

ამოცანა 1 – OOP

OOP სტრუქტურის გათვალისწინებით შექმენით უძრავი ქონების მართვის სისტემა(ჯამში 10 ქულა)

1. შექმენით კლასი **Realty**(უძრავი ქონება) პარამეტრებით **code**(საიდენტიფიკაციო კოდი), **price**(გადახდილი თანხა). კლასში დაიცავით ინკაფსულაციის პრინციპები(private ატრიბუტები, get-set მეთოდები, კონსტრუქტორები) - **1.5 ქულა**
2. Realty კლასი გამოაცხადეთ აბსტრაქტულად და დაამატეთ ერთი აბსტრაქტული მეთოდი, რომელიც აბრუნებს უძრავ ქონებაზე დაწესებულ გადასახადს - **1 ქულა**
3. შექმენით Realty კლასის შვილი კლასი **House**, რომელსაც დამატებით პარამეტრად ექნება **area**(ფართობი) და **placeCoefficient**(ადგილმდებარეობის კოეფიციენტი). დაიცავით ინკაფსულაციის და მემკვიდრეობითობის პრინციპები - **1 ქულა**
4. შექმენით Realty კლასის შვილი კლასი **Cur**, რომელსაც დამატებით პარამეტრად ექნება **verification**(არის თუ არა შემოწმებული - boolean). დაიცავით ინკაფსულაციის და მემკვიდრეობითობის პრინციპები - **1 ქულა**
5. ზემოთ აღწერილ შვილ კლასებში გადაფარეთ უძრავ ქონებაზე დაწესებული გადასახადის მეთოდი. House-ში დააბრუნეთ მნიშვნელობა შემდეგი ფორმულით: $(price + area * placeCoefficient) * 0.1$. Cur-ში დააბრუნეთ გადახდილი თანხის 10 პროცენტი თუ შემოწმება გავლილია, წინააღმდეგ შემთხვევაში გადახდილი თანხის 12 პროცენტი - **1.5 ქულა**
6. main მეთოდში შექმენით House და Car კლასის რამდენიმე სატესტო ობიექტი და ჩააგდეთ Realty კლასის მასივში ან სიაში. გადაურბინეთ ამ მონაცემთა სტრუქტურას და დაბეჭდეს ყველა უძრავი ქონების კოდი და შესაბამისი უძრავ ქონებაზე დაწესებული გადასახადი - **1.5 ქულა**
7. სანამ დაბეჭდავთ ინფორმაციას უძრავი ქონების შესახებ, გამოიძახეთ მეთოდი **validateRealties**, რომელიც დაადგენს გადაცემულ Realty-ებში ხომ არ მეორდება საიდენტიფიკაციო კოდი. თუ კოდი არ არის უნიკალური გაისროლეთ საკუთარი გამონაკლისი შემთხვევა -

RealtyException. ამ მეთოდის გამოძახება დაამუშავეთ try-catch ბლოკში
- 1.5 ქულა

8. აპლიკაციის მთლიანობა/გაშვებადობა - 1 ქულა