

№ 5 Делегаты, события и рефлексия

Задание

1. Возьмите за основу лабораторную №3. Используя делегаты (множественные) и события промоделируйте следующие ситуации. Можете добавить новые типы (классы), если существующих не достаточно.

Далее приведен перечень заданий.

Вариант	Задание
1, 2, 3, 13, 14, 15	Создать класс <i>Пользователь</i> с событиями <i>Переместить</i> (можно задать смещение) и <i>Сжать</i> (коэффициент сжатия). В main создать некоторое количество объектов <i>разного типа</i> . Часть объектов подписать на одно событие, часть на два (часть можете не подписывать). Проверить результаты изменения объектов после наступления событий.
4	Создать класс <i>Директор</i> с событиями <i>Повысить</i> (можно задать величину) и <i>Штраф</i> . В main создать некоторое количество объектов <i>токарей и студентов-заочников (или т.п.)</i> . Часть объектов подписать на одно событие, часть на два (часть можете не подписывать). Проверить результаты изменения зарплаты после наступления событий.
5, 7	Создать класс <i>Игра</i> с событиями <i>Атака</i> и <i>Лечить</i> . В main создать некоторое количество игровых объектов. Подпишите объекты на события произвольным образом. Реакция на события у разных объектов может быть разной (без изменения, увеличивается/уменьшается уровень жизни). Проверить результаты изменения состояния игровых объектов после наступления событий, возможно не однократном.
6, 8, 9	Создать класс <i>Boss</i> с событиями <i>Upgrade</i> и <i>Turn-on</i> (под определенным напряжением, учтите что техника или человек могут сломаться). В main создать некоторое количество объектов (техники или полу-техники). Подпишите объекты на события произвольным образом. Реакция на события у разных объектов будет разная. Проверить результаты изменения состояния объектов после наступления событий, возможно не однократном.
10	Создать класс <i>Программист</i> с событиями <i>Удалить</i> и <i>Мутировать</i> . В main создать некоторое количество объектов (списков). Подпишите объекты на события произвольным образом. Реакция на события может быть следующей: удаление

	первого/последнего элемента списка, случайное перемещение строк и т.п. Проверить результаты изменения состояния объектов после наступления событий, возможно не однократном
11	Создать класс <i>Пользователь</i> с событиями <i>upgrade</i> и <i>work</i> . В <i>main</i> создать некоторое количество объектов (ПО). Подпишите объекты на события произвольным образом. Реакция на события может быть следующая: обновление версии, вывод сообщений и т.п. Проверить результаты изменения объектов после наступления событий.
12	Создать класс <i>Программист</i> с событиями <i>Переименовать</i> и <i>Новое свойство</i> . В <i>main</i> создать некоторое количество объектов (языков программирования). Подпишите объекты на события произвольным образом. Реакция на события может быть следующая: изменение имени/версии, добавление новых операций, технологий или понятий. Проверить результаты изменения состояния объектов после наступления событий, возможно не однократном

2. Для изучения .NET Reflection API допишите класс *Рефлектор*, который будет содержать методы выполняющие следующие действия:

- выводит всё содержимое класса в текстовый файл (принимает в качестве параметра имя класса);
- извлекает все общедоступные публичные методы класса (принимает в качестве параметра имя класса);
- получает информацию о полях и свойствах класса;
- получает все реализованные классом интерфейсы;
- выводит по имени класса имена методов, которые содержат заданный (пользователем) тип параметра (имя класса передается в качестве аргумента);
- вызывает некоторый метод класса, при этом значения для его параметров необходимо прочитать из текстового файла (имя класса и имя метода передаются в качестве аргументов).

Продемонстрируйте работу Рефлектора на разных типах.

Вопросы

- Что такое делегат? Как определить делегат?
- Назначение делегатов.
- Какие есть способы присваивания делегату адреса метода?
- Поясните назначение метода *Invoke*.
- Что такое групповая адресация делегата?
- Как создать событие?
- Как события связаны с делегатами? Опишите и поясните схему взаимодействия.

8. Что такое лямбда-выражения? Приведите пример лямбда-выражения с несколькими параметрами.
9. Что такое ковариантность и контравариантность делегатов? Что это дает?
10. Поясните разницу между встроенными делегатами Action и Func.
11. Что такое анонимный метод и где его можно использовать?
12. Какие классы существуют для работы с файлами, текстовыми и бинарными?
13. Какие классы существуют для работы с файловой системой?
14. Что такое Рефлексия?
15. Как можно использовать класс System.Type?
16. Приведите три способа получения типа.