

Уфимский государственный нефтяной технический университет

Кафедра вычислительной техники и инженерной кибернетики

## ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ

Учебно-методическое пособие к курсовой работе для студентов направления 230105  
«Программное обеспечение вычислительной техники и автоматизированных систем»

Уфа – 2016

Составители: Усов Т. М., доцент кафедры ВТИК

## 1. Задание

Задания на курсовую работу приведены в разделе вариантов индивидуальных заданий. Вариант определяется преподавателем при получении задания и фиксируется в журнале выдачи заданий.

Для всех вариантов заданий необходимо создать программу на языке программирования C# с графическим пользовательским интерфейсом (Windows Forms), решающую поставленную задачу.

## 2. Требования к реализации программы и исходному коду

В настоящем разделе приведены *обязательные* требования к реализации программы и исходному коду. Невыполнение любого из требований ведет к снижению оценки на один балл.

Должны применяться возможности наследования классов.

Должны использоваться классы с абстрактными и виртуальными методами.

Должен применяться полиморфизм.

Все входные данные (пути файлов, параметры выполнения и другие) пользователь должен задать в графическом пользовательском интерфейсе. Должна быть предусмотрена защита от указания несуществующих на момент задания или в момент запуска файлов. Также должна быть предусмотрена защита от ввода некорректных входных параметров.

Должны применяться классы из пространства имен System.Collections.Generic.

Должны применяться обработчики исключений.

Работа с файлами должна выполняться с помощью классов из пространства имен System.IO.

Для входных и выходных файлов необходимо использовать собственный формат, который должен быть описан в пояснительной записке. Не допускается использовать XML. Не допускается использовать инструменты сериализации и десериализации для записи и чтения файлов.

Не допускается использование регулярных выражений.

Исходный код программы должен быть написан с учетом принятых в C# соглашений именования и форматирования, содержать необходимый объем комментариев.

## 3. Требования к содержанию и оформлению пояснительной записки

В настоящем разделе приведены *обязательные* требования к содержанию и оформлению пояснительной записки. Невыполнение любого из требований ведет к снижению оценки на один балл.

Пояснительная записка должна содержать разделы анализа, проектирования и реализации.

В разделе анализа необходимо привести анализ поставленной задачи, рассмотреть возможные способы ее решения. Раздел анализа заканчивается подразделом постановки задачи, содержащим исчерпывающее детальное описание требований к функциональным возможностям разрабатываемой программы, выработанное в ходе анализа.

В разделе проектирования необходимо привести описание архитектуры разрабатываемой программы, диаграмму классов, проект пользовательского интерфейса.

В разделе реализации должны быть представлены сведения о реализации, а также подразделы тестирования разработанной программы.

Пояснительная записка должна содержать приложение с исходным кодом (если вместе с приложением суммарный объем пояснительной записки превосходит 30 страниц, то приводится фрагмент исходного кода – ключевые его части).

## 4. Варианты индивидуальных заданий (2016–2017 уч. год)

1. Входной текстовый файл содержит команды работы с файлами: переименование файла, переименование папки, удаление файла, удаление папки со всем содержимым. Приложение должно выполнять команды этого входного файла и записывать отчет о выполнении в другой текстовый файл.

2. Входной текстовый файл данных содержит показания датчиков температуры (код датчика – время измерения – значение измерения). Входной текстовый файл фильтров содержит условия выявления событий (событие – условие – параметры условия). Должны поддерживаться следующие условия: превышение заданного значения на указанную величину абсолютное и относительное, отклонение от

заданного значения на указанную величину абсолютное и относительное. Приложение должно анализировать файл данных и выявлять перечень событий согласно файлу фильтров.

3. Входной текстовый файл содержит серию координат пути объекта. Файл зон содержит описание прямоугольных зон. Файл фильтров содержит условия: выход за границы указанной зоны, попадание в границы указанной зоны, пребывание внутри или вовне границы указанной зоны более указанного времени. Приложение должно анализировать файл данных и выявлять перечень событий согласно файлу фильтров.

4. В нескольких входных текстовых файлах задаются данные таблиц (каждый файл – отдельная таблица). В отдельном файле записываются команды выборки данных из этих таблиц. Реализовать программу, выполняющую эти команды и записывающую результат в новый файл. Поддерживаемые команды: левое соединение (left join) по одному полю, фильтрация (where) по одному полю, поддерживаемые условные операторы: меньше, больше, равно.

5. Поиск файлов в папки по заданным во входном файле критериям. Поддерживаемые критерии: маска имени, маска пути, размер (больше, меньше, равно), формат (по расширению), полнотекстовый поиск. Должна быть реализована поддержка возможности объединения условий операторами И, ИЛИ. Перечень найденных файлов необходимо записать в другой файл.

6. Реализовать эмулятор сигналов датчика (время – значение). Во входном файле задаются параметры сигнала, в выходном – серия значений. Поддерживаемые виды: сигнал на базе синусоиды, сигнал по линейной функции с интервалом повтора, в обоих случаях параметры: смещение, коэффициент, случайное отклонение ИЛИ начальное значение, минимум, максимум, случайное отклонение.

7. В нескольких входных текстовых файлах задаются данные таблиц (каждый файл – отдельная таблица). В отдельном файле записываются команды обновления данных в этих таблицах. Реализовать программу, выполняющую эти команды. Поддерживаемые команды: обновление условное по одному полю (update...where), условные операторы: меньше, больше, равно; вставка одной строки (insert).

8. Входной текстовый файл содержит команды работы с файлами: переименование файла, переименование папки, копирование файла, копирование папки со всем содержимым. Приложение должно выполнять команды этого входного файла и записывать отчет о выполнении в другой текстовый файл.

9. Входной текстовый файл данных содержит показания датчиков температуры (код датчика – время измерения – значение измерения). Входной текстовый файл фильтров содержит условия выявления событий (событие – условие – параметры условия). Должны поддерживаться следующие условия: превышение изменения за указанное время заданного значения на указанную величину абсолютное и относительное, отклонение изменения за указанное время от заданного значения на указанную величину абсолютное и относительное. Приложение должно анализировать файл данных и выявлять перечень событий согласно файлу фильтров.

10. Входной текстовый файл содержит серию координат пути объекта. Файл зон содержит описание прямоугольных зон. Файл фильтров содержит условия: последовательность пребывания в зонах в течение заданного периода или в рамках заданной длины пути. Приложение должно анализировать файл данных и выявлять перечень событий согласно файлу фильтров.

11. В нескольких входных текстовых файлах задаются данные таблиц (каждый файл – отдельная таблица). В отдельном файле записываются команды выборки данных из этих таблиц. Реализовать программу, выполняющую эти команды и записывающую результат в новый файл. Поддерживаемые команды: объединение (union), фильтрация (where) по одному полю, поддерживаемые условные операторы: меньше, больше, равно, сравнение текста по шаблону (like).

12. Реализовать эмулятор сигналов датчика (время – значение). Во входном файле задаются параметры сигнала, в выходном – серия значений. Поддерживаемые виды: сигнал по линейной функции с интервалом повтора (смещение, коэффициент, случайное отклонение ИЛИ начальное значение, минимум, максимум, случайное отклонение), сигнал по заданной серии значений (случайное отклонение, режим повтора: с начала или в обратном направлении).

13. Реализовать эмулятор сигналов датчика расположения объекта (время – значение). Во входном файле задаются параметры сигнала, в выходном – серия значений. Поддерживаемые параметры: серия координат (путь), по которому объект перемещается (режим повтора: с начала или в обратном направлении) направлении с переменной скоростью (параметры: минимум, максимум, тип изменения скорости: синусоида, линейная функция).

14. Во входном файле задан текст. В другом входном файле заданы команды его обработки. Реализовать программу, выполняющую эти команды. Поддерживаемые команды: изменить все буквы после точек на заглавные, все на заглавные, все на малые, удалить все множественные пробелы (более одного пробела подряд заменить на один). В отдельный выходной файл записать статистику встречаемости каждого символа.

15. Во входном файле задан текст. В другом входном файле заданы команды его обработки. Реализовать программу, выполняющую эти команды. Поддерживаемые команды: замена текста по словарю, замена с учетом и без учета регистра, разбиение на нескольких файлах по указанным последовательностям символов. В отдельный выходной файл записать статистику встречаемости каждого символа.

16. В файле задан путь автомобиля, полученный с датчика местоположения (время координаты). Реализовать программу, выявляющую в этом файле следующие события (параметры программы должны задаваться в отдельном входном файле): превышение заданной скорости, промежутки стоянки, превышение заданного ускорения, превышение заданной скорости более чем в течение заданного времени.