

Consider the following class diagram for a Building database system. There are two types of buildings in the system – for sale and for rent

Building

-id: int

- noOfFloors: int

-floorSize: int(in square feet)

-noOfUnits: int

-perSquareFeetPrice

- dateBuilt: Calendar

+Building()

+Building(int id, int noOfFloors, int noOfUnits, int day, int month, int year)

+setters and getters

BuildingForSell

+constructor, setters and getters

+getPrice()

BuildingForRent

+constructor, setters and getters

+getPrice()

Create the required classes with appropriate constructor , setters, getters, and to string methods. In the parameterized constructor of the parent class, you have to take day , month, and year as parameters and set a Calendar instance. Inside the constructor , you need to check a few things: id the number of floors is less than 5, the constructor should throw an user-defined exception stating “Not enough floors!”, if the numbers of units is laess than 2, the constructor should throw an user-defined exception stating ,”not enough units!”, and if the floor size is less than 1800 square feet, the constructor should throw an user-defined exception stating,”not big enough!”. You should create an user- defined exception class to implement these features.

getPrice() method of BuildingForSell class returns the price of the building. The price of a building depends on a few variables. Assuming the price of per square feet is 5000 BDT, to calculate the base price of a building, you have to multiply the total floor size of the entire building by 5000.However,the building also depends on how old the building is. Older buildings tend to have lower prices. The building price will be adjusted baes on the following table.

Old more than 30 years – Base Price

Old more than 30 years – Base Price + 10%

Old more than 30 years - Base Price + 20%

Old more than 30 years - Base Price + 30%

On the other hand, getPrice() method of BuildingForRent will calculate the monthly rent using the following formula-

$$\text{MonthlyRent} = (\text{Building price}/600) + ((\text{Building price}/600) * 50\%)$$

Create a static ArrayList of Building objects in the main class, take three BuildingForSell and three BuildingForRent objects as input, and add them to arraylist. Display all the information of the building from the arraylist .write “For sell” before printing the objects of BuildingForSell, and write “For rent” before the other object. Sort the arraylist based on their age. The newer buildings should appear first, If the date is equal, the building with higher square feet should appear first.