**Задача 1.** Нека създадем клас в C# описващ структурата на реален обек - Автомобил (Car).

Дефиниция на класа **Car**

* Полета и свойства:
  + Полетата **make**, **model**, **year** и **fuelLevel** съхраняват данните за автомобила.
  + Свойствата предоставят начин за контролирано задаване и извличане на стойности.Например за **Year** се проверява дали е в допустимия интервал.
* Конструктори:
  + Конструктор по подразбиране (default) – инициализира автомобил с предварително зададени стойности.
  + Конструктор с параметри, позволяващ създаването на автомобил с конкретна **марка**(make), **модел**(model) и **година**(year).
  + Трети конструктор позволяващ инициализация с всички атрибути.
* Методи:
  + **Drive** – симулира шофиране, намалява нивото на горивото (fuelLevel) в зависимост от изминатото разстояние.
  + **Refuel** – позволява добавяне на гориво.
  + **DisplayInfo** – извежда информация за автомобила.

Напишете примерен код, в който използвате класа **Car**.

**Задача 2.** Нека създадем клас в C# описващ структурата на реален обек - „умна лампа“ (SmartLight).

Дефиниция на класа **SmartLight**

* Полета и свойства:
  + Полетата **isOn**, **brightness** и **color** съхраняват данните за лампата.
  + Свойствата осигуряват контрол над стойностите и валидират входа. Например **brightness** трябва да е между 0 и 100.
* Конструктори:
  + Конструктор по подразбиране (default), който инициализира лампата като изключена с бял цвят и средна яркост.
  + Конструктор с параметри, позволяващ създаването на лампа с предварително зададена яркост и цвят.
* Методи:
  + **TurnOn и TurnOff** – включват и изключват лампата.
  + **AdjustBrightness** – регулира яркостта.
  + **ChangeColor** – променя цвета на лампата.
  + **DisplayStatus** – извежда информация за текущото състояние.

Напишете примерен код, в който използвате класа **SmartLight**.

**Задача 3.** Нека създадем клас в C# описващ структурата на реален обек - „банкова сметка“ (BankAccount).

Дефиниция на класа **BankAccount**

* Полета и свойства:
  + Полетата **accountNumber**, **owner** и **balance** съхраняват данните за сметката.
  + Свойствата предоставят контролиран достъп до тези полета и валидират входа.
* Конструктори:
  + Един конструктор приема номер на сметката и собственик, и създава сметка с нулев баланс.
  + Втори конструктор позволява създаването на сметка с начален баланс.
* Методи:
  + **Deposit** – позволява депозиране на средства.
  + **Withdraw** – позволява теглене на средства при условие, че има достатъчен баланс.
  + **DisplayAccountInfo** – извежда информация за сметката.

Напишете примерен код, в който използвате класа **BankAccount**.

**Задача 4.** Нека създадем клас в C# описващ структурата на реален обек - продукт (Product).

Дефиниция на класа **Product**

* Полета и свойства:
  + Полетата **name**, **price** и **quantity** съхраняват данните за продукта.
  + Свойствата предоставят контролиран достъп до тези полета и валидират входа.
* Конструктори:
  + Първи конструктор приема име на продукта, задавайки цена по подразбиране 0 и количество 1.
  + Втори конструктор позволява създаването на продукт със зададени име, цена и количество.
* Методи:
  + **Restock**(int amount)– позволява добавянето на ново количество от продукта.
  + **Sell**(int amount) – позволява продажба на продукта, ако има достатъчно наличност.
  + **DisplayProductInfo** – извежда информация за продукта.

Напишете примерен код, в който използвате класа **Product**.

**Задача 5.** Нека създадем клас в C# описващ структурата на реален обек - ученик (Student).

Дефиниция на класа **Student**

* Полета и свойства:
  + Полетата **name**, **grade** и **credits** съхраняват данните за ученик.
  + Свойствата предоставят контролиран достъп до тези полета и валидират входа.
* Конструктори:
  + Първи конструктор приема име на ученик, задавайки оценка по подразбиране 6 и кредит 0.
  + Втори конструктор позволява създаването на ученик със зададени име, оценка и кредити.
* Методи:
  + **Update**(decimal mark)– позволява добавянето на нова оценка и актуализиране на средната оценка.
  + **AddCredits**(int credits) – позволява добавянето на допълнителни кредити към ученика.
  + **DisplayStudentInfo** – извежда информация за ученик.

Напишете примерен код, в който използвате класа **Student**.

**Задача 6.** Нека създадем клас в C# описващ структурата на реален обек - „книга от библиотека“ (LibraryBook).

Дефиниция на класа **LibraryBook**

* Полета и свойства:
  + Полетата **isbn**, **title**, **author** и **isAvailable** съхраняват информация за книгата.
  + Свойствата предоставят контролиран достъп до тези полета и валидират входа:
    - ISBN: уникален идентификатор на книгата.
    - Title: заглавие на книгата.
    - Author: автор на книгата.
    - IsAvailable: статус дали книгата е налична за заемане.
* Конструктори:
  + Първи конструктор приема ISBN, заглавие и автор, като задава статус на книгата като налична (true) по подразбиране.
  + Втори конструктор позволява създаването на книга с посочен статус на наличност.
* Методи:
  + **Borrow**– позволява заемане на книгата, ако е налична.
  + **ReturnBook** – позволява връщане на книгата и отново я прави налична.
  + **DisplayBookInfo** – извежда информация за книгата.

Напишете примерен код, в който използвате класа **LibraryBook**.