Create a new class called `CarPart` that will represent a generic car part. This class should have attributes for the name, brand, model, and year of the part, as well as methods for getting and setting these attributes.

```
class CarPart:
    def __init__(self, name: str, brand: str, model: str, year: int):
        self.name = name
        self.brand = brand
        self.model = model
        self.year = year

    def get_name(self) -> str:
        return self.name

    def set_name(self, name: str) -> None:
        self.name = name

    def get_brand(self) -> str:
        return self.brand

    def set_brand(self, brand: str) -> None:
        self.brand = brand

    def get_model(self) -> str:
        return self.model

    def set_model(self, model: str) -> None:
        self.model = model

    def get_year(self) -> int:
        return self.year

    def set_year(self, year: int) -> None:
        self.year = year
```

Create a new class called `ServiceCriteria` that will represent the service criteria for a given car part. This class should have attributes for the criteria name, a list of criteria that must be met, and a list of criteria that are optional. It should also have methods for getting and setting these attributes.

```
class ServiceCriteria:
    def __init__(self, name: str, required_criteria: List[str], optional_criteria: List[str]):
        self.name = name
        self.required_criteria = required_criteria
        self.optional_criteria = optional_criteria

    def get_name(self) -> str:
        return self.name

    def set_name(self, name: str) -> None:
        self.name = name

    def get_required_criteria(self) -> List[str]:
        return self.required_criteria

    def set_required_criteria(self, required_criteria: List[str]) -> None:
        self.required_criteria = required_criteria

    def get_optional_criteria(self) -> List[str]:
        return self.optional_criteria

    def set_optional_criteria(self, optional_criteria: List[str]) -> None:
        self.optional_criteria = optional_criteria
```

Modify the `Battery` class to inherit from `CarPart` and add a `ServiceCriteria` attribute that represents the service criteria for the battery. Update the `__init__` method to take a `ServiceCriteria` object as a parameter and set it as an attribute.

```
class Battery(CarPart):
    def __init__(self, name: str, brand: str, model: str, year: int, service_criteria: ServiceCriteria):
        super().__init__(name, brand, model, year)
        self.service_criteria = service_criteria

    def get_service_criteria(self) -> ServiceCriteria:
        return self.service_criteria

    def set_service_criteria(self, service_criteria: ServiceCriteria) -> None:
        self.service_criteria = service_criteria
```

Modify the `Engine` class to inherit from `CarPart` and add a `ServiceCriteria` attribute that represents the service criteria for the engine. Update the `__init__` method to take a `ServiceCriteria` object as a parameter and set it as an attribute.

```
class Engine(CarPart):
    def __init__(self, name: str, brand: str, model: str, year: int, service_criteria: ServiceCriteria):
        super().__init__(name)
```