

## Делегаты и события

1. Создайте программу, которая принимает список чисел и сортирует его в порядке возрастания или убывания. Позвольте пользователю выбрать с помощью делегата `Comparison<T>` порядок сортировки. Создайте методы, которые будут вызываться в качестве аргумента делегата для сравнения чисел и передайте их в метод сортировки.
2. Создайте программу, которая принимает список строк и фильтрует его на основе определенного критерия. Разрешите пользователю выбрать с помощью делегата `Predicate<T>` условие фильтрации. Создайте методы, которые будут вызываться в качестве аргумента делегата для проверки строк и передайте их в метод фильтрации.
3. Создайте класс, представляющий банковский счет. Добавьте свойство для хранения баланса и методы для снятия и пополнения денег. Затем создайте свой собственный делегат `AccountTransactionDelegate`, который принимает в качестве параметров сумму транзакции и тип операции (пополнение или снятие). Используйте созданный делегат для регистрации и вызова обработчиков транзакций.
4. Создайте программу, которая принимает список слов и преобразует каждое слово с помощью заданного делегата `Func<T, TResult>`. Позвольте пользователю выбрать делегат для преобразования слов. Создайте методы, которые будут вызываться в качестве аргумента делегата для преобразования слов и передайте их в метод обработки списка слов.
5. Создайте класс `Timer`, который будет генерировать событие `'Tick'` каждый раз, когда истекает определенный интервал времени. Реализуйте событие `'Tick'`, которое будет генерироваться при каждом тике таймера. Затем создайте методы-обработчики, которые будут вызываться при возникновении события `'Tick'` и реагировать на это событие.
6. Создайте программу, которая имитирует работу магазина. Создайте класс `'Product'`, который представляет товар с определенным названием и ценой. Затем создайте класс `'PurchaseEventArgs'`, который наследуется от `'EventArgs'` и содержит информацию о покупке (например, название товара и сумма покупки). Добавьте событие `'PurchaseCompleted'`, которое будет генерироваться при каждой покупке, и передавать объект `'PurchaseEventArgs'` с информацией о покупке.