

Soal: Tahap 1:

1. (20 poin) Buatlah object-oriented analysis dan design program ini.

- (10 poin) Gambarkan hasil dari proses ood ke dalam diagram UML.
 - Class
 - Assosiasi (jumlah dan kepemilikan)
- (05 poin) Tambahkan property dan method (abstraksi) dari class tersebut, baik public maupun private method di dalam diagram UML.
- (05 poin) Berikan alasan dalam memilih abstraksi tersebut, kenapa method dan attribute menjadi public atau private dalam setiap class berupa python comment diatas attribute maupun method bersangkutan.

2. (10 poin) Bonus:

- Sebutkan dan terapkan AbstractClass dalam ood anda.

Jawaban : 1A,1B,1C, dan 2A

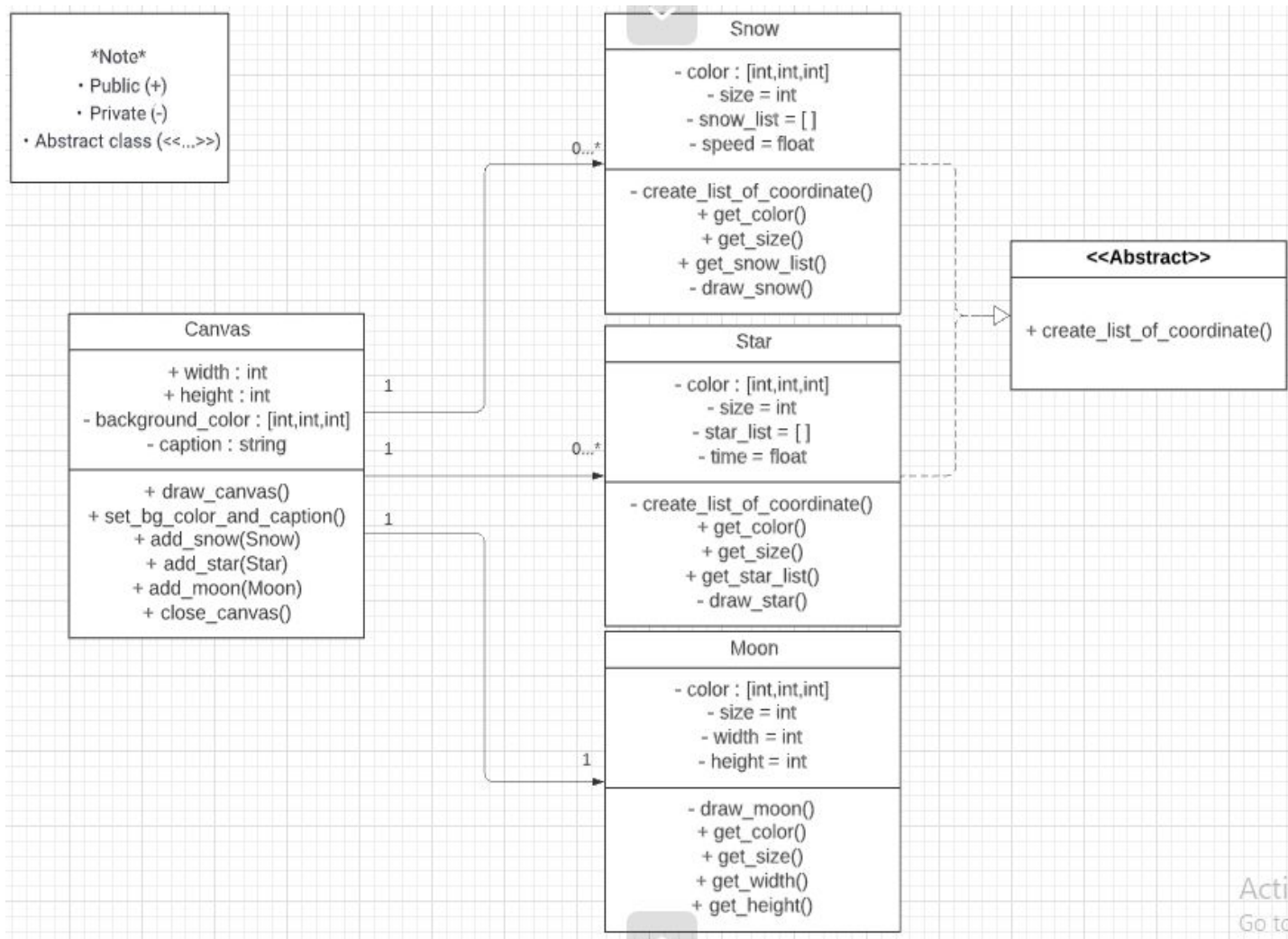


Diagram berikut merupakan diagram OOAD dari program stille nacht beserta dengan property dan method (public maupun private) yang dimiliki oleh tiap class.

Pada class Canvas

- Atribut background color dan caption diprivate karena sesuai dengan apa yang diminta bahwa background harus hitam dan juga caption dari canvas/window tidak bisa diubah.
- Atribut width dan height dipublic agar user bisa mengaksesnya, karena ada kemungkinan bahwa kita bisa mengatur/mengubah panjang atau lebar dari canvas.
- Method draw_canvas public karena ini berfungsi untuk menggambar canvas kita yang akan memakai atribut width dan height yang mana bisa diubah-ubah.
- Method set_bg_color_and_caption private karena user tidak bisa mengubah kondisi dari background maupun caption.
- Untuk Method add_snow(Snow), add_star(Star), dan add_moon(Moon) diprivate karena user tidak perlu mengetahui proses dan tidak perlu menambah/set jumlah salju(akan bergantung pada ukuran canvas), jumlah bintang, dan jumlah bulan. Tetapi perlu diingat bahwa ketiga method ini memerlukan parameter class dari masing-masing Snow, Star, dan Moon yang akan diakses supaya salju, bintang, dan bulan dapat tampil pada canvas.
- Method close_canvas public karena user bisa keluar dari program kapanpun dengan menekan tombol exit/close.

Pada class Snow

- Semua atribut diprivate karena user tidak perlu mengetahui dan juga tidak boleh untuk mengakses warna(color), ukuran(size), list koordinat dari setiap salju(snow_list), dan kecepatan salju turun(speed).
- Untuk Method create_list_of_coordinate dan draw_snow dibuat private karena user tidak perlu tahu proses menempatkan tiap koordinat tiap butir snow dan penggambaran snownya yang terus turun dari atas ke bawah sesuai dengan kecepatannya.
- Untuk semua Method get yang ada seperti get_ dibuat public supaya kita bisa mengakses tiap atribut snow, guna untuk membuat snow di canvas dengan method add_snow yang ada.
- Memiliki hubungan dengan abstract class atau inheritance pada abstract class yang memiliki method create_list_of_coordinate

Pada class Star

- Semua atribut diprivate karena user tidak perlu mengetahui dan juga tidak boleh untuk mengakses warna(color), ukuran(size), list koordinat dari setiap bintang(star_list), dan waktu bintang berkelap-kelip(time).

- Untuk Method `create_list_of_coordinate` dan `draw_star` dibuat `private` karena user tidak perlu tahu proses menempatkan tiap koordinat dari tiap bintang dan penggambaran star yang terus berkelap-kelip sesuai dengan waktu yang ditetapkan.
- Untuk semua Method `get` yang ada dibuat `public` supaya kita bisa mengakses tiap atribut star, guna untuk membuat/memasukkan star di canvas dengan method `add_star` yang ada pada class canvas.
- Memiliki hubungan dengan abstract class atau inheritance pada abstract class yang memiliki method `create_list_of_coordinate`

Pada class Moon

- Semua atribut `private` karena user tidak perlu mengetahui dan juga tidak boleh untuk mengakses warna(`color`), ukuran(`size`), titik/koordinat tinggi dari bulan (`height`), dan titik/koordinat lebar dari bulan(`width`).
- Untuk Method `draw_moon` dibuat `private` karena user tidak perlu tahu proses penempatan dan penggambaran moon yang berada pada `height` dan `width` yang dipasang.
- Untuk semua Method `get` yang ada dibuat `public` supaya kita bisa mengakses tiap atribut moon, guna untuk membuat/memasukkan moon di canvas dengan method `add_moon` yang sudah ada pada class canvas.

Abstract Class

- Class yang abstrak yang memiliki method tertentu dan berperan sebagai induk class(seperti pada kasus tree dimana abstract class ini adalah root dan yang mewarisi/keturunannya adalah leaf)
- Memiliki method `create_list_of_coordinate` sehingga dimiliki class star dan class snow atau dengan kata lain mereka memiliki/mewarisi abstract class. Methodnya tentu bersifat `public` supaya dapat bebas diakses oleh class lain.

.

.

.

.

.

.

.

Hanya contoh opsi lain dari diagram UML yang saya buat

Semua Method saya buat public supaya dapat diakses dimana saja

