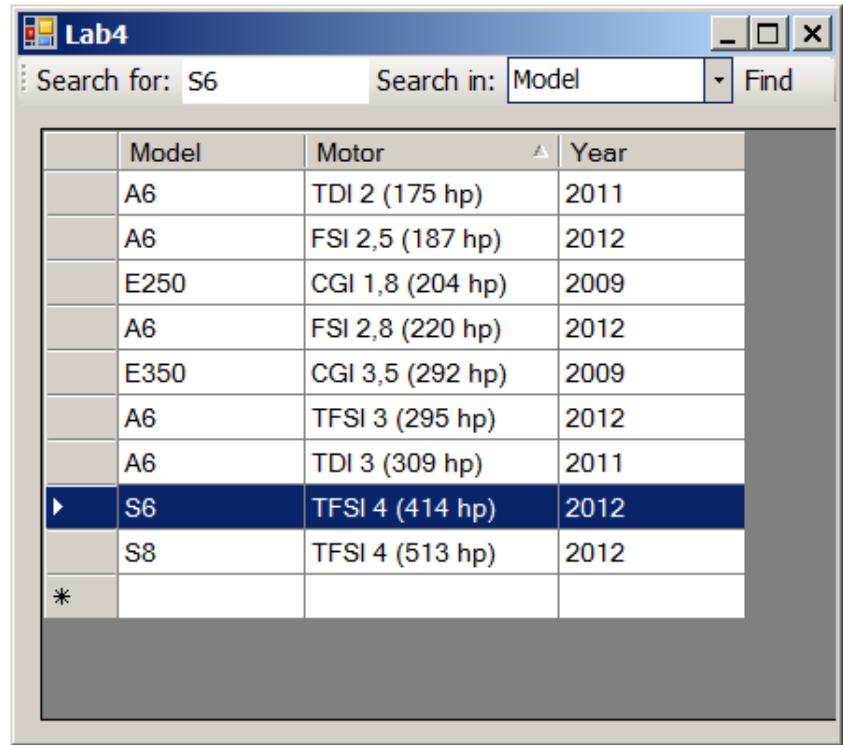


Dokumentacja

- [BindingList<T>](#)
- [BindingSource](#)
- [PropertyDescriptor](#)
- [Sortable BindingList<T>](#)
- [Searchable BindingList<T>](#)

Zadania wstępne

Zaimplementować aplikację Windows Forms lub Windows Presentation Foundation prezentującą elementy kolekcji `myCars` (wrap'owanej przez `BindingList<Car>`) za pomocą kontrolki `DataGridView`. Kontrolka ma umożliwiać dodawanie, usuwanie i edycję elementów oraz sortowanie po wskazanej kolumnie. Dodatkowo zapewnić interfejs do wyszukiwania elementów kolekcji. Wykorzystać implementację `Car` i `Engine` z poprzedniej laborki. Dodatkowo wyposażyć klasę `Engine` w implementację `IComparable` porównującą domyślnie po własności `horsePower`.



How to bind Windows Forms Controls with the BindingSource Component

```
List<Car> myCars = new List<Car>(){
    new Car("E250", new Engine(1.8, 204, "CGI"), 2009),
    new Car("E350", new Engine(3.5, 292, "CGI"), 2009),
    new Car("A6", new Engine(2.5, 187, "FSI"), 2012),
    new Car("A6", new Engine(2.8, 220, "FSI"), 2012),
    new Car("A6", new Engine(3.0, 295, "TFSI"), 2012),
    new Car("A6", new Engine(2.0, 175, "TDI"), 2011),
    new Car("A6", new Engine(3.0, 309, "TDI"), 2011),
    new Car("S6", new Engine(4.0, 414, "TFSI"), 2012),
    new Car("S8", new Engine(4.0, 513, "TFSI"), 2012)
};

BindingList<Car> myCarsBindingList = new BindingList<Car>(myCars);
BindingSource carBindingSource = new BindingSource();
carBindingSource.DataSource = myCarsBindingList;
//Drag a DataGridView control from the Toolbox to the form.
dataGridView1.DataSource = carBindingSource;
```

Punktacja

- Napisać jedno zapytanie LINQ ([przykład składni](#)) do kolekcji `myCars`, używając
 - query expression syntax (0.5 pkt),
 - method-based query syntax (0.5 pkt).

Zapytanie ma zwrócić kolekcję `IEnumerable`, której elementy są typu anonimowego o dwóch własnościach: `engineType` i `avgHPPL`:

- `engineType` ma wartość "diesel" dla samochodów A6 z silnikiem "TDI" a "petrol" dla pozostałych A6;
- `avgHPPL` to średnia arytmetyczna z $\{horsePower / displacement\}$ dla danego `engineType`.

Elementy mają być posortowane malejąco po własności `avgHPPL`.

```
var elements = ... //LINQ query
foreach (var e in elements) Console.WriteLine(e.engineType + ": " + e.avgHPPL);
```

Wykonanie powyższego kodu powinno wyświetlić:

```
diesel: 95,25
petrol: 83,9015873015873
```

- Nie używając wyrażen lambda, dopisz definicje zmiennych `arg1`, `arg2` i `arg3` przed poniższym kodem:

```
myCars.Sort(new Comparison<Car>(arg1));
myCars.FindAll(arg2).ForEach(arg3);
```

- `arg1` ma być instancją delegata `Func`, która sortuje samochody malejąco po mocy silnika (1/3 pkt)
- `arg2` ma być instancją delegata `Predicate`, która zwraca `true` dla samochodów z silnikiem "TDI" (1/3 pkt)
- `arg3` ma być instancją delegata `Action`, która wyświetli każdy element w osobnym `MessageBox` (1/3 pkt)

- Zaimplementować i zademonstrować subclassę `BindingList<T>` umożliwiającą:
 - sortowanie elementów po dowolnej własności typu `K`, pod warunkiem, że `K` implementuje `IComparable` (1 pkt);
wskazówka: `if (property.PropertyType.GetInterface("IComparable") != null) ...`
 - wyszukanie elementu po własności typu `string` lub `Int32` (0.5 pkt)

wskazówka: przeddefiniować metody `SupportsSortingCore`, `ApplySortCore`, `SupportsSearchingCore` i `FindCore`

- Stworzyć:
 - funkcjonalność dodawania, usuwania i edycji elementów kolekcji poprzez `DataGridView` (0.5 pkt)
 - generyczny (niezależny od źródła danych zasilającego `DataGridView`) GUI umożliwiający wyszukiwanie elementów w `DataGridView` po dowolnej własności typu `string` lub `Int32` (1 pkt)
 - przeciągnąć na formularz kontrolkę `ToolStrip`, a następnie "wybrać" z niej elementy `ComboBox`, `TextBox`, `ComboBox` i `Button` jak na rysunku
 - wszystkie własności typu `string` oraz `Int32` powinny zostać załadowane automatycznie do `ComboBox` w momencie zdarzenia `Enter`
 - szukaną wartość dla własności wskazanej w `ComboBox` wpisujemy w pole `TextBox`
 - kliknięcie `Button`'a uruchamia proces wyszukiwania - wywołanie metody `Find` w `BindingList`