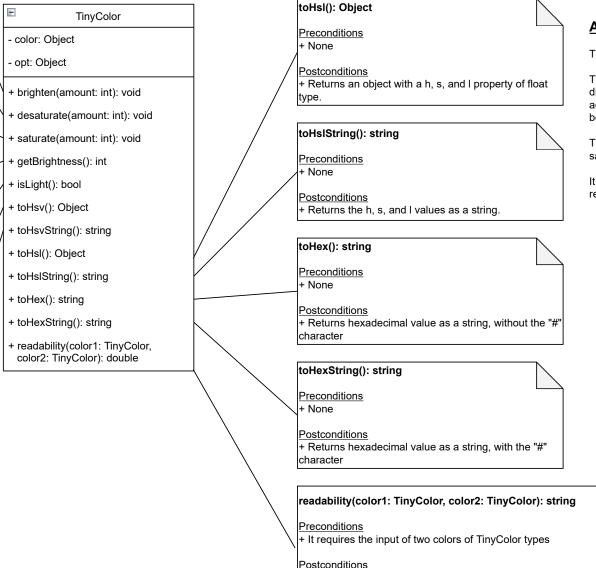
brighten(amount: int): void Preconditions + None <u>Postconditions</u> + Adjusts the TinyColor object's brightness by a given amount. Brightness level will range from 0 to 100 desaturate(amount: int): void **Preconditions** + None Postconditions + Reduces the TinyColor object's saturation by a given amount. The saturation level will range from 0 to 100 saturate(amount: int): void Preconditions + None **Postconditions** + Increases the TinyColor object's saturation by a given amount. The saturation level will range from 0 to 100 getBrigthness (): int **Preconditions** <u>Postconditions</u> + Returns the brightness value of the color from a range of 0 to 255. isLight(): bool **Preconditions** + None <u>Postconditions</u> + Returns a boolean whether the brightness value of the color is considered "light." toHsv(): Object Preconditions + None + Returns an object with a h, s, and v property of float toHsvString(): string **Preconditions** + None

+ Returns the h, s, and v values as a string.

Class Diagram

TinyColor Class - Descriptions, Methods' Preconditions & Postconditions CD-1B



+ Returns contrast ratio value as a double type

Additional Description: TinyColor Class / Object

The TinyColor class and object are from a library and dependency installed for the application.

The TinyColor class has a set of "setter" and "getter" methods that allow color conversation to different color models such as RGB, HSV, HSL, and HEX. It also allows various color adjustments, such as saturation, brightness, and hue. It can even determine the contrast ratio between two colors.

The TinyColor class/object is used as this project's color class/object and allowed the team to save some development time building a class with similar functionalities.

It is the property type of many classes in the application. It is also the parameter type and return type of many class methods.