

Контрольное домашнее задание № 3, модуль 2

Контрольное домашнее задание предполагает самостоятельную домашнюю работу. Вам потребуется:

1. Изучить некоторые теоретические материалы самостоятельно.
2. Самостоятельно поработать с документацией по языку C#, в т.ч. осуществлять информационный поиск.
3. Разработать программы, определённые основной задачей и индивидуальным вариантом.
4. Сдать в SmartLMS вовремя заархивированный проект с кодом проекта консольного приложения и библиотеки классов, определённые заданием и вариантом.

Время выполнения

На выполнение работы отводятся две недели, точные даты выполнения устанавливаются в SmartLMS.

Формат сдачи работы

Для проверки предоставляется **решение, содержащие два проекта**: консольное приложение и библиотеку классов. Решение должно быть заархивировано и приложено в качестве ответа на задание в SmartLMS.

Общее задание

Разрабатываемая в рамках контрольного домашнего задания программа предназначена для работы с данными **csv**-файла и должна быть реализована с использованием объектно-ориентированной парадигмы. **Соблюдение принципов единственной ответственности, принципа подстановки Лисков и инверсии зависимостей обязательно.**

В одном решении разместить проект библиотеки классов и проект консольного приложения. Дополнительные требования к библиотеке и приложению см. в индивидуальном варианте.

Требования к библиотеке классов

1. Обязательно присутствуют типы данных, заданные индивидуальным вариантом (таблица 2).
2. Реализации классов не должны нарушать инкапсуляцию данных и принцип единственной ответственности (Single Responsibility Principle).
3. Иерархии не должны нарушать принципа подстановки Лисков (Liskov Substitution Principle) и проектируются, исходя из соблюдения принципа инверсии зависимостей (Dependency Inversion Principle).
 - a. Архитектурные принципы (<https://learn.microsoft.com/ru-ru/dotnet/architecture/modern-web-apps-azure/architectural-principles>)

4. Реализации классов должны содержать регламентированный доступ к данным.
5. Классы библиотеки должны быть доступны за пределами сборки.
6. Каждый нестатический класс обязательно должен содержать, в числе прочих, конструктор без параметров или эквивалентные описания, допускающие его прямой вызов или неявный вызов.
7. Запрещено изменять набор данных (удалять / дополнять), хранящихся в классах или не использовать указанные в задании открытые варианты поведения.
8. Допускается расширение открытого поведения или добавление закрытых функциональных членов класса.
9. Допускается использование собственных (самописных) иерархий классов в дополнение к предложенным в индивидуальном варианте, но требования по ООП принципам к ним сохраняются.
10. Допускается использование абстрактных типов данных, таких как List, ArrayList, Set, Stack, их обобщённых реализаций и проч., но не в качестве замены определённых вариантов структур данных.
11. Поскольку в описаниях классов присутствует «простор» для принятия решений, то каждое такое решение должно быть описано в комментариях к коду программы. Например, если выбран тип исключения, то должно быть письменно обосновано, почему вы считаете его наиболее подходящим в рамках данной задачи.

Требования к консольному приложению

1. Консольное приложение должно использовать типы данных, заданные индивидуальным вариантом (таблица 2) и позволять (в консольном меню):
 - a. Открыть *.csv файл с исходными данными. Структура файла определена файлом индивидуального варианта (таблица 1). В случае ошибок открывания файла или некорректных данных программа должна выводить пользователю сообщение. Пути к файлам получать от пользователя.
 - b. Предоставить для просмотра N первых (top) или последних записей (bottom) из файла ($N > 1$ выбирается пользователем и не превышает количества записей в csv файле). Выбор первых или последних записей организовать с использованием перечисления (enum, <https://learn.microsoft.com/ru-ru/dotnet/csharp/language-reference/builtin-types/enum>).
 - c. Выполнить сортировку или фильтрацию по полям, указанным в индивидуальном варианте (таблица 3). Значения полей для фильтрации вводятся пользователем с клавиатуры.
 - d. Сохранять результаты сортировок и фильтров в файл формата csv. Сохранение производится обязательно после выполнения сортировки / фильтрации. В случае ошибок открывания / сохранения файла или некорректных данных, программа должна выводить сообщение. Структура файла для сохранения данных совпадает со структурой файла, заданного индивидуальным вариантом.
 - e. Режимы сохранения в файл. Режим позволяет по-разному сохранять данные в файл. Выбор режима остаётся за пользователем, потребуется организовать с ним диалог. Пути к файлам получать от пользователя.
 - i. создание нового файла.
 - ii. замена содержимого уже существующего файла.

- iii. добавление сохраняемых данных к содержимому существующего файла.
2. Предусмотреть **проверку корректности для каждого ввода данных**, обработку исключений, в т.ч. порождаемых классами библиотеки, **организацию повторения решения**.
3. Допускается использование абстрактных типов данных, таких как List, ArrayList, Set, Stack, их обобщённых реализаций и проч., но не в качестве замены определённых вариантов структур данных.
4. Все данные приложения читают из **csv** файлов, результат работы **и выводится на экран, и сохраняется в csv файле**. Программа должна уметь читать, созданные ей при **сохранении данных файлы**.
5. Программа обязательно должна читать все созданные ей самой файлы без ошибок.

Общие требования к работе

1. **Цикл повторения решения и проверки корректности получаемых данных обязательны**.
2. Соблюдение определённых программой учебной дисциплины требований к программной реализации работ – обязательно.
3. Соблюдение соглашений о качестве кода – обязательно (<https://learn.microsoft.com/ru-ru/dotnet/csharp/fundamentals/coding-style/coding-conventions>).
4. Весь программный код должен быть написан на языке программирования C# с учётом использования .net 6.0;
5. исходный код должен **содержать комментарии, объясняющие неочевидные фрагменты и решения**, резюме кода, описание целей кода (см. материалы [лекции 1](#), модуль 1);
6. при перемещении папки проекта библиотеки (копировании / переносе на другое устройство) **файлы должны открываться программой также успешно, как и на компьютере создателя, т.е. по относительному пути**;
7. текстовые данные, включая данные на русском языке, успешно декодируются при представлении пользователю и человекочитаемы;
8. программа не допускает пользователя до решения задач, пока с клавиатуры **не будут введены корректные данные**;
9. консольное приложение обрабатывает исключительные ситуации, связанные (1) со вводом и преобразованием / приведением данных как с клавиатуры, так и из файлов; (2) с созданием, инициализацией, обращением к элементам массивов и строк; (3) вызовом методов библиотеки.
10. представленная к проверке библиотека классов должна решать все поставленные задачи, успешно компилироваться.
11. **Запрещено** использование сторонних библиотек, файлов исходного кода, NuGet-пакетов.

Что сдаём?

Архив с папкой решения (Solution), содержащий проекты консольного приложения и библиотеки классов, определённых индивидуальным вариантом.

Данные для индивидуальных вариантов

Таблица 1. Исходные файлы и краткое описание

№ варианта	Файл с исходными данными	Описание
1	ЗАГС-инфо.csv	Органы ЗАГС Москвы версия 2.4 от 02.10.2014
2	ОП охраны порядка.csv	Общественные пункты охраны порядка версия 1.1 от 16.12.2014
3	Штабы городских народных дружин.csv	Штабы городских народных дружин версия 1.2 от 05.06.2014
4	Театры.csv	Театры версия 2.5 от 29.10.2014
5	Средняя образовательная школа.csv	Средняя образовательная школа версия 1.1 от 31.10.2013
6	Кинотеатры.csv	Кинотеатры версия 2.3 от 14.10.2014.csv
7	Естественнонаучное направление.csv	Образовательные учреждения естественнонаучного направления версия 1.1 от 31.10.2013
8	Диспансеры версия.csv	Диспансеры версия 2.4 от 13.10.2014
9	Детские сады.csv	Детские сады версия 1.1 от 31.10.2013
10	Поликлиническая помощь взрослым.csv	Поликлиническая помощь взрослым версия 2.4 от 20.10.2014
11	Банкоматы.csv	Банкоматы «Сбербанка»
12	МосГАЗ.csv	Привязки улиц г. Москва к отделению обслуживания МосГаза
13	Реестр адвокатов.csv	Данные о членах коллегии адвокатов
14	Реестр нотариусов Москвы.csv	Данные о нотариусах города Москвы
15	Колледжи.csv	Колледжи 1.1 состояние на 31.10.2013

Таблица 2. Обязательные используемые типы данных для представления данных из csv-файла

№ варианта	Обязательный тип 1	Обязательный тип 2
------------	--------------------	--------------------

1	Основная информация о ЗАГСе хранится в объектах класса «ЗАГС». Набор полей класса задаётся полями csv файла, кроме полей, содержащих адрес. Адрес представлен полем обязательного типа «Адрес». Класс «ЗАГС» находится в отношении композиции с классом «Адрес».	Структура «Адрес» представляет адреса (город, улица, номер дома, индекс и т.п.). Для всех указанных элементов тип предусматривает отдельное поле.
2	Основная информация о Пунктах охраны правопорядка хранится в объектах класса «Пункт Охраны Правопорядка». Набор полей класса задаётся полями csv файла, кроме полей, содержащих адрес и телефон. Адрес представлен полем обязательного типа «Контакты». Класс «Пункт Охраны Правопорядка» находится в отношении композиции с классом «Контакты».	Класс «Контакты» представляет адрес (индекс, город, улица, номер дома) и телефон. Для всех указанных элементов тип предусматривает отдельное поле.
3	Основная информация о Штабах городских народных дружин хранится в объектах класса «Штаб». Набор полей класса задаётся полями csv файла, кроме полей, содержащих адрес и телефон. Адрес представлен полем обязательного типа «Контакты».	Структура «Контакты» представляет адрес (индекс, город, улица, номер дома) и телефон. Для всех указанных элементов тип предусматривает отдельное поле.
4	Основная информация о Театрах города Москва хранится в объектах класса «Театр». Набор полей класса задаётся полями csv файла, кроме полей, содержащих адрес и телефон. Адрес представлен полем обязательного типа «Контакты». Класс «Театр» находится в отношении композиции с классом «Контакты».	Класс «Контакты» представляет адрес (индекс, город, улица, номер дома) и телефон. Для всех указанных элементов тип предусматривает отдельное поле.
5	Основная информация о Средних образовательных школах хранится в объектах класса «Школа». Набор полей класса задаётся полями csv файла, кроме полей, содержащих адрес и телефон. Адрес представлен полем обязательного типа «Контакты».	Структура «Контакты» представляет адрес (индекс, город, улица, номер дома), электронная почта, веб-сайт и телефон. Для всех указанных элементов тип предусматривает отдельное поле.
6	Основная информация о Кинотеатрах хранится в объектах класса «Кинотеатр». Набор полей класса задаётся полями csv файла, кроме полей, содержащих адрес, телефон(ы), факс, электронную почту. Эти поля представлены полем обязательного типа «Контакты». Класс «Кинотеатр» находится в отношении композиции с классом «Контакты».	Класс «Контакты» представляет адрес (индекс, город, улица, номер дома), телефон, факс и электронную почту. Для всех указанных элементов тип предусматривает отдельное поле.
7	Основная информация о Образовательных учреждениях естественнонаучного направления хранится в объектах класса	Структура «Контакты» представляет адрес (индекс (если есть), город, улица, номер дома), электронная почта, веб-

	«Кружок». Набор полей класса задаётся полями csv файла, кроме полей, содержащих адрес, телефон, электронная почта и веб-сайт. Эти поля представлены полем обязательного типа «Контакты».	сайт и телефон. Для всех указанных элементов тип предусматривает отдельное поле.
8	Основная информация о Диспансерах города Москвы хранится в объектах класса «Диспансер». Набор полей класса задаётся полями csv файла, кроме полей, содержащих адрес, телефон и факс. Эти данные файла представлены полем обязательного типа «Адрес». Класс «Диспансер» находится в отношении композиции с классом «Адрес».	Класс «Адрес» представляет адрес (индекс, город, улица, номер дома, телефон и факс). Для всех указанных элементов тип предусматривает отдельное поле.
9	Основная информация о Детских садах города Москвы хранится в объектах класса «Детский сад». Набор полей класса задаётся полями csv файла, кроме полей, содержащих адрес, телефон, электронную почту и веб-сайт. Эти данные представлены полем обязательного типа «Адрес». Класс «Детский сад» находится в отношении композиции с классом «Адрес».	Класс «Адрес» представляет адрес (адрес, телефон, электронную почту и веб-сайт). Для всех указанных элементов тип предусматривает отдельное поле.
10	Основная информация о Взрослых поликлиниках хранится в объектах класса «Поликлиника». Набор полей класса задаётся полями csv файла, кроме полей, содержащих адрес, факс, телефон, и электронную почту. Эти данные представлены полем обязательного типа «Контакты».	Структура «Контакты» представляет адрес (индекс, город, улица, номер дома, факс, телефон и электронная почта). Для всех указанных элементов тип предусматривает отдельное поле.
11	Основная информация о Банкоматах сбербанка хранится в объектах класса «Банкомат». Набор полей класса задаётся полями csv файла, кроме полей, содержащих информацию об адресе и месте установки. Эти данные представлены полем обязательного типа «Размещение».	Структура «Размещение» представляет название адрес банкомата (город, улица, номер дома), место установки и контакты отделения банка (название, телефон, идентификатор). Для всех указанных элементов тип предусматривает отдельное поле.
12	Основная информация о Привязке улицы к отделению обслуживая МосГаза хранится в объектах класса «МосГаз». Набор полей класса задаётся полями csv файла, кроме полей, содержащих информацию об административном округе и районе. Эти данные представлены полем обязательного типа «Административная единица». Класс «МосГаз» находится в отношении агрегации с классом «Административная единица», учитывается, что если	Класс «Административная единица» хранит данные о (округе и районе). Для всех указанных элементов тип предусматривает отдельное поле.

	административная единица уже есть, как экземпляр, то новая не создаётся, т.е. несколько объектов «МосГаз» могут быть связаны с одним объектом «Административная единица».	
13	Основная информация о Членах коллегии адвокатов хранится в объектах класса «Адвокат». Набор полей класса задаётся полями csv файла, кроме полей с реквизитами решения о присвоении статуса адвоката, которые представлены полем обязательного типа «Реквизиты»	Структура «Реквизиты» представляет номер постановления и дату. Для всех указанных элементов тип предусматривает отдельное поле.
14	Основная информация о Нотариусах города Москвы хранится в объектах класса «Нотариус». Набор полей класса задаётся полями csv файла, кроме полей, содержащих адрес улица, номер дома, индекс) и телефон. Эти данные представлены полем обязательного типа «Контакты». Класс «Нотариус» находится в отношении композиции с классом «Контакты».	Класс «Контакты» представляет адрес (улица, номер дома, индекс) и телефон (ы). Для всех указанных элементов тип предусматривает отдельное поле.
15	Основная информация о Колледжах города Москвы хранится в объектах класса «Колледж». Набор полей класса задаётся полями csv файла, кроме полей, содержащих индекс, город, улица, номер дома), электронная почта, веб-сайт и телефон. Эти данные представлены полем обязательного типа «Контакты».	Структура «Контакты» представляет адрес (индекс, город, улица, номер дома), электронная почта, веб-сайт и телефон

Таблица 3. Фильтрации и сортировки по вариантам

№ варианта	Данные, по которым производится сортировка и тип сортировки	Поля, по которым производится фильтрация
1	поля: AdmArea Сортировать объекты типа ЗАГС, в порядке возрастания или убывания в зависимости от значения поля “Административный округ”. Т.е. ЗАГСы упорядочены по алфавиту, если административные округа идут по алфавиту и по убыванию, если административные округа отсортированы в обратном алфавитному порядке.	AdmAreaCode, AdmArea
2	поля: AdmArea	District, AdmArea

	Сортировать объекты типа «Пункт Охраны Правопорядка», в порядке возрастания или убывания в зависимости от значения поля “Административный округ”. Т.е. пункты охраны упорядочены по алфавиту, если административные округа идут по алфавиту и по убыванию, если административные округа отсортированы в обратном алфавитному порядке.	
3	поля: AdmArea Сортировать объекты типа «Штаб», в порядке возрастания или убывания в зависимости от значения поля “Административный округ”. Т.е. пункты охраны упорядочены по алфавиту, если административные округа идут по алфавиту и по убыванию, если административные округа отсортированы в обратном алфавитному порядке.	District, AdmArea
4	поля: MainHallCapacity, AdditionalHallCapacity Сортировать значения объекта типа «Театр», в порядке возрастания или убывания в зависимости от его вместимости. Т.е. театры упорядочены по возрастанию, если общая вместимость (сумма основной и дополнительной) возрастает и по убыванию, если значение общей вместимости убывает.	ChiefName, AdmArea
5	поля: okrug Сортировать объекты типа «Школа», в порядке возрастания или убывания в зависимости от значения поля “округ”. Т.е. школы упорядочены по алфавиту, если районы идут по алфавиту и по убыванию, если округа отсортированы в обратном алфавитному порядке.	form_of_incorporation, submission
6	поля: TotalSeatsAmount Сортировать объекты типа «Кинотеатр», в порядке возрастания или убывания в зависимости от его вместимости. Т.е. кинотеатры упорядочены по возрастанию, если общее количество мест возрастает и по убыванию, если значение общего количества мест убывает.	NumberOfHalls, ChiefOrg

7	<p>поля: rayon</p> <p>Сортировать объекты типа «Кружок», в порядке возрастания или убывания в зависимости от значения поля “район”. Т.е. кружки упорядочены по алфавиту, если районы идут по алфавиту и по убыванию, если районы отсортированы в обратном алфавитному порядке.</p>	form_of_incorporation, vid_uchrezhdeniya
8	<p>поля: District</p> <p>Сортировать объекты типа "Диспансер", в порядке возрастания или убывания в зависимости от значения поля “Административный округ”. Т.е. адреса упорядочены по алфавиту, если административные округа идут по алфавиту и по убыванию, если административные округа отсортированы в обратном алфавитному порядке.</p>	Specialization, ChiefPosition
9	<p>поля: rayon</p> <p>Сортировать объекты типа «Детский сад», в порядке возрастания или убывания в зависимости от значения поля “район”. Т.е. детские сады упорядочены по алфавиту, если районы идут по алфавиту и по убыванию, если районы отсортированы в обратном алфавитному порядке.</p>	form_of_incorporation, vid_uchrezhdeniya
10	<p>поля: District</p> <p>Сортировать объекты типа «Поликлиника», в порядке возрастания или убывания в зависимости от значения поля “Административный округ”. Т.е. поликлиники упорядочены по алфавиту, если административные округа идут по алфавиту и по убыванию, если административные округа отсортированы в обратном алфавитному порядке.</p>	PaidServicesInfo, AdmArea
11	<p>поля: installplace</p> <p>Сортировать объекты типа «Банкомат», в порядке возрастания или убывания в зависимости от значения поля “место установки”. Т.е. банкоматы упорядочены по алфавиту, если места установки идут по</p>	american_express, access

	алфавиту и по убыванию, если места установки отсортированы в обратном алфавитному порядке.	
12	поля: okrug Сортировать объекты типа «МосГаз», в порядке возрастания или убывания в зависимости от значения поля “округ”. Т.е. банкоматы упорядочены по алфавиту, если округа идут по алфавиту и по убыванию, если округа отсортированы в обратном алфавитному порядке.	streetname, area
13	поля: Фамилия Сортировать объекты типа «Адвокат», в порядке возрастания или убывания в зависимости от значения поля “Фамилия”. Т.е. адвокаты упорядочены по алфавиту, если их фамилии идут по алфавиту и по убыванию, если фамилии отсортированы в обратном алфавитному порядке.	Приостановление статуса адвоката, Возобновление статуса адвоката
14	поля: address Сортировать объекты типа «Нотариус», в порядке возрастания или убывания в зависимости от значения улицы. Т.е. нотариусы упорядочены по возрастанию, если улицы идут по алфавиту и по убыванию, если улицы отсортированы по алфавиту в обратном порядке.	fullname, metrostations
15	поля: rayon Сортировать объекты типа «Колледж», в порядке возрастания или убывания в зависимости от района. Т.е. колледжи упорядочены по возрастанию, если районы отсортированы по алфавиту и по убыванию, если значения райнов отсортированы в обратном алфавитному порядке.	form_of_incorporation, submission