

## Zadanie 1. Zabawki.

Opracować aplikację umożliwiającą dodawanie i obsługę zabawek.

1. Napisać minimum 3 interfejsy, każdy z metodą pozwalającą zmieniać parametr zabawki (np. IAccelerate - metoda Accelerate(int change) - zmiana szybkości, IDive - zmiana głębokości, IRise - zmiana wysokości)
2. Napisać minimum 4 klasy zabawek, każda klasa implementuje różne kombinacje interfejsów (np. car, submarine, plane, computer - nie implementuje żadnego interfejsu)
3. Napisać aplikację Windows Forms pozwalającą na a) dodawanie nowych obiektów wybranej klasy b) zmianę parametrów wybranego obiektu. Wykorzystać ComboBox do wyboru typu i obiektu. Rozpoznanie typu obiektu jaki należy utworzyć zrobić dowolnie, rozpoznanie kontrolki, które należy wyświetlić aby zmieniać parametry zabawki zrobić poprzez rozpoznanie implementowanych interfejsów
4. Aplikacja powinna wyświetlać aktualny stan wybranej zabawki (szybkość, głębokość, wysokość itp.)

## Zadanie 2.

Napisać tablicę mogącą się automatycznie rozszerzać

1. Tablica może być np. tablicą obiektów typu int bądź referencji do typu Object.
2. Tablica ma posiadać indeksator do zapisu/odczytu oraz metodę Add do dopisywania elementów na końcu
3. W przypadku dopisywania i konieczności rozszerzenia tablicy należy zarezerwować pamięć na  $2*n$  elementów, gdzie  $n$  to aktualny rozmiar. Rezerwacja nie oznacza tu wpisania do tablicy elementów (czyli zmiany wielkości tablicy) - ważne przy próbie odczytu poza aktualną wielkością tablicy
4. W przypadku odczytu indeksatorem poza aktualną wielkością tablicy rzucamy wyjątek
5. W przypadku zapisu elementu poza aktualną wielkością tablica jest rozszerzana do żadanego indeksu. W tym wypadku wszystkie elementy między poprzednim ostatnim a nowym dodanym traktujemy jako istniejące, z wartością domyślną

## Zadanie 3

Uzupełnić tablicę z zadania 2 o obsługę zdarzeń

1. Zdarzenia są wywoływane w momencie dodania elementu lub zmiany rozmiaru tablicy
2. W argumencie zdarzenia należy przekazać aktualny rozmiar tablicy.

## Zadanie 4

W oparciu o tablicę z zadania 3 napisać program wielowątkowy. Synchronizację wymaganą do poprawnego działania zawrzeć w klasie tablica.

1. Zapewnić bezpieczeństwo wszystkich metod i właściwości dla operacji wielowątkowych
2. Napisać dwie wersje metody Add: jedna z nich (blokująca) czeka z dodaniem elementu aż będzie to możliwe, druga (nieblokująca), jeśli nie jest możliwe dodanie elementu, zwraca kontrolę z odpowiednim wynikiem
3. Utworzyć program z kilkoma wątkami, z których każdy będzie wpisywać swoje dane (np. identyfikator wątku) do tablicy przy pomocy wybranych metod z p. 2. Przy wykorzystaniu metody Add blokującej logować czas oczekiwania na dostęp, przy wykorzystaniu metody Add nieblokującej logować informację o niemożności dodania elementu
4. Zapisać otrzymaną tablicę do pliku tekstowego