

МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт (факультет)	Институт информационных технологий
----------------------	------------------------------------

Кафедра	Кафедра математического и программного обеспечения ЭВМ
---------	--

Задание на лабораторную работу №4

Дисциплина: С#-программирование

Темы:	Делегаты и события
--------------	--------------------

Среда разработки: Microsoft Visual Studio 2022

Язык программирования: С# 9.0

Тип проекта: Консольное приложение

ЗАДАНИЕ

ТРЕБОВАНИЯ К РАЗРАБОТКЕ

1. Придерживайтесь принципа DRY (Don't repeat yourself).
2. Используйте комментарии и xml-комментарии.
3. Обработка и генерация исключительных ситуаций на ваше усмотрение.

ЗАДАНИЯ

1. Измените поведение смены идентификатора витрины класса-контейнера:
 - Создайте делегат, информирующий о смене идентификатора.
 - Крайне желательно использовать готовые делегаты.
 - Для информирования о смене идентификатора, передается ссылка на саму витрину.
 - Используя методы расширения, добавьте метод в обобщенный товар, соответствующий сигнатуре делегата, описанного выше;
 - Метод расширения можно перегрузить для случая, когда меняется позиция товара на витрине;
 - Методы расширения, это именно методы расширения, а не просто добавленные в класс новые функции;
 - Обобщенный товар, это не новый класс, который требуется разработать, вспомните об обобщенном классе-контейнере и функций переинициализации штрихкода/QR-кода (в каких двух местах они раньше вызывались).
 - Обеспечьте правильный вызов, подписку и отписку делегата смена идентификатора витрины;
 - Для информации, товару нужно знать на какой идентификатор сменила витрина и в какой позиции сейчас товар.
2. Создайте запакованный класс производный от **EventArgs**, хранящий информацию по-старому и новому идентификатору:
 - Добавьте в интерфейс товара событие смены его идентификатора, и реализуйте его в абстрактном классе товара.
 - Делегат должен быть типа: **EventHandler<T>**, где **T** – ваш производный класс от **EventArgs**;

C#-программирование

- Обеспечьте правильную подписку и отписку от события в классе-контейнере, позволяющую правильно рассчитать новый штрихкод/QR-код товара на складе при смене его идентификатора.
3. Свяжите функции сортировки товара через новую функцию, с использованием делегатов.
 - Определите общую часть кода в методах сортировки;
 - Вынесите общую часть кода в новый метод;
 - Отличающуюся логику в методе замените делегатом, нужной сигнатуры.
 4. Свяжите функции поиска позиции товара через новую функцию, с использованием делегатов.
 - Определите общую часть кода в функциях поиска;
 - Вынесите общую часть кода в новую функцию;
 - Отличающуюся логику в функции замените делегатом, нужной сигнатуры.
 5. Протестируйте работу программы.

ТЕСТИРОВАНИЕ

Псевдокод:

```
static void TestLab4()
{
    Console.Clear();
    var lab4Data = new List<Товароподобный>
    {
        new ОбычныйТовар(3000, "ВОЙНА И МИРЬ III", "Л.Н. Толстой", 1867, 300000),
        new ОбычныйТовар(1000, "ВОЙНА И МИРЬ I", "Л.Н. Толстой", 1863, 1000000),
        new ОбычныйТовар(2000, "ВОЙНА И МИРЬ II", "Л.Н. Толстой", 1865, 200000),
        new КонечныйТовар(5555, "Хранители", "С. Маккоуд", 2008, 2071),
        new КонечныйТовар(6666, "Понимание комикса", "А. Шпигельман", 1990, 860)
    };

    ВитринаТовароподобных a = (7, 1); // витрина на 7 позиций с 1м идентификатором
    foreach (var товар in lab4Data)
    {
        a.Push(товар);
    }
    a.OrderByName();

    var sample1 = new КонечныйТовар(7777, "Ходячие мертвецы", "Р. Кирман", 2003, 2257);
    var sample2 = new ОбычныйТовар(4000, "ВОЙНА И МИРЬ IV", "Л.Н. Толстой", 1869, 400000);
    a[5] = sample1;
    a[6] = sample2;

    // смена ID
    a.Id = 2;
    sample1.Id++;
    sample2.Id++;

    // Каждый товар должен правильно отображать штрих код:
    // - ОбычныйТовар на витрине отображает штрихкод с позицией и ID витрины
    // - КонечныйТовар, отображает только ID товара, обрамленный *
    Console.WriteLine(a);
}
```

ВОПРОСЫ

1. Чем делегат отличается от события?
2. Можно ли делегаты объявлять вне класса?
3. Что такое Action, Func, Predicate?
4. Можно ли связать Event с Delegate?
5. Что такое Add и Remove?
6. Можно ли выполнять арифметические операции с делегатами?
7. Является ли любой Delegate – MulticastDelegate-ом?