

# Основы технологии программирования .NET

## Практические задания 2022-2023гг

### 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.

Нельзя сдавать два задания в одном проекте одновременно.

### Задание 1. ООП.

Разработать набор классов, представляющих собой абстракцию над предметной областью (п.о.), 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. Приложение консольное (консольные интерфейсы).

№	Предметная область
1	Автомобилестроение
2	Компьютерная техника
3	Биология
4	Химия
5	Пчелиная ферма
6	Управление предприятием
7	Конфетная фабрика
8	Пекарня (кулинария)
9	Танкостроение
10	Авиастроение
11	Музыка
12	Живопись

13	Литература
14	Базы данных
15	Free theme (свободная тема)
16	Языки программирования
17	Огород/садоводство
18	Зоопарк
19	Парк аттракционов
20	Космос
21	Продуктовый магазин
22	География
23	Салон красоты
24	Оркестр (концерт)
25	Военная сфера
26	Футбол
27	Столовая (буфет)

Описание особенностей реализации по темам размещено после заданий [ниже].

#### **Условия задачи:**

- Однозначно определить свою предметную область (здесь и далее п.о.) и согласовать с преподавателем
- Соответствие требованиям, изложенным в описании особенностей реализации предметной области в соответствии с темой (описание представлено после заданий)
- Описать п.о. в файле description.txt в корне проекта (рекомендуется). 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). Интерфейс взаимодействия с приложением в данный список не входит и сложность (объем реализации) зависит от автора реализации и п.о.
- Реализовать как минимум один ООП интерфейс (необязательное)
- Использовать getters и setters как минимум в одном классе (необязательное)
- 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, ICloneable, IEnumerable, 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. Поток.**

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)

### **Описание особенностей реализации по темам**

Представленные названия классов могут меняться автором реализации на синонимы или другие названия, представляющих абстрактно аналогичное назначение.

Необязательные классы влияют на сложность разработки и вариативность реализации. Например, в теме «Космос» при реализации абстракции звездной системы (без галактики) класс «спутник» требуется для описания планет.

Рекомендации и требования представляют собой ограничения на минимальный размер реализуемого функционала и границы предметной области, но не ограничивают объем реализации (по желанию разработчика).

Интерфейс приложения зависит от темы и формата реализации:

- симуляции процесса;
- симулятора (пользователь взаимодействует с основными функциями приложения и влияет на симуляцию);
- «библиотека» (взаимодействие запросами, пример, создание библиотеки в теме «литература»);
- конструктор (например, «конфетная фабрика»)
- генератор (генерация структуры в соответствии с темой и ее использование как библиотеки или более, например, создание звездной системы в теме «Космос»);
- информативное приложение – задача (дается информация по назначению, возможна настройка, задание условий задачи и получения результата).

#### **1. Автомобилестроение:**

- Обязательные классы: Автомобиль, Салон/Гараж, Деталь (часть, part)
- Обеспечить: 5 автомобилей (около 20 деталей)

#### **2. Компьютерная техника:**

- Обязательные классы: Компьютер, Часть (part, устройство и т.д.)
- Обеспечить: минимум 20 комплектующих частей

- Рекомендуется: функции приложения по вычислению стоимости и/или эффективности сборки

### **3. Биология:**

- Обязательные классы: Фауна, Флора, Территория
- Обеспечить: 2-5 территорий, около 10 объектов из около 20-40 возможных
- Рекомендуется: связи объектов на территории (круговорот в природе)

### **4. Химия:**

- Обязательные классы: Вещество, Колба (Контейнер и др.), классы различных факторов взаимодействия веществ, Формула
- Необязательные классы: Молекула, другие классы факторов взаимодействия веществ
- Обеспечить: минимум 1 класс факторов
- Разработать: приложения получения одного вещества из других (с учетом разных факторов и побочных веществ)

### **5. Пчелиная ферма:**

- Обязательные классы: Мед, Пчела, Матка, Соты
- Необязательные классы: Составляющие меда (Вещество), класс симуляции
- Рекомендуется: реализация симуляции какого-либо процесса (например, пчеловодство, рой, транспортная система)
- Варианты функций приложения: качество меда (смотреть «Составляющие меда»), производство меда

### **6. Управление предприятием:**

- Обязательные классы: Работник, Менеджер, Продукты (Продукция, Товар)
- Определить: функциональное назначение и предприятие
- Рекомендуется: симуляция работы с документами или симулятор работы предприятия

### **7. Конфетная фабрика:**

- Обязательные классы: Начинка, Глазурь, Фантик, Коробка
- Необязательные классы:
- Обеспечить: интерфейс создания конфеты и ее упаковка
- Варианты функций приложения: стоимость и время изготовления

### **8. Пекарня (кулинария):**

- Обязательные классы: Хлебобулочное изделие, Витрина

- Необязательные классы: Торт, Пекарь
- Обеспечить: минимум 10 наименований изделий
- Определить: функциональное назначение

## **9. Танкостроение:**

- Обязательные классы: Часть (part, составляющая), Танк, Ангар
- Необязательные классы: Снаряд, Полигон
- Обеспечить: около 5 машин
- Определить: функциональное назначение (симуляция танка или библиотека)
- Варианты функций приложения: стоимость и время ремонта/изготовления танка

## **10. Авиастроение:**

- Обязательные классы: Ангар, Самолет, Вертолет
- Обеспечить: от 5 до 10 машин
- Рекомендуется: создание информативного приложения
- Варианты функций приложения: функции аэропорта

## **11. Музыка:**

- Обязательные классы: Композиция, Автор и/или Исполнитель, Альбом, Жанр, Библиотека
- Обеспечить: около 20 композиций
- Рекомендуется: реализация информативного (новостного) приложения

## **12. Живопись:**

- Обязательные классы: Картинка, Художник, Зал, Жанр
- Необязательные классы: Выставка
- Варианты функций приложения: экскурсии; получение размера групп для посещения (количество зрителей) и/или времени прохождения; стоимости билета
- Рекомендуется: не реализовывать симуляторы с экскурсиями или не использовать роли кроме экскурсовода и участника группы

## **13. Литература:**

- Обязательные классы: Автор, Жанр, Книга, Зал
- Обеспечить: около 40 наименований
- Рекомендуется: архитектура наследования на основе класса «Жанр»; реализовать библиотеку, но не симулятор библиотеки

#### **14. Базы данных:**

- Обязательные классы: СУБД, БД (Хранилище), запрос
- Необязательные классы: Хранилище
- Обеспечить: 5 записей в хранилище
- Рекомендуются: реализовать две СУБД, их отличия

#### **15. Free theme (Свободная тема):**

- Все нюансы и тема разработки согласуются с преподавателем по практике

#### **16. Языки программирования:**

- Обязательные классы: Язык программирования
- Необязательные классы: Команда, Код, Конструкция
- Рекомендуются: либо реализация приложения нового языка программирования (функционального и/или динамического), либо реализация перевода между двумя (тремя) языками выделенных базовых конструкций и команд

#### **17. Огород/садоводство:**

- Обязательные классы: Овощи, Фрукты (Ягоды)
- Необязательные классы: Урожай, Фермер
- Обеспечить: не менее 10 наименований
- Рекомендуются: мини-игры, например, симулятор фермера

#### **18. Зоопарк:**

- Обязательные классы: Вольер/Клетка, Корм, Животное
- Обеспечить: не менее 10 разных видов животных
- Определить: функциональное назначение

#### **19. Парк аттракционов:**

- Обязательные классы: Аттракцион, Парк (территория)
- Необязательные классы:
- Обеспечить: минимум 10 аттракционов, около 5 разных.
- Определить: функциональное назначение
- Варианты функций приложения: получение размера групп для посещения (количество зрителей) и/или времени прохождения; стоимости билета

#### **20. Космос:**

- Обязательные классы: Звездная система, Звезда, Планета



- Необязательные классы: Свободный объект, Спутник
- Определить: масштаб предметной области (галактика или звездная система) и функциональное назначение – библиотека или генератор звездной системы или галактики.
- Рекомендуется: реализовать генератор или «библиотеку» звездной системы (в библиотеке – нашей звездной системы)
- Условия: в системе всегда одно солнце; спутник как отдельный класс только для масштаба звездной системы; в свободных объектах (если используется такой класс), запрещены планеты.
- Варианты функций приложения: получение расстояния и другой информации

## **21. Продуктовый магазин:**

- Обязательные классы: Корзина, Касса, Продукт/Товар
- Необязательные классы: Продавец, Зал/Категория
- Обеспечить: не менее 20 товаров
- Варианты функций приложения: стоимость покупки/заказа

## **22. География:**

- Обязательные классы: Город, Страна, Класс использования (в зависимости от назначения «карты»)
- Необязательные классы: Соседи (Связи)
- Определить: функциональное назначение (например, транспортная система, месторождения или социальная)
- Обеспечить: наполнение в зависимости от назначения
- Варианты функций приложения: исходя из назначения, например, в транспортной – составления маршруты, стоимости и др.
- Рекомендуется: реализация «библиотеки», информативного приложения-карты или генератора мира/карты.

## **23. Салон красоты:**

- Обязательные классы: Прическа, Услуга
- Обеспечить: не менее 10 услуг
- Варианты функций приложения: стоимость и время услуги

## **24. Оркестр (концерт):**

- Обязательные классы: Музыкальный инструмент, Коллектив
- Необязательные классы: Композиция, Профессия
- Варианты функций приложения: разные коллективы, которые по-разному играют разные композиции; возможна мини-игра - симулятор.

## **25. Военная сфера:**

- Все нюансы и подтема разработки согласуются с преподавателем по практике:

## **26. Футбол:**

- Обязательные классы: Футболист, Матч, Команда
- Необязательные классы: Тренер, Игровое поле, Комментатор
- Рекомендуются: симуляция матча с интеграцией конструктора команд/матча

## **27. Столовая (буфет):**

- Обязательные классы: Продукт/Товар
- Необязательные классы: Кассир, Столик (Место)
- Обеспечить: не менее 10 наименований
- Варианты функций приложения: стоимость покупки/заказа