Recap

Inheritance

- a new kind of relationship between classes
- "is a"
- all methods of a parent are inherited by a child class
- what can the child class do about an inherited method?
    1. Simply inherit it as-is, ie, accept it
    2. Override it to implement it
    3. Override it to replace it
    4. Override it to extend it

- A method is "abstract" if it raises a NotImplementedError.
- A class is abstract if it has any abstract methods
- What's the point???

    Parent can define an interface that all of it's kids will share

    Client code can count on this.

Examples

`fleet: List[Vehicle]`

```
for v in fleet:
    v.[          ]
```

`staff: Dict[int, Employee]`

```
for staff_id, emp in staff.items():
    emp.[          ]
```

Bonus material posted after class.

- monsters.py: 1) special method `__str__`
  2) attributes are not inherited
- SDM-28jan.py: Move code + comments on value of a shared public interface