**«Платформа Microsoft .NET та мова програмування C#»**

**Задание 1 Написать программу. – 2 бала**

Необходимо создать класс — зоомагазин. В классе должны быть следующие поля: животное ( напр. волк, пингвин, собака ), пол, имя , цена, количество. Включить в состав класса необходимый минимум методов, обеспечивающий полноценное функционирование объектов указанного класса:

Конструкторы (по умолчанию, с параметрами, копирования);

Деструктор;

Переопределить возможные для класса операции, продумать порядок их выполнения;

Добавить необходимые методы.

Предоставить возможность вводить данные с клавиатуры или из файла (с помощью конструктора или операцией »).

**Задание 2 Написати програму. – 3 бала**

Описать класс, реализующий десятичный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне. Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, — и свойство, позволяющее получить его текущее состояние. Написать программу, демонстрирующую все возможности класса.

**Задание 3 Написати програму антиплагиат . – 5 балов**

С каждого предложения должно выбираться три подряд стоящих слова в любом городе данного предложения. Данная выборка в дальнейшем проверяется в файлах в заданной папке (файлов должно быть не менее 10). Напечатать процент совпадения.

**Задание 4. «Погода». – 6 балов**

Разработать приложение, позволяющее просматривать погоду на текущую дату выбранного города.

Необходимо реализовать библиотеку, которая загружает xml файл, обрабатывает его и выводит погоду для выбранного города.

Дополнительно:

* найти самую теплую погоду в представленных городах;
* пары «код-имя города» хранить в файле, что бы программа позволяла добавлять новые города, не требуя перекомпиляции;
* сохранять xml-файлы для формирования статистки о погоде за выбранное количество дней (при наличии информации).

**Задание 5. Создать приложение «Словари» – 12 балів.**

Основная задача проекта: хранить словари на разных языках и разрешать пользователю находить перевод нужного слова или фразы. Интерфейс приложения должен предоставлять такие возможности:

■ Создавать словарь. При создании нужно указать тип словаря. Например, англо-русский или русско-английский.

■ Добавлять слово и его перевод в уже существующий словарь. Так как у слова может быть несколько переводов, необходимо поддерживать возможность создания нескольких вариантов перевода.

■ Заменять слово или его перевод в словаре.

■ Удалять слово или перевод. Если удаляется слово, все его переводы удаляются вместе с ним. Нельзя удалить перевод слова, если это последний вариант перевода.

■ Искать перевод слова.

■ Словари должны храниться в файлах.

■ Слово и варианты его переводов можно экспортировать в отдельный файл результата.

■ При старте программы необходимо показывать меню для работы с программой. Если выбор пункта меню открывает подменю, то тогда в нем требуется предусмотреть возможность возврата в предыдущее меню.

**Задание 6. Программа «Тренажер пилота самолета» - 12 балов**

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 : +150км/ч, Shift- Left: -150км/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.