Вот список тем по объектно-ориентированному программированию в C# для новичков, расположенных от простых к более сложным:

1. Классы и объекты

2. Поля и свойства

3. Методы и конструкторы

4. Инкапсуляция

5. Наследование

6. Полиморфизм

7. Абстрактные классы

8. Интерфейсы

9. Перегрузка методов

10. Статические члены класса

11. Модификаторы доступа

12. Пространства имен

13. Обработка исключений в контексте ООП

14. Виртуальные методы

15. Запечатанные классы (sealed)

16. Частичные классы и методы (partial)

17. Обобщения (generics) в контексте ООП

18. Делегаты и события

19. Анонимные типы

20. Расширяющие методы (extension methods)

Для обучения начинающих студентов по каждой теме из списка прошу создать по три задачи легкие и три средней сложности, подробно описать каждую задачу, и написать подсказки для задач средней сложности.

Edit

Конечно, я подготовлю по три легкие и три задачи средней сложности для каждой темы из списка. Учитывая объем информации, я буду предоставлять материал по частям. Давайте начнем с первых двух тем.

1. Классы и объекты

Легкие задачи:

1.1. Создайте класс "Книга" с полями "название" и "автор". Создайте объект этого класса и выведите информацию о книге.

1.2. Разработайте класс "Прямоугольник" с полями "длина" и "ширина". Создайте метод для вычисления площади. Создайте объект и вычислите его площадь.

1.3. Создайте класс "Студент" с полями "имя" и "возраст". Создайте несколько объектов этого класса и выведите информацию о студентах.

Задачи средней сложности:

1.4. Разработайте класс "Банковский счет" с полями "номер счета", "баланс" и методами для внесения и снятия денег. Создайте несколько счетов и выполните операции с ними. Подсказка: Используйте методы для изменения баланса и проверяйте достаточность средств при снятии.

1.5. Создайте класс "Автомобиль" с полями "марка", "модель", "год выпуска" и "пробег". Добавьте метод для обновления пробега и метод для вывода полной информации об автомобиле. Подсказка: Убедитесь, что новый пробег не меньше текущего.

1.6. Разработайте класс "Библиотека", который может хранить список объектов класса "Книга". Добавьте методы для добавления книги, удаления книги и вывода всех книг. Подсказка: Используйте список (List<>) для хранения книг.

1. Поля и свойства

Легкие задачи:

2.1. Создайте класс "Круг" с приватным полем "радиус" и публичным свойством для доступа к этому полю.

2.2. Разработайте класс "Температура" с приватным полем для хранения значения в градусах Цельсия и публичными свойствами для получения значения в Цельсиях и Фаренгейтах.

2.3. Создайте класс "Сотрудник" с приватными полями "имя" и "зарплата", и публичными свойствами для доступа к ним. Свойство для зарплаты должно быть доступно только для чтения.

Задачи средней сложности:

2.4. Разработайте класс "Счетчик" с приватным полем для хранения значения и свойством, которое позволяет увеличивать и уменьшать значение счетчика, но не позволяет установить значение напрямую. Подсказка: Используйте только методы get и set в свойстве, без поля.

2.5. Создайте класс "Пароль" с приватным полем для хранения пароля и публичным свойством, которое при установке значения проверяет его на сложность (минимум 8 символов, наличие цифр и букв). Подсказка: Используйте регулярные выражения для проверки пароля.

2.6. Разработайте класс "Корзина" с приватным полем для хранения списка товаров и публичным свойством, которое возвращает общую стоимость товаров в корзине. Подсказка: Используйте LINQ для подсчета общей стоимости.

Это задачи для первых двух тем. Хотите ли вы продолжить с следующими темами или нужны дополнительные пояснения по этим задачам?

CopyRetry

**VV**

продолжай

Edit

Хорошо, продолжим со следующими двумя темами.

1. Методы и конструкторы

Легкие задачи:

3.1. Создайте класс "Калькулятор" с методами для сложения, вычитания, умножения и деления двух чисел.

3.2. Разработайте класс "Приветствие" с методом, который принимает имя и возвращает строку приветствия.

3.3. Создайте класс "Точка" с конструктором, принимающим координаты x и y, и методом для вычисления расстояния до начала координат.

Задачи средней сложности:

3.4. Разработайте класс "Массив" с методами для нахождения максимального, минимального элементов и их индексов. Реализуйте конструктор, позволяющий инициализировать массив случайными числами. Подсказка: Используйте класс Random для генерации случайных чисел.

3.5. Создайте класс "Строка" с методом для подсчета количества слов и методом для нахождения самого длинного слова. Реализуйте конструктор, принимающий строку. Подсказка: Используйте методы Split() и Max() для работы со строками.

3.6. Разработайте класс "Дата" с конструктором, принимающим день, месяц и год. Добавьте методы для проверки високосного года и определения дня недели для даты. Подсказка: Используйте класс DateTime для работы с датами.

1. Инкапсуляция

Легкие задачи:

4.1. Создайте класс "Счет" с приватным полем баланса и публичными методами для пополнения и снятия денег.

4.2. Разработайте класс "Человек" с приватными полями имени и возраста. Реализуйте публичные методы для получения и изменения этих данных.

4.3. Создайте класс "Термометр" с приватным полем температуры и публичными методами для получения значения в Цельсиях и Фаренгейтах.

Задачи средней сложности:

4.4. Разработайте класс "Сейф" с приватным полем для кода доступа и баланса. Реализуйте методы для изменения кода, пополнения и снятия денег, требующие ввода правильного кода. Подсказка: Используйте приватный метод для проверки кода.

4.5. Создайте класс "Электронная почта" с приватными полями для адреса и пароля. Реализуйте методы для изменения пароля (требуется старый пароль) и отправки письма (требуется пароль). Подсказка: Используйте регулярные выражения для проверки корректности email.

4.6. Разработайте класс "Игровой персонаж" с приватными полями здоровья, брони и урона. Реализуйте методы для атаки другого персонажа и получения урона, учитывая броню. Подсказка: Используйте закрытые методы для расчета итогового урона.

Это задачи для тем 3 и 4. Хотите ли вы продолжить с следующими темами или нужны дополнительные пояснения по этим задачам?

CopyRetry

**VV**

продолжай

Edit

Хорошо, продолжим с темами 5 и 6.

1. Наследование

Легкие задачи:

5.1. Создайте базовый класс "Животное" с методом "Издать звук". Создайте производные классы "Собака" и "Кошка", переопределяющие этот метод.

5.2. Разработайте базовый класс "Фигура" с методом "Площадь". Создайте производные классы "Круг" и "Прямоугольник", реализующие этот метод.

5.3. Создайте базовый класс "Транспортное средство" с свойствами "Скорость" и "Вместимость". Реализуйте производные классы "Автомобиль" и "Велосипед".

Задачи средней сложности:

5.4. Разработайте иерархию классов для банковских счетов: базовый класс "Счет" и производные "Сберегательный счет" и "Кредитный счет" с различной логикой начисления процентов. Подсказка: Используйте виртуальные методы в базовом классе для реализации различного поведения в производных классах.

5.5. Создайте иерархию классов для геометрических фигур с методами вычисления площади и периметра. Реализуйте не менее трех уровней наследования. Подсказка: Используйте абстрактные методы в базовом классе для обязательной реализации в производных классах.

5.6. Разработайте систему классов для игры: базовый класс "Персонаж" и производные классы для разных типов персонажей (воин, маг, лучник) с уникальными способностями. Подсказка: Используйте защищенные (protected) поля и методы для доступа к ним из производных классов.

1. Полиморфизм

Легкие задачи:

6.1. Создайте базовый класс "Форма" с виртуальным методом "Нарисовать". Реализуйте производные классы "Круг", "Квадрат" и "Треугольник" с переопределением этого метода.

6.2. Разработайте базовый класс "Работник" с виртуальным методом "Расчет зарплаты". Создайте производные классы "Менеджер" и "Рабочий" с различной логикой расчета.

6.3. Создайте интерфейс "IПлавающий" с методом "Плыть". Реализуйте этот интерфейс в классах "Лодка", "Утка" и "Рыба".

Задачи средней сложности:

6.4. Разработайте систему классов для игры в шахматы. Создайте базовый класс "Фигура" и производные классы для различных фигур. Реализуйте полиморфный метод "Ход". Подсказка: Используйте абстрактные методы и свойства в базовом классе для обеспечения общего интерфейса всех фигур.

6.5. Создайте систему классов для обработки различных типов файлов (текстовые, изображения, аудио). Реализуйте общий интерфейс для открытия, чтения и закрытия файлов. Подсказка: Используйте интерфейс IDisposable для корректного освобождения ресурсов.

6.6. Разработайте иерархию классов для различных типов транспортных средств с методами "Начать движение", "Остановиться" и "Заправиться". Реализуйте эти методы с учетом особенностей каждого типа транспорта. Подсказка: Используйте виртуальные методы и переопределение для реализации различного поведения.

Это задачи для тем 5 и 6. Хотите ли вы продолжить с следующими темами или нужны дополнительные пояснения по этим задачам?

CopyRetry

**VV**

все хорошо, продолжай

Edit

Отлично, продолжим с темами 7 и 8.

1. Абстрактные классы

Легкие задачи:

7.1. Создайте абстрактный класс "Фигура" с абстрактным методом "Площадь". Реализуйте классы "Круг" и "Прямоугольник", наследующие этот класс.

7.2. Разработайте абстрактный класс "Животное" с абстрактными методами "Есть" и "Спать". Создайте классы "Собака" и "Кошка", реализующие эти методы.

7.3. Создайте абстрактный класс "Сотрудник" с абстрактным методом "Расчет зарплаты". Реализуйте классы "Менеджер" и "Разработчик".

Задачи средней сложности:

7.4. Разработайте абстрактный класс "Транспортное средство" с абстрактными методами "Запустить двигатель", "Остановить двигатель" и свойством "Скорость". Реализуйте классы "Автомобиль" и "Мотоцикл". Подсказка: Добавьте в абстрактный класс некоторую общую функциональность, например, метод "Описание".

7.5. Создайте абстрактный класс "Оружие" с абстрактными методами "Атака" и "Перезарядка". Реализуйте несколько классов различных типов оружия. Подсказка: Используйте защищенные поля в абстрактном классе для хранения общих характеристик оружия.

7.6. Разработайте абстрактный класс "Банковский счет" с абстрактными методами "Внести деньги" и "Снять деньги". Реализуйте классы "Сберегательный счет" и "Кредитный счет" с различной логикой работы. Подсказка: Добавьте в абстрактный класс виртуальный метод для расчета процентов, который можно переопределить в дочерних классах.

1. Интерфейсы

Легкие задачи: 8.1. Создайте интерфейс "IФигура" с методами "Площадь" и "Периметр". Реализуйте этот интерфейс в классах "Круг" и "Прямоугольник". 8.2. Разработайте интерфейс "IСравнимый" с методом "СравнитьС". Реализуйте этот интерфейс в классе "Число". 8.3. Создайте интерфейс "IПлавающий" с методом "Плыть" и интерфейс "IЛетающий" с методом "Лететь". Реализуйте оба интерфейса в классе "Утка".

Задачи средней сложности: 8.4. Разработайте систему интерфейсов для игры: IДвижимый (с методами Двигаться и Остановиться), IАтакующий (с методом Атаковать) и IУязвимый (с методом ПолучитьУрон). Реализуйте эти интерфейсы в классах различных игровых персонажей. Подсказка: Используйте множественную реализацию интерфейсов для создания разнообразных типов персонажей.

8.5. Создайте интерфейс IХранилище с методами Добавить, Удалить и Найти. Реализуйте этот интерфейс в классах СписокТоваров и КаталогКниг, используя разные структуры данных для хранения элементов. Подсказка: Используйте обобщения (generics) для создания универсального интерфейса хранилища.

8.6. Разработайте систему классов для обработки различных типов данных (текст, изображение, аудио) с общим интерфейсом IОбрабатываемый, содержащим методы Открыть, Обработать и Сохранить. Подсказка: Реализуйте также интерфейс IDisposable для корректного освобождения ресурсов.

Это задачи для тем 7 и 8. Хотите ли вы продолжить с следующими темами или нужны дополнительные пояснения по этим задачам?

CopyRetry

**VV**

продолжим

Edit

Хорошо, продолжим с темами 9 и 10.

1. Перегрузка методов

Легкие задачи: 9.1. Создайте класс "Калькулятор" с перегруженными методами "Сложить" для сложения двух и трех чисел. 9.2. Разработайте класс "Принтер" с перегруженными методами "Печать" для вывода строки, числа и массива. 9.3. Создайте класс "Точка" с перегруженными конструкторами для создания точки с координатами (0,0), (x,0) и (x,y).

Задачи средней сложности: 9.4. Разработайте класс "Массив" с перегруженными методами "Поиск" для поиска элемента по значению, по индексу и по условию (предикату). Подсказка: Используйте делегаты для реализации поиска по условию.

9.5. Создайте класс "Форматтер" с перегруженными методами "Форматировать" для различных типов данных (строки, числа, даты) с возможностью указания дополнительных параметров форматирования. Подсказка: Используйте необязательные параметры для гибкости форматирования.

9.6. Разработайте класс "Конвертер" с перегруженными методами для конвертации различных единиц измерения (длины, веса, температуры) с возможностью указания точности округления. Подсказка: Используйте перечисления (enums) для определения единиц измерения.

1. Статические члены класса

Легкие задачи: 10.1. Создайте класс "Счетчик" со статическим полем для хранения количества созданных объектов класса. 10.2. Разработайте класс "Конвертер" со статическими методами для конвертации температуры между шкалами Цельсия и Фаренгейта. 10.3. Создайте класс "Математика" со статическими методами для вычисления факториала и степени числа.

Задачи средней сложности: 10.4. Разработайте класс "Логгер" со статическим методом для записи сообщений в файл. Реализуйте возможность настройки уровня логирования (информация, предупреждение, ошибка) через статическое свойство. Подсказка: Используйте блокировку (lock) для обеспечения потокобезопасности при записи в файл.

10.5. Создайте класс "Кэш" со статическими методами для сохранения и извлечения объектов. Реализуйте механизм автоматической очистки кэша по истечении времени жизни объектов. Подсказка: Используйте словарь (Dictionary) для хранения объектов и их времени жизни.

10.6. Разработайте класс "Конфигурация" для хранения настроек приложения. Реализуйте статические методы для загрузки настроек из файла и сохранения их обратно в файл. Подсказка: Используйте сериализацию для сохранения и загрузки настроек в/из файла.

Это задачи для тем 9 и 10. Хотите ли вы продолжить с следующими темами или нужны дополнительные пояснения по этим задачам?

CopyRetry

**VV**

продолжай

Edit

Хорошо, продолжим с темами 11 и 12.

1. Модификаторы доступа

Легкие задачи: 11.1. Создайте класс "Сотрудник" с приватными полями для имени и зарплаты, и публичными методами для их получения и изменения. 11.2. Разработайте класс "Автомобиль" с защищенным (protected) полем для модели и публичными методами для работы с ним. 11.3. Создайте класс "Счет" с внутренним (internal) методом для обновления баланса и публичными методами для внесения и снятия денег.

Задачи средней сложности: 11.4. Разработайте базовый класс "Фигура" с защищенными полями и методами. Создайте производные классы, которые используют эти защищенные члены. Подсказка: Используйте виртуальные методы в базовом классе для переопределения в производных классах.

11.5. Создайте класс "Безопасное хранилище" с приватными полями для хранения чувствительных данных и публичными методами для безопасного доступа к ним. Подсказка: Реализуйте механизм шифрования для защиты данных.

11.6. Разработайте систему классов для игры, где некоторые классы и методы должны быть доступны только внутри сборки (internal). Подсказка: Используйте internal для классов, которые не должны быть доступны извне библиотеки игры.

1. Пространства имен

Легкие задачи: 12.1. Создайте пространство имен "Геометрия" и поместите в него классы "Круг" и "Прямоугольник". 12.2. Разработайте пространство имен "Банк" с классами "Счет" и "Клиент". 12.3. Создайте два пространства имен "Животные" и "Растения" и поместите в них соответствующие классы.

Задачи средней сложности: 12.4. Разработайте систему пространств имен для игрового движка: "Графика", "Физика", "Звук" и "Ввод". Создайте классы в каждом пространстве имен. Подсказка: Используйте вложенные пространства имен для более детальной организации.

12.5. Создайте библиотеку классов с пространством имен "МояБиблиотека". Реализуйте в ней несколько полезных утилитных классов. Затем создайте консольное приложение, использующее эту библиотеку. Подсказка: Используйте директиву using для импорта пространства имен в консольное приложение.

12.6. Разработайте систему пространств имен для приложения электронной коммерции: "Товары", "Заказы", "Доставка" и "Оплата". Реализуйте классы в каждом пространстве имен и продемонстрируйте их взаимодействие. Подсказка: Используйте полные имена классов (с указанием пространства имен) для разрешения конфликтов имен.

Это задачи для тем 11 и 12. Хотите ли вы продолжить с следующими темами или нужны дополнительные пояснения по этим задачам?

CopyRetry

**VV**

продолжай

Edit

Хорошо, давайте продолжим с темами 13 и 14.

1. Обработка исключений в контексте ООП

Легкие задачи: 13.1. Создайте класс "Калькулятор" с методом деления. Реализуйте обработку исключения DivideByZeroException. 13.2. Разработайте класс "Массив" с методом получения элемента по индексу. Обработайте исключение IndexOutOfRangeException. 13.3. Создайте класс "Файловый менеджер" с методом чтения файла. Обработайте исключение FileNotFoundException.

Задачи средней сложности: 13.4. Разработайте класс "Банковский счет" с методами снятия и внесения денег. Создайте пользовательское исключение "НедостаточноСредствИсключение" и выбрасывайте его при попытке снять больше денег, чем есть на счете. Подсказка: Унаследуйте ваше исключение от класса Exception и добавьте в него дополнительную информацию о транзакции.

13.5. Создайте иерархию исключений для игровой системы: "ИгровоеИсключение" как базовый класс, и производные "ПерсонажУмерИсключение", "НедостаточноРесурсовИсключение". Реализуйте классы игровой логики, которые выбрасывают и обрабатывают эти исключения. Подсказка: Используйте свойства исключений для передачи дополнительной информации об игровой ситуации.

13.6. Разработайте класс "Парсер конфигурации", который читает настройки из файла. Реализуйте обработку различных исключений (FileNotFoundException, FormatException, OverflowException) и создайте собственное исключение "НекорректнаяКонфигурацияИсключение". Подсказка: Используйте блок finally для закрытия файла, независимо от возникновения исключений.

1. Виртуальные методы

Легкие задачи: 14.1. Создайте базовый класс "Фигура" с виртуальным методом "Площадь". Реализуйте производные классы "Круг" и "Прямоугольник" с переопределением этого метода. 14.2. Разработайте базовый класс "Животное" с виртуальным методом "Голос". Создайте производные классы "Собака" и "Кошка" с переопределением этого метода. 14.3. Создайте базовый класс "Сотрудник" с виртуальным методом "РасчетЗарплаты". Реализуйте производные классы "Менеджер" и "Разработчик" с различной логикой расчета зарплаты.

Задачи средней сложности: 14.4. Разработайте иерархию классов для игровых персонажей с базовым классом "Персонаж" и виртуальными методами "Атака" и "ПолучениеУрона". Реализуйте несколько типов персонажей с различным поведением. Подсказка: Используйте защищенные виртуальные методы для реализации базовой логики, которую можно расширить в производных классах.

14.5. Создайте базовый класс "Транспортное средство" с виртуальными методами "Старт", "Движение" и "Остановка". Реализуйте производные классы для различных типов транспорта (например, "Автомобиль", "Самолет", "Корабль") с соответствующим поведением. Подсказка: Используйте ключевое слово base для вызова реализации базового класса в переопределенных методах.

14.6. Разработайте систему классов для обработки различных типов файлов. Создайте базовый класс "ОбработчикФайла" с виртуальными методами "Открыть", "Обработать" и "Закрыть". Реализуйте производные классы для обработки текстовых файлов, изображений и аудиофайлов. Подсказка: Используйте абстрактные методы для операций, которые должны быть обязательно реализованы в производных классах, и виртуальные методы для операций с базовой реализацией.

Это задачи для тем 13 и 14. Хотите ли вы продолжить с следующими темами или нужны дополнительные пояснения по этим задачам?

CopyRetry

**VV**

продолжай

Edit

Отлично, продолжим с темами 15 и 16.

1. Запечатанные классы (sealed)

Легкие задачи: 15.1. Создайте запечатанный класс "Настройки" для хранения конфигурации приложения. 15.2. Разработайте иерархию классов "Фигура" -> "Четырехугольник" -> "Прямоугольник", где "Прямоугольник" будет запечатанным классом. 15.3. Создайте запечатанный класс "Утилиты" со статическими методами для общих операций.

Задачи средней сложности: 15.4. Разработайте систему классов для игры в шахматы. Создайте базовый класс "Фигура" и производные классы для каждого типа фигур. Сделайте классы конкретных фигур (например, "Ферзь", "Ладья") запечатанными. Подсказка: Используйте виртуальные методы в базовом классе для определения поведения фигур.

15.5. Создайте класс "Шифрование" с виртуальными методами "Зашифровать" и "Расшифровать". Реализуйте несколько производных классов для разных алгоритмов шифрования, сделав последний в иерархии класс запечатанным. Подсказка: Продумайте, почему важно запретить дальнейшее наследование от классов с реализацией конкретных алгоритмов шифрования.

15.6. Разработайте систему классов для представления различных типов банковских счетов. Сделайте классы конкретных типов счетов (например, "СберегательныйСчет", "КредитныйСчет") запечатанными. Подсказка: Используйте абстрактный базовый класс для общей функциональности всех типов счетов.

1. Частичные классы и методы (partial)

Легкие задачи: 16.1. Создайте частичный класс "Клиент" с методами в разных файлах: один для личной информации, другой для финансовой. 16.2. Разработайте частичный класс "Автомобиль" с методами в разных файлах: один для характеристик, другой для действий. 16.3. Создайте частичный класс "Калькулятор" с базовыми операциями в одном файле и дополнительными в другом.

Задачи средней сложности: 16.4. Разработайте частичный класс "Форма" для пользовательского интерфейса. В одной части класса определите свойства и поля, в другой - методы для обработки событий пользовательского ввода. Подсказка: Используйте частичные методы для объявления в одной части класса и реализации в другой.

16.5. Создайте частичный класс "БазаДанных" с методами для работы с различными таблицами, распределенными по разным файлам. Подсказка: Используйте частичные методы для возможности расширения функциональности в будущем.

16.6. Разработайте систему классов для игрового движка. Создайте частичный класс "ИгровойДвижок" с различными компонентами (графика, физика, звук, ввод), распределенными по разным файлам. Подсказка: Используйте частичные методы для определения точек расширения в каждом компоненте.

Это задачи для тем 15 и 16. Хотите ли вы продолжить с следующими темами или нужны дополнительные пояснения по этим задачам?

CopyRetry

**VV**

продолжай

Edit

Отлично, продолжим с темами 17 и 18.

1. Обобщения (generics) в контексте ООП

Легкие задачи: 17.1. Создайте обобщенный класс "Пара<T>" для хранения пары значений одного типа. 17.2. Разработайте обобщенный метод "Обмен<T>", который меняет местами значения двух переменных. 17.3. Создайте обобщенный класс "Стек<T>" с методами Push, Pop и Peek.

Задачи средней сложности: 17.4. Разработайте обобщенный класс "Хранилище<T>" для хранения объектов. Реализуйте методы добавления, удаления и поиска элементов. Добавьте ограничение, чтобы T реализовывал интерфейс IComparable<T>. Подсказка: Используйте ограничение where T : IComparable<T> для обеспечения возможности сравнения объектов.

17.5. Создайте обобщенный класс "Калькулятор<T>" с методами для базовых арифметических операций. Реализуйте этот класс для типов int, double и decimal. Подсказка: Используйте ограничения типов для обеспечения возможности выполнения арифметических операций.

17.6. Разработайте обобщенный класс "ДвоичноеДерево<T>" для представления бинарного дерева поиска. Реализуйте методы вставки, удаления и поиска элементов. Подсказка: Используйте рекурсию для реализации операций с деревом.

1. Делегаты и события

Легкие задачи: 18.1. Создайте делегат для представления простой математической операции и используйте его в методе "ВыполнитьОперацию". 18.2. Разработайте класс "Таймер" с событием, которое срабатывает каждую секунду. 18.3. Создайте класс "Логгер" с событием, которое вызывается при записи нового сообщения в лог.

Задачи средней сложности: 18.4. Разработайте систему подписки на новости. Создайте класс "НовостнойАгрегатор" с событием "НоваяНовость". Реализуйте несколько классов подписчиков, которые будут реагировать на это событие. Подсказка: Используйте делегат EventHandler<T> для определения события.

18.5. Создайте класс "БанковскийСчет" с событиями "СредстваДобавлены" и "СредстваСняты". Реализуйте класс "АудитТранзакций", который будет подписываться на эти события и вести журнал всех операций. Подсказка: Создайте пользовательский класс для аргументов события, содержащий информацию о транзакции.

18.6. Разработайте простую систему событий для игрового движка. Создайте классы для различных игровых объектов с событиями (например, "СтолкновениеПроизошло", "ЗдоровьеИзменилось"). Реализуйте систему обработки этих событий. Подсказка: Используйте многоадресные делегаты для подписки нескольких обработчиков на одно событие.

Это задачи для тем 17 и 18. Хотите ли вы продолжить с последними двумя темами или нужны дополнительные пояснения по этим задачам?

CopyRetry

**VV**

продолжай

Edit

Отлично, давайте завершим с последними двумя темами - 19 и 20.

1. Анонимные типы

Легкие задачи: 19.1. Создайте метод, возвращающий анонимный тип с информацией о текущей дате (день, месяц, год). 19.2. Напишите запрос LINQ, использующий анонимный тип для выборки имени и возраста из списка объектов класса "Человек". 19.3. Создайте массив анонимных объектов, представляющих точки с координатами x и y.

Задачи средней сложности: 19.4. Разработайте метод, который принимает список объектов "Товар" и возвращает список анонимных объектов с информацией о названии товара, цене и категории. Отсортируйте результат по цене. Подсказка: Используйте LINQ и метод OrderBy для сортировки.

19.5. Создайте метод, который группирует список сотрудников по отделам, используя анонимные типы для представления результатов группировки. Подсказка: Используйте LINQ и метод GroupBy для группировки данных.

19.6. Разработайте систему анализа продаж, которая использует анонимные типы для создания сводных отчетов. Создайте методы для расчета общей суммы продаж по категориям, среднего чека и топ-5 продаваемых товаров. Подсказка: Комбинируйте различные LINQ-запросы и анонимные типы для создания сложных отчетов.

1. Расширяющие методы (extension methods)

Легкие задачи: 20.1. Создайте расширяющий метод для строки, который подсчитывает количество слов. 20.2. Напишите расширяющий метод для целочисленного массива, который находит среднее значение элементов. 20.3. Разработайте расширяющий метод для DateTime, который возвращает название дня недели на русском языке.

Задачи средней сложности: 20.4. Создайте набор расширяющих методов для работы с коллекциями (например, List<T>). Реализуйте методы для нахождения второго по величине элемента, удаления дубликатов и разбиения коллекции на части заданного размера. Подсказка: Используйте LINQ в сочетании с расширяющими методами для эффективной обработки коллекций.

20.5. Разработайте расширяющие методы для класса string, которые реализуют простые операции шифрования и дешифрования (например, шифр Цезаря). Подсказка: Используйте методы класса string и char для работы с отдельными символами.

20.6. Создайте набор расширяющих методов для работы с числами. Реализуйте методы для проверки на простоту числа, нахождения наибольшего общего делителя двух чисел и генерации последовательности Фибоначчи до заданного числа. Подсказка: Для метода генерации последовательности Фибоначчи используйте yield return для создания итератора.

Это завершает наш список задач по всем 20 темам объектно-ориентированного программирования в C#. Эти задачи охватывают широкий спектр концепций ООП и должны помочь студентам углубить свое понимание и практические навыки в C#. Есть ли у вас какие-либо вопросы по этим задачам или нужны ли дополнительные пояснения?