### Контрольная работа 1

Разработать класс, содержащий **лишь** статические методы, производящие обработку данных (первым параметром таких методов должна быть обрабатываемая структура – List<…>). Сами методы ничего не выводят!

**Генерировать и обрабатывать исключительные ситуации. Catch Exception не использовать!**

**Приготовить два текстовых файла (в первом файле input1.txt – объекты первого типа, во втором input2.txt – второго)**. Для исключения проблем с кодировкой в текстовых файлах использовать только латиницу!

**Более предпочтительным вариантом является использование стандартных методов, продемонстрируйте хотя бы раз stream api.**

В оконном swing приложении (более предпочтительный вариант, но допустим и консольный вариант) реализовать команды для открытия и вывода файлов первого и файлов второго вариантов (предпочтительнее через компоненту файлового диалога) в оконную компоненту №1.

Команды обработки данных выводят результат в оконную компоненту №2 (перед каждым выводом чистите ее). Для пятого метода предусмотреть компоненту №3.

Т.о., нужно реализовать следующие классы: абстрактный, два наследника, класс с пятью методами и оконное приложение.

При использовании параметризации ограничить класс своим абстрактным.

К – константа; значения констант в разных классах различны.

Если не указан вариант сравнения, то считается «по возрастанию»; «по убыванию» указано явно.

*Вариант 1 (фамилии А-К)*

**Абстрактный класс** *Дерево:* название, возраст, тип (лиственное/хвойное), метод расчета экономической эффективности.

**Наследник** *ЛесноеДерево* с доп. информацией о количестве древесины; экон. эффективность рассчитывается как

возраст х К х количество древесины.

**Наследник** *ПлодовоеДерево* с доп. информацией о массе урожая и средней продолжительности хранения; экон. эффективность рассчитывается как

возраст х К х масса урожая / средняя продолжительность.

**Методы**:

1. Возврат коллекции в порядке убывания возраста, при равенстве – по названию;
2. Возврат количества деревьев определенного типа;
3. Метод бинарного поиска элемента, сравнивать по всем полям;
4. Возврат коллекции по соотношению эффективность / возраст по убыванию;
5. Возврат списка названий деревьев, начинающихся на ‘С’ (каждое название по одному разу) по убыванию;

*Вариант 2 (фамилии Л-П)*

**Абстрактный класс** *Ученик* : фамилия, название уч. заведения, рейтинг уч. заведения, средняя успеваемость, метод расчета рейтинга ученика.

**Наследник** *Студент* с доп. информацией о курсе; рейтинг рассчитывается как

К1 х рейтинг уч. заведения х средняя успеваемость + К2 х курс.

**Наследник** *Школьник* с доп. информацией о классе и оценке поведения; рейтинг рассчитывается как

К1 х рейтинг уч. заведения х средняя успеваемость х класс + К2 х поведение.

**Методы**:

1. Возврат коллекции в порядке возрастания уч. заведения, при равенстве – по убыванию фамилии;
2. Возврат количества учеников в определенном заведении;
3. Метод бинарного поиска элемента, сравнивать по всем полям;
4. Возврат коллекции по соотношению рейтинг ученика / рейтинг заведения;
5. Возврат списка фамилий учеников, начинающихся на ‘В’ (каждая фамилия по одному разу) по убыванию;

*Вариант 3 (фамилии Р-Я)*

**Абстрактный класс** *Автомобиль* : название, цвет (по числовой шкале), рейтинг, метод расчета цены.

**Наследник** *ЛегковойАвтомобиль* с доп. информацией о максимальной скорости; цена рассчитывается как

К1 х рейтинг х скорость – К2 х цвет.

**Наследник** *Автобус* с доп. информацией о количестве мест и количестве дверей; цена рассчитывается как

К1 х рейтинг х двери + К2 х места – К3 х цвет.

**Методы**:

1. Возврат коллекции по убыванию цены, при равенстве –по убыванию рейтинга;
2. Возврат количества машин определенного цвета;
3. Метод бинарного поиска элемента, сравнивать по всем полям;
4. Возврат коллекции по соотношению цена / цвет;
5. Возврат списка названий автомобилей, начинающихся на ‘Т’ (каждое название по одному разу);