

## Задание 1.

Написать приложение, выполняющее следующие функции:

- Ввод с клавиатуры данных о студентах в массив объектов класса Student.
- Вывод списка всех студентов с указанием среднего балла каждого студента в порядке возрастания среднего балла.
- Определение количества студентов, получивших больше двух оценок 10 в массиве.
- Вывод списка двоечников в заданной группе (если таких студентов нет, вывести соответствующее сообщение).

Класс Student должен содержать закрытые поля: фамилия, номер группы, успеваемость (массив оценок) и все необходимые для решения задачи свойства и методы.

Для определения, является ли студент двоечником или отличником, написать методы-расширения.

В программе должны перехватываться все выброшенные исключения.

*Можно также создать класс – контейнер для студентов (по желанию).*

## Задание 2.

Написать приложение «Автомобильные гонки». В гонке участвует от 2 до 7 автомобилей (количество задается пользователем перед началом каждой гонки). Автомобили двигаются по экрану консоли от левого края к правому с переменной скоростью. Победителем гонки считается автомобиль, который первым достиг правого края консоли. Автомобили отображать в консоли с помощью символов псевдографики. Для решения задачи необходимо реализовать класс «Автомобиль», который имеет цвет (красный, синий и т.д. – назначается автомобилю случайным образом в конструкторе), номер (1, 2, и т.д. – задается программой). Предусмотреть возможность поломки автомобиля во время гонок – вероятность поломки – 5%. В случае поломки объект «Автомобиль» генерирует исключительную ситуацию, которая должна быть обработана в программе – поломанный автомобиль перестает двигаться, но остается на экране (отображается на экране как поломанный), и выбывает из гонок. Пользователь перед началом гонок может сделать ставку на один из автомобилей. В случае, если побеждает автомобиль пользователя – программа сообщает «Вы выиграли», иначе – «Вы проиграли». Программа должна иметь меню, предлагающее пользователю сделать ставку и начать новую гонку или выйти из программы.

## Замечания:

1. Для задания цвета фона используйте свойство `BackgroundColor`, цвета текста — `ForegroundColor`.

Например:

```
Console.BackgroundColor = ConsoleColor.Blue;  
Console.ForegroundColor = ConsoleColor.Red;
```

2. Для задания позиции курсора на экране используйте метод `SetCursorPosition`.

Например:

```
Console.SetCursorPosition(30, 23);
```

3. Для определения какой управляющей или символьной клавиша была нажата создайте объект структуры `ConsoleKeyInfo`, используйте метод `ReadKey`.

Например:

```
ConsoleKeyInfo cki;  
cki = Console.ReadKey();  
if (cki.KeyChar == 'y' || cki.KeyChar == 'Y')  
{  
    . . . . .  
}
```