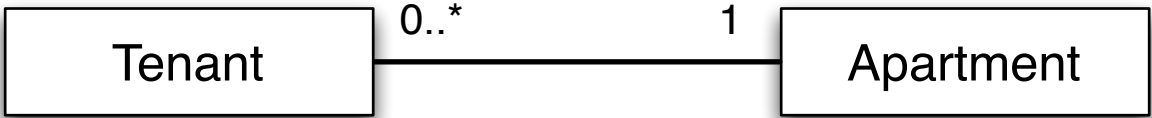
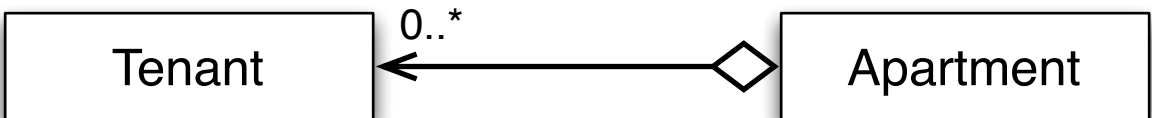
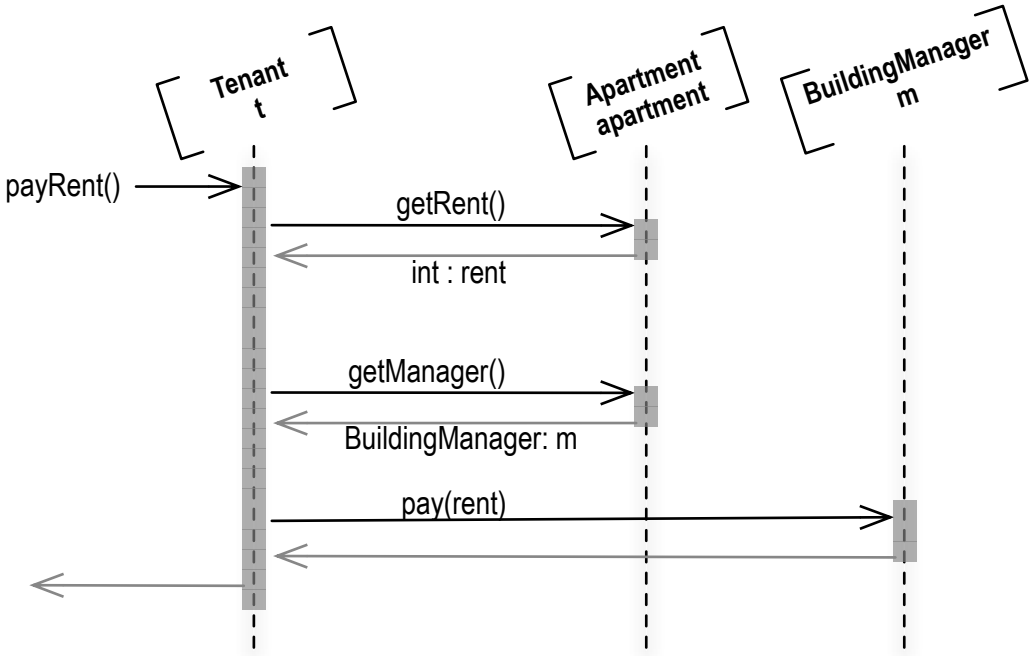


1	
1a	 <pre> classDiagram class Tenant class Apartment Tenant "0..*" -- "1" Apartment </pre>
1b	 <p style="text-align: center;">optional diamond</p>
1c	No - it's not substitutable because it strengthens/narrows the precondition (it only runs properly if the range is narrower than the superclass range).
1d	 <pre> sequenceDiagram participant t as Tenant t participant apartment as Apartment apartment participant m as BuildingManager m t->>t: payRent() activate t t->>apartment: getRent() activate apartment apartment-->>t: int : rent deactivate apartment t->>apartment: getManager() activate apartment apartment-->>t: BuildingManager: m deactivate apartment t->>m: pay(rent) activate m m-->>t: deactivate m t-->>: deactivate t </pre>
2	A E B E C D
3a	<pre> /* REQUIRES: nothing or no requires clause at all. EFFECTS: Returns a new record made up of the name, id and result Strings throws NullNameException if name is null throws NullIDException if id is null throws NullResultException if result is null */ public Record mergeRecord (String name, String id, String result) throws NullNameException, NullIDException, NullResultException { if (name==null) throw new NullNameException; if (id==null) throw new NullIDException; if (result==null) throw new NullResultException; return new Record(name,id,result); } </pre>

CPSC210 - 2015W1 MT2 Solutions Guide.

3b

name	id	result	expected
null	null or non-null	null or non-null	throws NullPointerException
non-null	null	null or non-null	throws NullIDException
non-null	non-null	null	throws NullResultException

4

```
public class Person implements Student, Jogger{
    public String study(){
        return "studying";
    }
    public String jog(){
        return "jogging";
    }
}

public interface Student{
    public String study();
}

public interface Jogger{
    public String jog();
}
```

5

```
public class CensusTaker{
    List<Person> people;
    List<Address> addresses;
    public CensusTaker(){
        people = new ArrayList<Person>();
        addresses = new ArrayList<Address>();
    }
    public void takeCensus(){
        for (Person p : people){
            addresses.add(p.getAddress());
        }
    }
}

public class Person {
    address a;
    public Person(Address a){
        this.address = a;
    }
    public Address getAddress(){
        return a;
    }
}
```

6a

Location: Pet class. Override hashCode and equals

6b

Location: PetSitter:26. add record.makeEntry(did);

6c

Location: TestPetSitter:34. add margaret.assign(angelo);

6d

Location: PetSitter:39 remove: return Null, add throw new NoSuchRecordException();