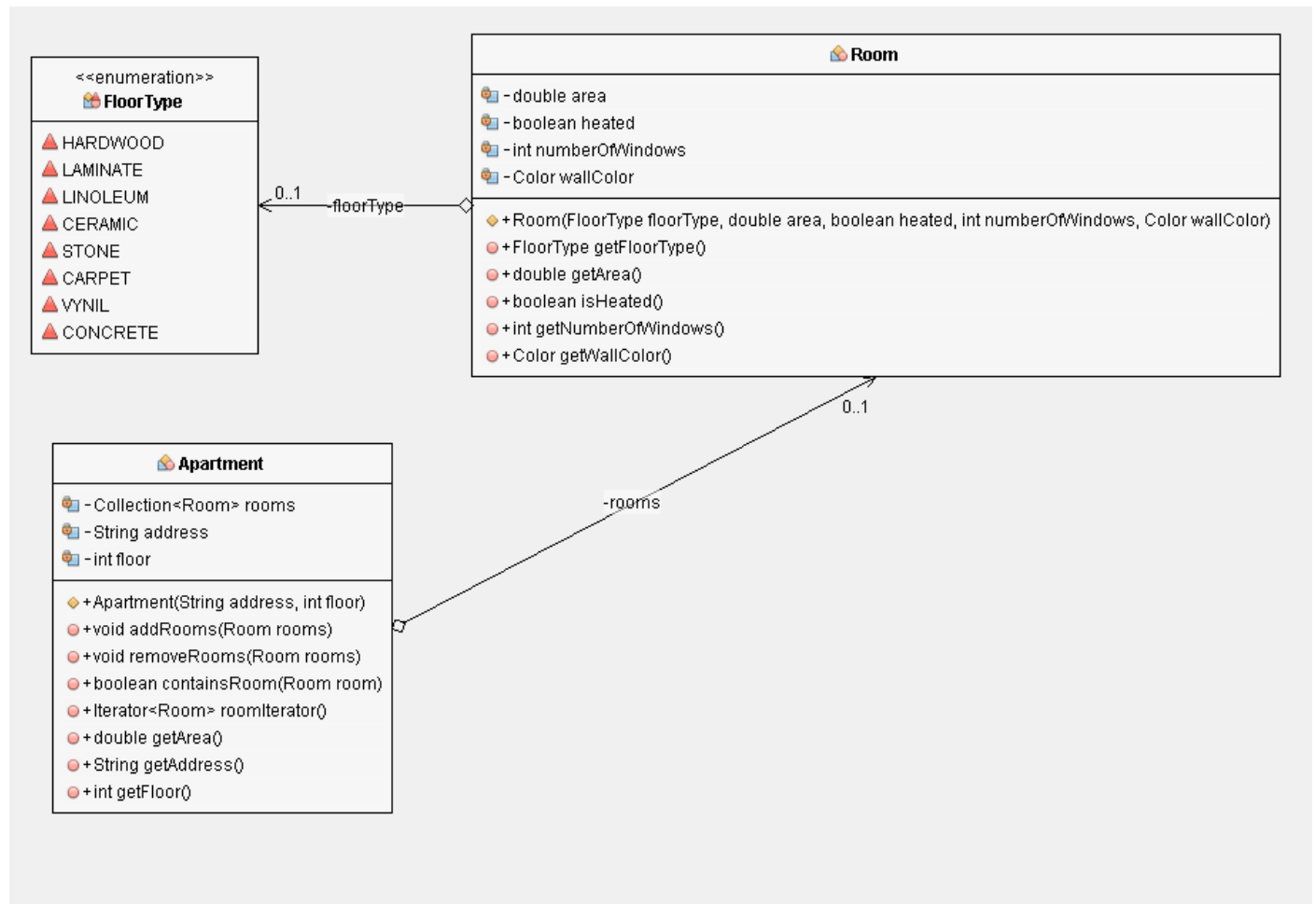


Zadaci za vježbu iz upotrebe vlastitih klasa s Javinim okvirom kolekcija i komparatora

U sljedećem nizu zadataka potrebno je nadograditi programski kod primjera koji modelira aplikaciju za iznajmljivanje apartmana. Programski kod primjera se sastoji od klasa `Room` i `Apartment` te enumeracije `FloorType`. UML dijagram ovih klasa je prikazan na sljedećoj slici.



1. Napišite programski kod svih klasa, a pri tome atribut `rooms` u klasi `Apartment` koji je tipa `Collection<Room>` modelirajte klasom `ArrayList`. Nakon toga provjerite ispravnost napisanog programskog koda izvršavanjem sljedeće metode main klase `Main`:

```
public static void main(String[] args) {
    Apartment apartment = new Apartment("Unska 2, 10000 Zagreb", 2);

    Room kitchen = new Room(FloorType.CERAMIC, 8.56, false, 1,
        Color.YELLOW);
    Room bathRoom = new Room(FloorType.CERAMIC, 7.28, true, 0,
        Color.BLUE);
    Room bedRoom1 = new Room(FloorType.HARDWOOD, 12.23, true, 1,
        Color.PINK);
    Room bedRoom2 = new Room(FloorType.HARDWOOD, 17.33, true, 2,
        Color.GREEN);
```

```

        Room livingRoom = new Room(FloorType.HARDWOOD, 20.08, true, 3,
Color.ORANGE);

        apartment.addRooms(kitchen, bathRoom, bedRoom1, bedRoom2,
livingRoom);

        System.out.println("Added room found: " +
apartment.containsRoom(bedRoom2));
        System.out.println("Identical room found: " +
apartment.containsRoom(new Room(FloorType.HARDWOOD, 17.33, true, 2,
Color.GREEN)));
    }

```

Zašto identična soba nije pronađena? Na koji način se soba pronalazi u apartmanu (tj. na koji se način neki element pronalazi u kolekciji tipa `ArrayList`)? Pogledajte izvorni kod klase `ArrayList`.

2. U klasi `Room` automatski generirajte (korištenjem razvojne okoline) **isključivo** metodu `equals`. Obratite pažnju na koji način su uspoređene vrijednosti atributa tipa `double` i atributa koji su objekti. Proučite izvorni kod klase `Objects` i `Double` te utvrdite na koji način su uspoređeni ovi tipovi atributa. Je li ovaj put ispravno pronađena identična soba?
3. Promijenite klasu `Apartment` tako da atribut `rooms` koji je tipa `Collection<Room>` modelirate klasom `HashSet`, umjesto `ArrayList`. Zašto ovaj put nije pronađena identična soba? Na koji način neki element pronalazi u kolekciji tipa `HashSet`? Pogledajte izvorni kod klase `HashSet`.
4. U klasi `Room` implementirajte metodu `hashCode`. Za izračun *hash* vrijednosti na osnovu više atributa koristite metodu `hash` iz klase `Objects`. Proučite izvorni kod klase `Objects` te utvrdite na koji način se računa *hash* vrijednost metodom `hash` za attribute tipa `double`.
5. Zakomentirajte metodu `equals` u klasi `Room`. Zašto niti ovaj put nije pronađena identična soba? Proučite programski kod klase `HashSet`.
6. Otkomentirajte metodu `equals` u klasi `Room`. Hoće li se u apartmanu uspjeti pronaći treća soba koja je skoro pa identična drugoj sobi u liniji ispod? Objasnite odgovor.

```

System.out.println("Almost identical room found: " +
apartment.containsRoom(new Room(FloorType.HARDWOOD, 18.01 - 0.68, true, 2,
Color.GREEN)));

```

7. Prepravite programski kod gornjih klasa na način da se umjesto tipa `double` koristi tip `BigDecimal`. Proučite JavaDoc klase `BigDecimal`. Hoće li ovaj put u apartmanu biti pronađena treća soba koja je skoro pa identična identičnoj sobi iz prethodnog primjera? Ovo provjerite za slučajeve kad je atribut `rooms` u klasi `Apartment` modeliran klasama `HashSet` i `ArrayList`.
8. U aplikaciji za modeliranje apartmana je zaključeno da generička klasa `Room` ne modelira dovoljno dobro kupaonicu. Zbog toga iz klase `Room` izvedite novu klasu `BathRoom` koja ima sljedeća tri dodatna atributa: `boolean hasBidet`, `int numberOfSinks` te `String bathtubType`. Zamijenite sljedeću liniju u metodi `main` s onom koja je prikazana ispod nje:

```

Room bathRoom = new Room(FloorType.CERAMIC, new BigDecimal("7.28"), true, 0,
Color.BLUE);//original line

```

```

Room bathRoom = new BathRoom(true, 1, "shower", FloorType.CERAMIC, new
BigDecimal("7.28"), true, 0, Color.BLUE);//changed line

```

Što će se ispisati za sljedeće liniju u metodi `main`? U čemu je problem i kako ga možemo riješiti?

```

System.out.println(apartment.containsRoom(new BathRoom(true, 1, "shower",
FloorType.CERAMIC, new BigDecimal("7.28"), true, 0, Color.BLUE)));

```

```
System.out.println(apartment.containsRoom(new BathRoom(false, 2, "bath",
FloorType.CERAMIC, new BigDecimal("7.28"), true, 0, Color.BLUE)));
```

9. Nadjačajte i implementirajte metode `equals` i `hashCode` u klasi `BathRoom`. Što će sad ispisati linije iz prethodnog zadatka?
10. Promijenite klasu `Apartment` tako da atribut `rooms` koji je tipa `Collection<Room>` modelirate klasom `TreeSet`. Koju promjenu uočavate?
11. U klasi `Apartment` sortirajte elemente kolekcije `rooms` **isključivo** po broju prozora. Pri tome iskoristite podrazumijevanu metodu `comparing` u sučelju `Comparator`. U `main` metodi ispišite broj soba u apartmanu. Radi li primjer ispravno? Proučite programski kod klase `TreeSet`.
12. U klasi `Apartment` sortirajte elemente kolekcije `rooms` po broju prozora te zatim po ostalim atributima (nebitno kojim poretom). Boje usporedite po njihovoj RGB vrijednosti koja je tipa `int`. Radi li primjer ispravno?
13. Predložite način rješavanja prethodno uočenog problema. Hoće li predloženo rješenje raditi baš u svim situacijama? Zašto ovaj problem nismo imali s kolekcijama `ArrayList` i `HashSet`?
14. Promijenite klasu `Apartment` tako da atribut `rooms` koji je tipa `Collection<Room>` modelirate klasom `HashSet`, umjesto `TreeSet`. Iskoristite gotovu klasu `ApartmentLoader` i njenu statičku metodu `load` za učitavanje polja `apartmana`. Hoće li ispravno raditi sljedeće linije metode `main`? Objasnite odgovor.

```
Apartment[] apartments = ApartmentLoader.load();
List<Apartment> listOfApartments = Arrays.asList(apartments);
```

```
System.out.println("Added apartment found: " +
listOfApartments.contains(apartments[2]));
System.out.println("Identical apartment found: " +
listOfApartments.contains(apartment));
System.out.println("Are these apartments identical? " +
apartments[0].equals(apartment));
```

15. Nadjačajte i implementirajte metode `equals` i `hashCode` u klasi `Apartment`. Što će sad ispisati metoda `main`? Prokomentirajte kako bi izgledale metode `equals` i `hashCode` da je atribut `rooms` bio tipa `Room[]`.
16. U klasi `Apartment` napravite atributne komparatore za attribute `address` i `floor`. Osim toga napravite komparatore po ukupnom broju soba i površini. U glavnom programu sortirajte apartmane (koje ste prethodno učitali metodom `ApartmentLoader.load()`) po sljedeća 3 kriterija odjednom 1) silazno po broju soba pa 2) uzlazno po katu i 3) silazno po površini te ih ispišite po ovim kriterijima usporedbe.

Rješenja zadataka: <https://github.com/FER-OOP/Lectures/tree/master/Exercises/Homework-12>