Основы технологии программирования .NET

Практические задания 2019-2020гг

Прежде, чем приступить к выполнению заданий.

Студент имеет право задавать вопросы **ДО** того, как приступит к сдаче практического задания.
Преподаватель имеет право задавать вопросы **ПРИ** сдаче студентом практического задания. В случае сомнений в знаниях студента преподаватель имеет право выдать дополнительное задание.

Важно! Студент должен быть в состоянии объяснить все элементы своего программного кода, а также знать теоретический материал данной области. Пример: если в коде используется переменная типа Int32, необходимо знать различия между int, Int32, Int64 и аргументировать свой выбор.
Все имена полей, свойств, классов, методов должны быть понятны, читаемы и задокументированы.
В каждом задании требуется демонстрация реализованного кода с консольным выводом, что позволяет проверить и протестировать правильность реализации предметной области и задания. Класс main не относится к разрабатываемым архитектурам ООП, но через него можно взаимодействовать с реализованной архитектурой.

Задание 1. ООП.

Разработать набор классов, представляющих собой абстракцию над предметной областью, с использованием языка программирования C#.
Однозначно определить свою предметную область, выбрав ее из списка или придумав самостоятельно (в таком случае предметную область нужно согласовать с преподавателем), и реализовать проект, продемонстрировав работу с программой.

№	Предметная область
1	Автомобилестроение
2	Компьютерная техника
3	Биология
4	Химия
5	Физика
6	Управление предприятием
7	Конфетная фабрика
8	Лингвистика
9	Танкостроение
10	Авиостроение
11	Музыка
12	Живопись
13	Литература

14	Базы данных
15	Free theme
16	Языки программирования
17	Приложения
18	Операционные системы
19	Архитектура (IT)
20	Архитектура
21	Дизайн
22	География
23	Геология
24	Математика
25	Военная сфера

Условия задачи:

- Однозначно определить свою предметную область (здесь и далее п.о.) и согласовать с преподавателем
- Описать п.о. в файле description.txt в корне проекта. Должна быть разработана логика взаимодействия объектов п.о.
- Сформировать структуру классов, описывающих выбранную п.о. При проектировании можно использовать паттерны проектирования, например, Абстрактная фабрика.
- В рамках ООП архитектуры должна быть реализована как минимум одна иерархия
- Проект должен содержать более 8 классов (из них как минимум 4 класса не должны относиться к реализованной иерархии классов). В том числе абстрактный(ые) класс(ы).
- Реализовать как минимум один ООП интерфейс
- Использовать getters и setters как минимум в одном классе
- Показать и объяснить использование инкапсуляции, полиморфизма и наследования.

Задание 2. Обобщения (generics).

Разработать собственную обобщённую коллекцию в рамках выбранной п.о. и внедрить в проект.

Условия задачи:

- Поддержка как минимум одного интерфейса из следующих интерфейсов: ICollection, IClonable, IEnumerator, IEnumerable
- Внедрить поддержку обобщений в проект.
- Продемонстрировать ковариантность и контравариантность обобщённых интерфейсов.
- Добавить одно ограничение при обобщении.

Задание 3. Делегаты.

К разработанной в практическом задании №2 коллекции добавить возможность сортировки и сравнения элементов. Условия сравнения элементов задаются из внешнего по отношению к классу-коллекции источника.

Условия задачи:

- Использование делегатов
- Использование классов Action и Func

Задание 4. События.

Добавить в проект логирование основных этапов выполнения программы.

Условия задачи:

- Использовать отдельный класс для логирования с обобщенными методами.
- Класс должен поддерживать два источника вывода: консоль и файл. Для пользователя должен быть единый интерфейс.
- Метод непосредственной печати лога должен находиться во внешнем источнике (классе). В самом классе должно быть описано только событие.

Задание 5. Исключения.

Разработать класс исключений для проекта, логирующий внештатные ситуации.

Условия задачи:

- Добавить (если ещё не было сделано) файл какой-либо конфигурации к проекту, оставаясь в рамках п.о.
- Производить считывание конфигурации из файла
- Обеспечить выброс исключений в случаях ошибок при чтении\записи файлов и других ошибок в ходе выполнения программы
- Разделить обработку стандартных исключений и пользовательских (минимум два стандартных исключения и одно пользовательское)
- Разработать диаграмму классов для текущего состояния проекта

Задание 6. Потоки.

Обеспечить обработку сортировки пользовательской коллекции отдельным потоком.

Условия задачи:

- Сделать сортировку коллекции асинхронной операцией
- Вынести логирование сортировки (сообщения о старте сортировки, о выполнении сортировки, сколько элементов было обработано (служебная информация)) в отдельный поток
- Обеспечить взаимодействие двух потоков

Задание 7. Сериализация.

Разработать набор классов для сериализации и десериализации пользовательской коллекции в различные форматы данных

Условия задачи:

- Должен быть единый интерфейс
- Продемонстрировать интерфейс на поддержке форматов XML и JSON (изначально «программные заглушки»)
- Поддержка формата XML: реализовать работу с форматом данных XML

Задание 8

Обеспечить покрытие тестами кода проекта

Условия задачи:

- Написать не менее 20 тестов и обеспечить покрытие основного кода бизнес-логики продукта (проекта)
- Использовать при написании тестов атрибуты Test, TestFixture, SetUp и TearDown
- Использовать утверждения – методы из класса Assert (не менее 4 различных методов и, соответственно, примеров их применения)