**Тема: Основы программирования на C#.**

**Задание 1. Программа «Тренажер пилота самолета»**

Написать приложение «**Тренажер пилота самолета**». Задача приложения – тренировать начинающих пилотов.

В приложении должен быть реализован класс «**Самолет**» (в процессе тренировки пилотов самолета используется только один объект самолета). Самолет может изменять скорость и высоту.

**Скорость - изменяется клавишами-стрелками Left и Right:**

*(Right: + 50км/ч, Left: -50км/ч, Shift- Right : +150 км/ч, Shift- Left: -150 км/ч).*

**Высота - изменяется клавишами-стрелками Up и Down:**

*(Up: +250м, Down: -250м, Shift- Up: +500м, Shift- Down: -500м).*

Самолет может лететь, если его контролируют минимум 2 диспетчера. Для этого в приложении должен быть создан класс «**Диспетчер**».

Перед началом полета летчик должен ввести диспетчеров, которые должны контролировать его полет (**Полет не может начаться, если диспетчеров меньше двух**). Каждый диспетчер имеет имя (вводится с клавиатуры и передается в конструктор при создании объекта «**Диспетчер**»), и так называемую «**корректировку для погодных условий**». Будем считать, что диспетчера находятся в разных городах, где погода разная, и эта погода влияет на соотношение скорости/высоты полета самолета. Величина корректировки должна генерироваться случайным образом (**см. ниже**). Все объекты-диспетчеры должны размещаться в соответствующей коллекции-члене класса «**Самолет**».

Задача пилота – взлететь на самолете, набрать **максимальную (**1000 км/ч.**)** скорость, а затем посадить самолет. В процессе полета объект-самолет автоматически сообщает всем объектам-диспетчерам все изменения в скорости и высоте полета с помощью **делегатов**. Диспетчер, получив информацию о текущей скорости/высоте самолета, выводит на экран информацию о **рекомендуемой** высоте полета. Рекомендуемая высота полета вычисляется по формуле:

**Hp=7\*Скорость (км/ч) – N,**

где N– «**корректировка погодных условий**». Величина «**корректировки погодных условий**» генерируется случайным образом один раз для каждого диспетчера при создании объекта. Значение величины N выбирается от -200 до +200 метров. Управление самолетом диспетчерами начинается (прекращается) при наборе (при снижении) самолетом скорости более (менее) 50 км/ч. Графики изменения рекомендуемых высот и изменения скорости показаны на рисунке 1.

**Каждый диспетчер, обнаружив разницу между рекомендуемой величиной высоты полета самолета, и текущей величиной начисляет штрафные очки**:

* Если разница в диапазоне от **300 до 600** то пилот получает **25** штрафных очков, если от **600 до 1000** – то 50 очков.
* Если разница превышает **1000**, то объект-диспетчер генерирует исключительную ситуацию «**Самолет разбился**», которая должна быть обработана приложением, как прекращение тренировочного полета с соответствующей информацией на экране.
* Если пилот, не завершив полет, набирает **1000** штрафных очков от любого диспетчера – то этот объект диспетчер генерирует исключительную ситуацию «**Непригоден к полетам**», которая также обрабатывается приложением.
* Так же недопустимо, чтобы самолет в любой момент времени имел нулевую высоту и нулевую скорость (если это случилось – исключение «**Самолет разбился**» генерирует диспетчер). Кроме момента начала взлета и посадки (высота и скорость равны нулю)
* При попытке превысить максимальную скорость диспетчер штрафует пилота на 100 очков и требует немедленно снизить скорость.

В процессе полета самолета, летчик может добавлять новых диспетчеров или удалять уже существующих из коллекции диспетчеров, контролирующих полет. Для этого необходимо предусмотреть соответствующие пункты меню, а также сохранение штрафных очков, полученных от удаляемого диспетчера. (При этом не забываем о правиле, что самолет контролируют не менее 2 диспетчеров)

При успешной посадке самолета, приложение должно перебрать всех диспетчеров в коллекции и просуммировать все штрафные очки в общую сумму и вывести их на экран.

Все команды управлением самолета, а также текущие характеристики полета и рекомендации диспетчеров должны быть на экране.

**Результат решения:**