

Konstruktory

Celem zadania jest stworzenie prostego symulatora jazdy samochodem. Realizując kolejne podpunkty na każdym kroku upewnij się, że Twój model jest spójny z rzeczywistością!

1. Stwórz nową aplikację konsolową, a następnie dodaj dwie nowe klasy reprezentujące odpowiednio **samochód** oraz **silnik**.
2. Dodaj w definicji **silnika** dwie metody pozwalające na przeliczenie spalania z litrów na 100 km na mile na galon (mpg) i odwrotnie. Metody mają przyjmować spalanie wyrażone jedną wielkością i zwracać wartość przekonwertowaną. Skorzystaj z odpowiedniej metody w funkcji głównej aby sprawdzić jakie jest spalanie Twojego samochodu w mpg.
3. Silnik charakteryzuje **pojemność** oraz **ilość paliwa**, a także **pojemność zbiornika na paliwo**. Zadbaj o to, aby wartości określającej **pojemność baku** nie można było modyfikować po stworzeniu instancji **silnika**.
4. Dodaj w definicji **silnika** pole, które będzie przechowywało **standardową (domyślną) pojemność zbiornika paliwa** dla każdego tworzonego silnika.
5. Dodaj konstruktor, który umożliwi ustawienie **pojemności** oraz **ilości paliwa**. Konstruktor ustawia również **pojemność baku** na **wartość domyślną**.
6. Posłuż się konstruktorem w głównej funkcji programu tworząc przykładowy **silnik**.
7. Dodaj drugi konstruktor, który poza **pojemnością** i **ilością paliwa** pozwoli również na ustawienie **pojemności baku**. Posłuż się wcześniej utworzonym konstruktorem.
8. Skorzystaj z drugiego konstruktora w funkcji głównej.
9. **Samochód** charakteryzowany jest przez **markę**, **model** oraz **silnik**. Posiada również metodę **Jedź()**, która przyjmuje jako parametr dystans do przejechania liczony w km. Metoda działa w następujący sposób:
 - wypisuje na ekran komunikat **Jadę**;
 - odczekuje czas liczony zgodnie z założeniem że **samochód** przejeżdża 1km na 100ms (`Thread.Sleep()`);
 - wywołuje metodę **Działaj()** na rzecz **silnika**, która zmniejsza **ilość benzyny** w baku (przyjmij że **silnik** spala czterokrotność **pojemności silnika** na 100km);
 - wypisuje na ekran komunikat **Jestem**.
10. **Samochód** posiada trzy konstruktory, które umożliwiają podanie następujących wartości
 - **marka, model, pojemność, ilości benzyny, pojemność baku**;
 - **marka, model, pojemność, ilości benzyny**;
 - **marka, model, silnik**.
11. Stwórz nowy **samochód** w funkcji głównej programu i przetestuj jego działanie.
12. Dodaj funkcję **Tankuj()** pozwalającą uzupełnić ilość paliwa w **samochodzie**.
13. Zaimplementuj symulator jazdy samochodem. Na początku działania programu użytkownik podaje niezbędne parametry swojego pojazdu, a następnie może rozpocząć jazdę na wprowadzanych przez

siebie dystansach. Pomiędzy przejazdami może również tankować samochód wskazaną ilością paliwa.

14. Zmień implementację metody Jedź() w taki sposób, aby wyświetlała animację poruszającego się po ekranie samochodu.

Zadanie dodatkowe

15. Zapoznaj się zagadnieniem serializacji w dokumentacji języka C#. W szczególności zwróć uwagę na serializację do formatu binarnego (`BinaryFormatter`) oraz do formatu XML (`XmlSerializer`).
16. Zmodyfikuj program w taki sposób, aby po zakończeniu działania serializował samochód do pliku binarnego, a po ponownym uruchomieniu – deserializował. Przy uruchomieniu programu, po deserializacji użytkownik nie musi już wprowadzać danych pojazdu – można od razu rozpocząć korzystanie z pojazdu.
17. Zmień serializację i deserializację z binarnej na format XML.