

## ZADATAK

Izradite dijagram klasa (upload-ati png/jpg sliku dijagrama) (**1 bod**) i implementaciju (Visual Studio Solution) (**1 bod**) jednostavnog softverskog modula (Class Library – dll) za sustav pametne kuće, koji bi reprezentirao mjerne uređaje za potrošnju različitih resursa (npr. voda, struja). Pretpostavimo da iz fizičkih mjernih uređaja možemo dobiti sirove podatke o potrošnji za svih 24 sata prethodnog dana, i to u tekstualnom obliku (satne vrijednosti su odvojene sa ';' npr. „1.2; 0.9; 1.1; 1.6; ...“).

Analizirajući korisničke zahtjeve odlučili ste krenuti sa sljedećim dizajnom:

- Klasa **HourlyData** treba imati podatak o satu u danu za koji je izmjerena potrošnja (svojstvo **Hour** tipa int), te vrijednost potrošnje za taj sat (svojstvo **Value** tipa double). Također, definira i konstruktor preko kojeg proslijeđujemo *hour* i *value* vrijednosti.
- S obzirom da možemo imati različite mjerne uređaje, koncept mjernog uređaja bi implementirali na sljedeći način:
  - Ono što bi svaki mjerni uređaj trebao imati bi propisali sučeljem **IMeter**. Svaki mjerni uređaj bi morao imati ime (svojstvo **Name** tipa string), jedinicu mjere (svojstvo **Unit** tipa string), te podatke o potrošnji za 24 sata prethodnog dana (svojstvo **Data** koje bi bilo lista od 24 objekta **HourlyData**).
  - Sučelje **IMeter** bi bilo realizirano apstraktnom klasom **Meter**. Klasa bi imala **konstruktor** kroz koji bi proslijedili parametre za *name*, *unit* i *rawData* (sirovi podaci o potrošnji u obliku stringa, sa vrijednostima odvojenim znakom ';'). Klasa bi također definirala i **apstraktnu zaštićenu** metodu **PopulateDailyData** koja bi primala parametar *rawData*, i bila bi tipa *void*. Ovu metodu treba pozvati u konstruktoru apstraktne klase.
  - Apstraktnu klasu **Meter** bi naslijedile konkretne klase **ElectricityMeter** i **WaterMeter**.
    - Klasa **ElectricityMeter** bi implementirala konstruktor kojem bi proslijedili parametre za *name*, *unit* i *rawData*. Također, „pregazila“ bi metodu **PopulateDailyData** i ponudila svoju implementaciju (uz pomoć metode **Split** iz parametra *rawData* treba izvući 24 vrijednosti potrošnje, kreirati listu **HourlyData** objekata i dodijeliti je **Data** svojstvu).
    - Klasa **WaterMeter** bi implementirala konstruktor kojem bi proslijedili parametre za *name*, *unit* i *rawData*. Također, „pregazila“ bi metodu **PopulateDailyData** i ponudila svoju implementaciju (uz pomoć metode **Split** iz parametra *rawData* treba izvući 24 vrijednosti potrošnje, s obzirom da su vrijednosti u litrama potrebno je podijeliti ih sa 1000 da dobijemo m<sup>3</sup>, zatim je potrebno kreirati listu **HourlyData** objekata i dodijeliti je **Data** svojstvu).