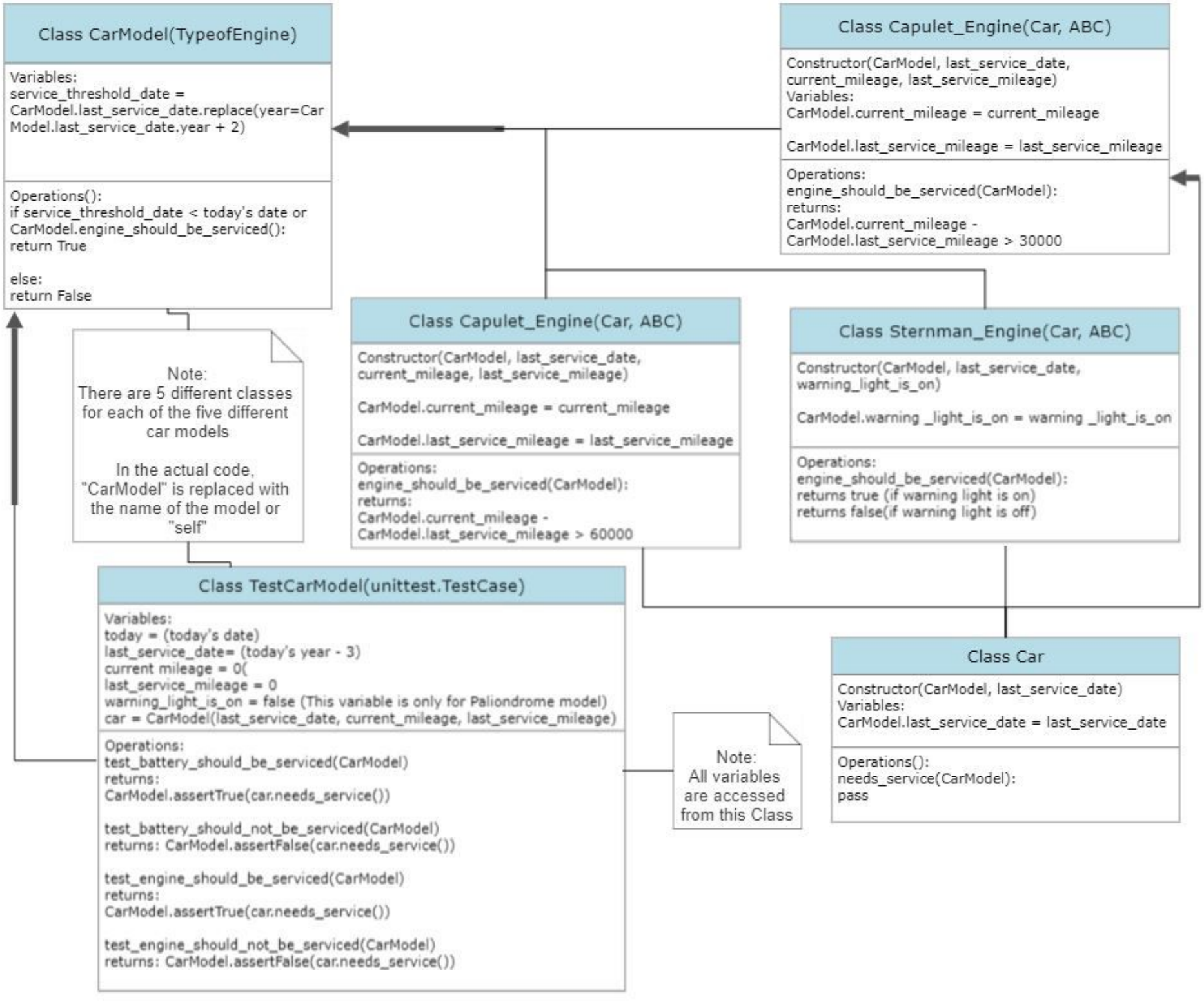
Class CarModel(TypeofEngine)
Variables:
service_threshold_date = CarModel.last_service_date.replace(year=CarModel.last_service_date.year + 2)
Operations():
if service_threshold_date < today's date or CarModel.engine_should_be_serviced():
return True
else:
return False
Class Capulet_Engine(Car, ABC)
Constructor(CarModel, last_service_date, current_mileage, last_service_mileage)
Variables:
CarModel.current_mileage = current_mileage
CarModel.last_service_mileage = last_service_mileage
Operations:
engine_should_be_serviced(CarModel):
returns:
CarModel.current_mileage - CarModel.last_service_mileage > 30000
Note:
There are 5 different classes for each of the five different car models
In the actual code, "CarModel" is replaced with the name of the model or "self"
Class Capulet_Engine(Car, ABC)
Constructor(CarModel, last_service_date, current_mileage, last_service_mileage)
CarModel.current_mileage = current_mileage
CarModel.last_service_mileage = last_service_mileage
Operations:
engine_should_be_serviced(CarModel):
returns:
CarModel.current_mileage - CarModel.last_service_mileage > 60000
Class Sternman_Engine(Car, ABC)
Constructor(CarModel, last_service_date, warning_light_is_on)
CarModel.warning _light_is_on = warning _light_is_on
Operations:
engine_should_be_serviced(CarModel):
returns true (if warning light is on)
returns false(if warning light is off)
Class TestCarModel(unittest.TestCase)
Variables:
today = (today's date)
last_service_date= (today's year - 3)
current mileage = 0(
last_service_mileage = 0
warning_light_is_on = false (This variable is only for Paliondrome model)
car = CarModel(last_service_date, current_mileage, last_service_mileage)
Operations:
test_battery_should_be_serviced(CarModel)
returns:
CarModel.assertTrue(car.needs_service())
test_battery_should_not_be_serviced(CarModel)
returns: CarModel.assertFalse(car.needs_service())
test_engine_should_be_serviced(CarModel)
returns:
CarModel.assertTrue(car.needs_service())
test_engine_should_not_be_serviced(CarModel)
returns: CarModel.assertFalse(car.needs_service())
Note:
All variables are accessed from this Class
Class Car
Constructor(CarModel, last_service_date)
Variables:
CarModel.last_service_date = last_service_date
Operations():
needs_service(CarModel):
pass