

**«Разработка ОО программы моделирования заданной
предметной области»**

Содержание

1. Задание	3
2. Описание предметной области	3
3. Диаграмма классов	5
4. Описание полей и методов классов	6
5. Описание характеристик танков	15
6. Описание пользовательского интерфейса	16
7. Тестирование программы	18
8. Выводы	23
Приложение А. Код иерархии классов.	24
Приложение Б. Код основной программы	32

1. Задание

Разработать ОО программу моделирования заданной предметной области на языке программирования C#. Реализовать для иерархии механизм интерфейсов, один из классов должен реализовывать как минимум два интерфейса. Использовать для проверки всех методов данного класса многоадресный делегат. Реализовать обработку ошибок (переопределив с помощью наследования одно из событий: `StackOverflowException`, `ArrayTypeMismatchException`, `DivideByZeroException`, `IndexOutOfRangeException`, `InvalidCastException`, `OutOfMemoryException`, `OverflowException`). Показать пример использования полиморфизма.

2. Описание предметной области

Предметная область: Танковое сражение

В мире танковых сражений пользователю предоставляется возможность исследовать и использовать танки различных уровней, от 1 до 4. Каждый танк имеет свои характеристики, включая уровень урона, а также специальные способности, которые зависят от класса техники.

Одна из основных задач пользователя – исследовать танки в условиях ограниченных ресурсов. Перед началом сражения пользователю доступно 280 000 опыта (Exp) и 113 000 рублей (RUB), которые потребуется для исследования и покупки танков. Пользователь может выбирать танки для исследования, начиная с уровня 1 и двигаясь к уровню 4. Чем выше уровень танка, тем больше урона он способен нанести и тем более мощные способности у него имеются. Исследование танков возможно строго последовательно согласно дереву исследования, которое представлено на рис. 1. Стоимость исследования танков по уровням приведена в табл.1.

Таблица 1. Стоимость исследования танков

Ресурс \ Уровень	1	2	3	4
Exp	0	10 000	30 000	80 000
RUB	0	1 000	10 000	40 000

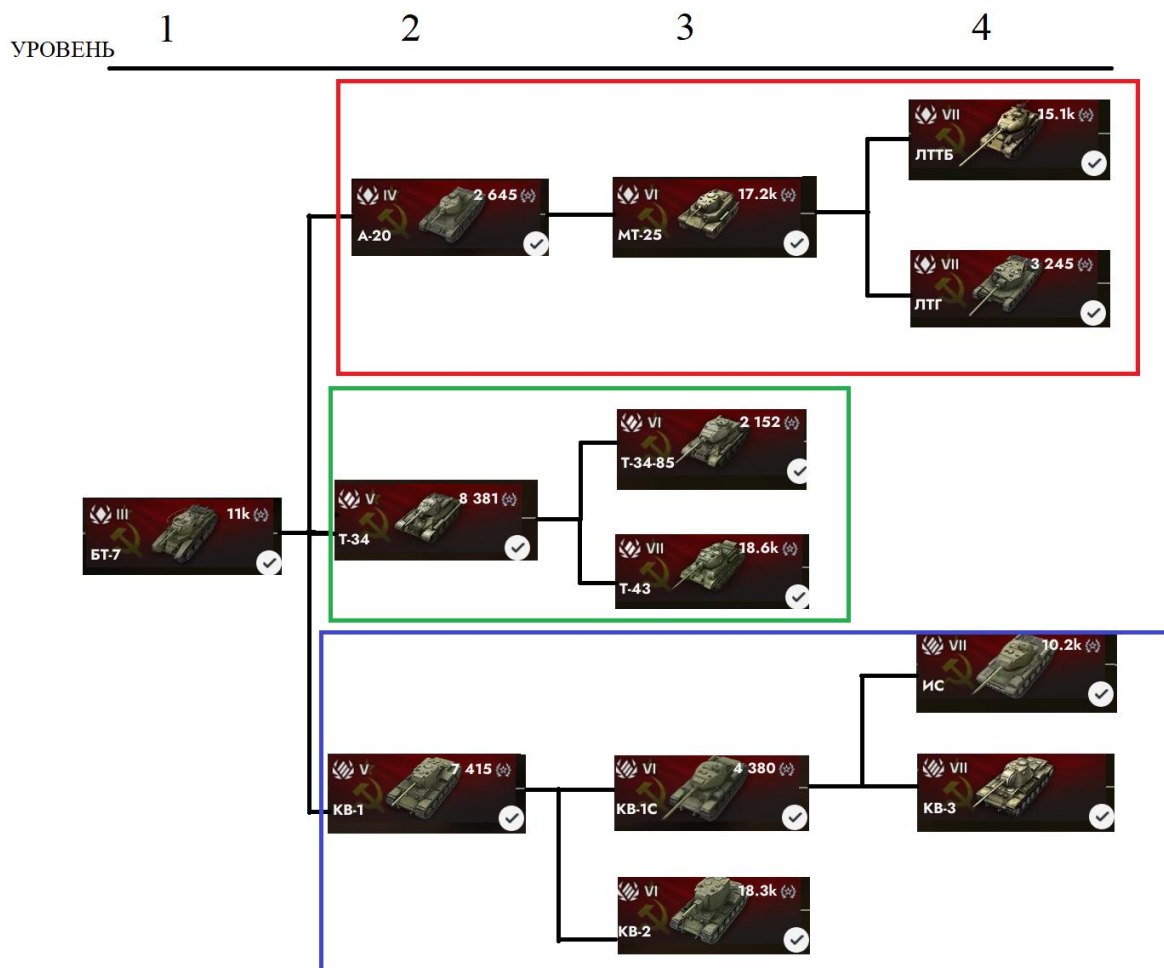


Рис. 1. Дерево исследования танков

Красным цветом выделены легкие танки, зеленым выделены средние танки, а синим – тяжёлые. Танк первого уровня БТ-7 относится к категории легких танков, но не обладает дополнительными способностями. Каждый класс техники обладает своими уникальными способностями, которые будут описаны далее в методах классов.

После исследования танк выезжает на поле боя и делает несколько выстрелов, после чего возвращается в ангар до конца боя. Задача пользователя исследовать танки и нанести 8000 суммарного урона по базе противника. Если суммарный нанесенный урон составляет не менее 8000, то пользователь одерживает победу, иначе терпит поражение.

После окончания боя пользователь может повторить моделирование танкового сражения, запустив приложение еще раз. У него появится возможность выбрать другую стратегию исследования танков или попробовать повторить предыдущую.

3. Диаграмма классов

Диаграмма классов представлена на рис. 2 и рис. 3.

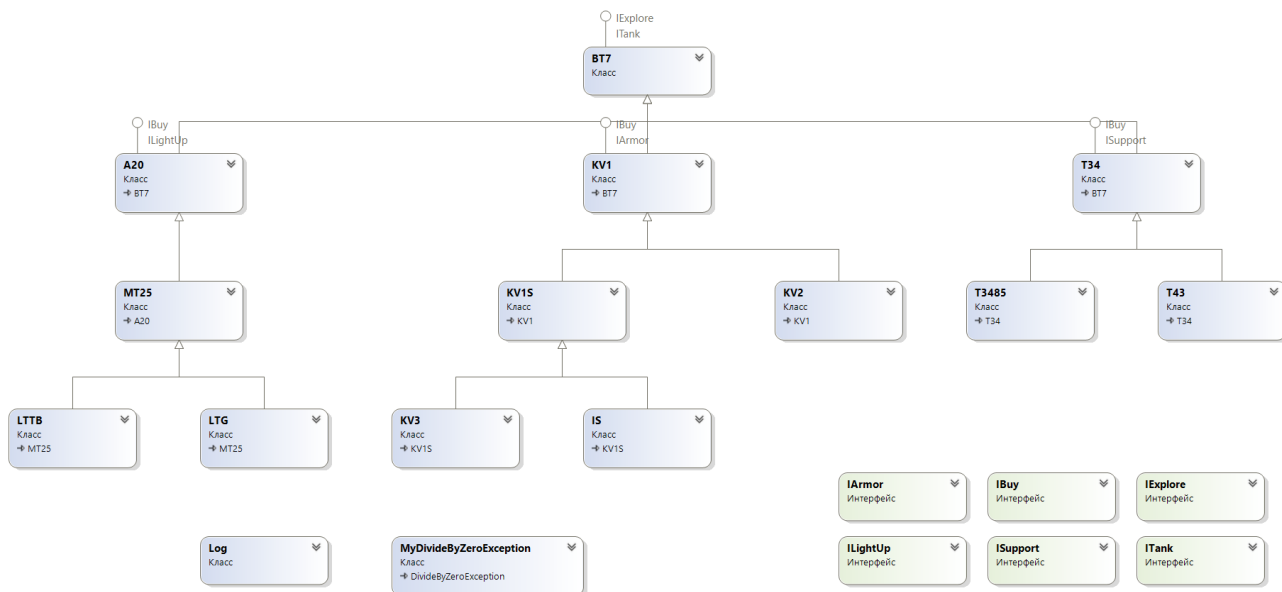


Рис. 2. Диаграмма классов

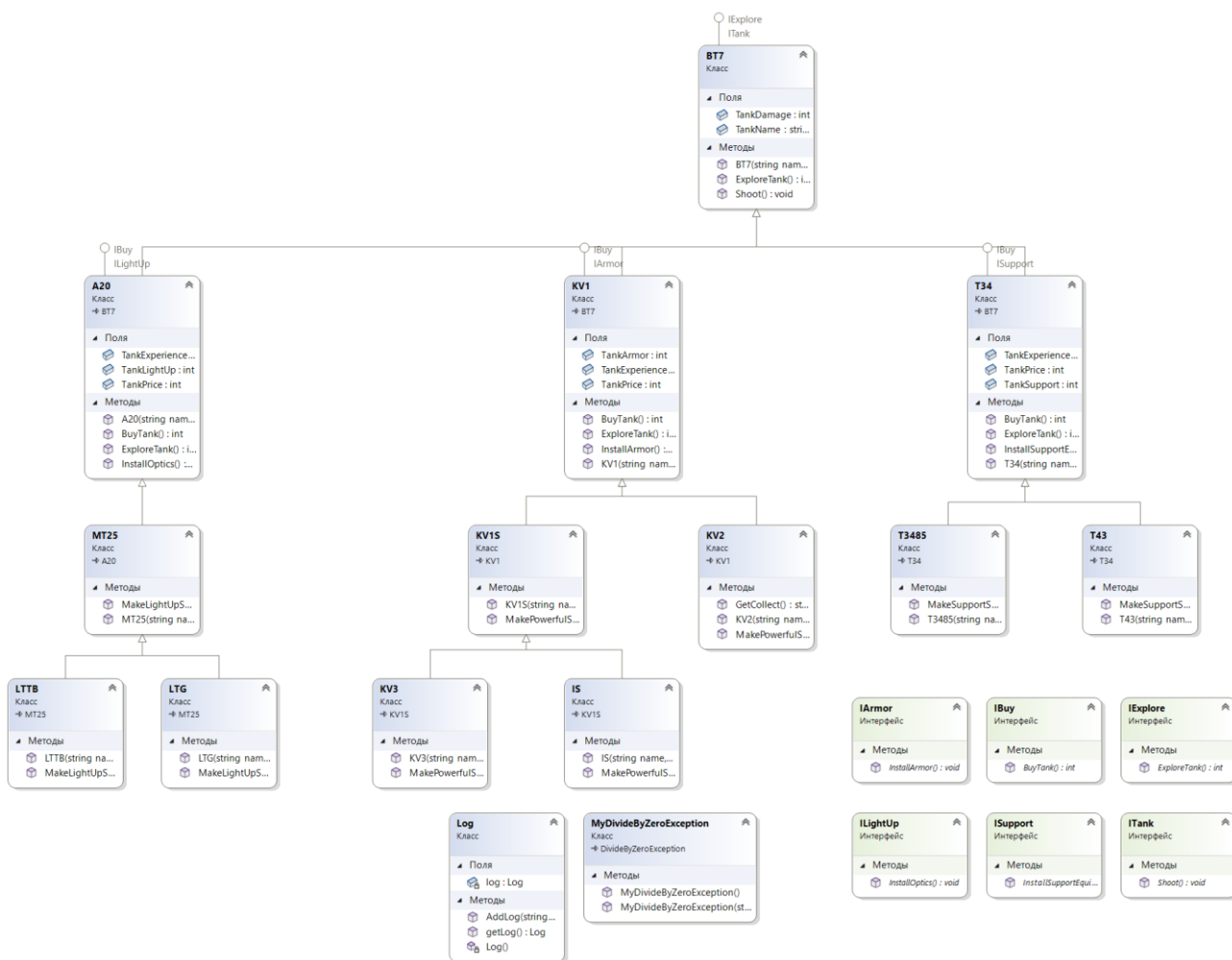


Рис. 3. Диаграмма классов в развернутом виде

4. Описание полей и методов классов

Описание полей и методов класса BT7 приведено в табл. 2.

Таблица 2. Поля и методы класса BT7

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Поля					
TankName	string	Название танка	public	-	-
TankDamage	int	Нанесенный урон за один выстрел	public		
Методы					
BT7	-	Конструктор класса	public	string name, int damage	-
ExploreTank	int	Исследование танка. Возвращает количество требуемого опыта для исследования. В данном случае 0, так как танк первого уровня	public virtual	-	int 0
Shoot	void	Фиксирует в логе программы урон за выстрел для танка	public	-	-

Описание полей и методов класса KV1 приведено в табл. 3. Данный класс является дочерним к классу BT7, поэтому наследует все его поля и методы.

Таблица 3. Поля и методы класса KV1

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Поля					
TankArmor	int	Показатель брони танка – уникальный показатель тяжелого танка	public	-	-
TankPrice	int	Стоимость покупки танка	public	-	-
TankExperience	int	Стоимость исследования танка	public	-	-

Таблица 3. Продолжение

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Методы					
KV1	-	Конструктор класса	public	string name, int damage, int experience, int price, int armor	-
ExploreTank	int	Возвращает требуемое количество опыта для исследования танка	public override	-	int TankExperience
BuyTank	int	Возвращает требуемое количество денег для покупки танка	public	-	int TankPrice
InstallArmor	void	Фиксирует в логе программы показатель брони тяжелого танка	public	-	-

Описание полей и методов класса T34 приведено в табл. 4. Данный класс является дочерним к классу BT7, поэтому наследует все его поля и методы.

Таблица 4. Поля и методы класса T34

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Поля					
TankSupport	int	Показатель поддержки танка – уникальный показатель среднего танка	public	-	-
TankPrice	int	Стоимость покупки танка	public	-	-
TankExperience	int	Стоимость исследования танка	public	-	-

Таблица 4. Продолжение

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Методы					
T34	-	Конструктор класса	public	string name, int damage, int experience, int price, int support	-
ExploreTank	int	Возвращает требуемое количество опыта для исследования танка	public override	-	int TankExperience
BuyTank	int	Возвращает требуемое количество денег для покупки танка	public	-	int TankPrice
InstallSupportEquipment	void	Фиксирует в логе программы показатель поддержки среднего танка	public	-	-

Описание полей и методов класса A20 приведено в табл. 5. Данный класс является дочерним к классу BT7, поэтому наследует все его поля и методы.

Таблица 5. Поля и методы класса A20

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Поля					
TankLightUp	int	Показатель обзора («засвета») танка – уникальный показатель легкого танка	public	-	-
TankPrice	int	Стоимость покупки танка	public	-	-
TankExperience	int	Стоимость исследования танка	public	-	-

Таблица 5. Продолжение

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Методы					
A20	-	Конструктор класса	public	string name, int damage, int experience, int price, int light_up	-
ExploreTank	int	Возвращает требуемое количество опыта для исследования танка	public override	-	int TankExperience
BuyTank	int	Возвращает требуемое количество денег для покупки танка	public	-	int TankPrice
InstallOptics	void	Фиксирует в логге программы показатель обзора («засвета») легкого танка	public	-	-

Описание методов класса KV1S приведено в табл. 6. Данный класс является дочерним к классу KV1, поэтому наследует все его поля и методы.

Таблица 6. Поля и методы класса KV1S

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Методы					
KV1S	-	Конструктор класса	public	string name, int damage, int experience, int price, int armor	-
MakePowerfulShot	int	Уникальная способность тяжелого танка 3 уровня и выше. Наносит дополнительный выстрел с повышенным уроном, равным x2 от базового урона	public	-	int TankDamage * 2

Описание методов класса KV2 приведено в табл. 7. Данный класс является дочерним к классу KV1, поэтому наследует все его поля и методы.

Таблица 7. Поля и методы класса KV2

Имя	Тип	Описание	Область видимо сти	Входные параметры	Выходные параметры
Методы					
KV2	-	Конструктор класса	public	string name, int damage, int experience, int price, int armor	-
MakePowerfulShot	int	Уникальная способность тяжелого танка 3 уровня и выше. Наносит дополнительный выстрел с повышенным уроном, равным x3 от базового урона	public	-	int TankDamage * 3
GetCollect	string	Возвращается строку с сообщением, что данный танк является уникальным и коллекционным (так как обладает самым большим разовым уроном из всех)	public		string "This is a unique collectible tank!"

Описание методов класса T3485 приведено в табл. 8. Данный класс является дочерним к классу T34, поэтому наследует все его поля и методы.

Таблица 8. Поля и методы класса T3485

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Методы					
T3485	-	Конструктор класса	public	string name, int damage, int experience, int price, int support	-
MakeSupportShot	int	Уникальная способность среднего танка 3 уровня. Наносит дополнительный выстрел с повышенным уроном, на 20 больше базового урона	public	-	int TankDamage +20

Описание методов класса T43 приведено в табл. 9. Данный класс является дочерним к классу T34, поэтому наследует все его поля и методы.

Таблица 9. Поля и методы класса T43

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Методы					
T43	-	Конструктор класса	public	string name, int damage, int experience, int price, int support	-
MakeSupportShot	int	Уникальная способность среднего танка 3 уровня. Наносит дополнительный выстрел с повышенным уроном, на 30 больше базового урона	public	-	int TankDamage +30

Описание методов класса MT25 приведено в табл. 10. Данный класс является дочерним к классу A20, поэтому наследует все его поля и методы.

Таблица 10. Поля и методы класса MT25

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Методы					
MT25	-	Конструктор класса	public	string name, int damage, int experience, int price, int light_up	-
MakeLightUpShot	int	Уникальная способность легкого танка 3 уровня и выше. Наносит дополнительный выстрел с пониженным уроном, на 20 меньше базового урона	public	-	int TankDamage -20

Описание методов класса KV3 приведено в табл. 11. Данный класс является дочерним к классу KV1S, поэтому наследует все его поля и методы.

Таблица 11. Поля и методы класса KV3

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Методы					
KV3	-	Конструктор класса	public	string name, int damage, int experience, int price, int armor	-
MakePowerfulShot	int	Уникальная способность тяжелого танка 4 уровня Наносит дополнительный выстрел с повышенным уроном, равным $x(2 + \text{chance})$ от базового урона. Значение chance равно либо 0, либо 1. Т.е. танк имеет возможность нанести выстрел с еще более высоким уроном	public	int chance-	int TankDamage * (2+chance)

Описание методов класса IS приведено в табл. 12. Данный класс является дочерним к классу KV1S, поэтому наследует все его поля и методы.

Таблица 12. Поля и методы класса IS

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Методы					
KV3	-	Конструктор класса	public	string name, int damage, int experience, int price, int armor	-
MakePowerfulShot	int	Уникальная способность тяжелого танка 4 уровня Наносит дополнительный выстрел с повышенным уроном, равным $x(2 + \text{chance1} - \text{chance2})$ от базового урона. Значение chance1 равно 0/1/2, а chance2 равно либо 0, либо 1. Т.е. танк имеет возможность нанести выстрел с более высоким/низким или равным уроном	public	int chance1, int chance2-	int TankDamage * (2 + chance1 - chance2)

Описание методов класса LTTB приведено в табл. 13. Данный класс является дочерним к классу MT25, поэтому наследует все его поля и методы.

Таблица 13. Поля и методы класса LTTB

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Методы					
LTTB	-	Конструктор класса	public	string name, int damage, int experience, int price, int light_up	-
MakeLightUpShot	int	Уникальная способность легкого танка 4 уровня и выше. Наносит дополнительный выстрел с пониженным на 10 единиц или таким же уроном. Значение chance равно либо 10, либо 20.	public	int chance-	int TankDamage -20+ chance

Описание методов класса LTG приведено в табл. 14. Данный класс является дочерним к классу MT25, поэтому наследует все его поля и методы.

Таблица 14. Поля и методы класса LTG

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Методы					
LTG	-	Конструктор класса	public	string name, int damage, int experience, int price, int light_up	-
MakeLightUpShot	int	Уникальная способность легкого танка 4 уровня и выше. Наносит дополнительный выстрел с пониженным или таким же уроном. Значение chance1 равно 10/20/30/40. Значение chance2 равно 0/10/20.	public	int chance1, int chance2	int TankDamage -20 + chance1 - chance2

Класс Log используется для логирования всех происходящих действий в программе Описание полей и методов данного класса приведено в табл. 15.

Таблица 15. Поля и методы класса Log

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Поля					
log	Log	Экземпляр класса Log	private	-	-
Методы					
Log	-	Конструктор класса	private	-	-
getLog	Log	Создает только один экземпляр класса.	public	-	Экземпляр класса Log -
AddLog	void	Записывает строку в лог файл	public	string msg – сообщение для записи в лог файл	-

Класс MyDivideByZeroException является пользовательским исключением, которое наследуется от стандартного исключения DivideByZeroException. Этот класс используется для создания переопределенного исключения при делении на ноль. Описание полей и методов данного класса приведено в табл. 16.

Таблица 16. Поля и методы класса MyDivideByZeroException

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Методы					
MyDivideByZeroException	-	Конструктор без параметров вызывает конструктор базового класса (DivideByZeroException) с заданным сообщением об ошибке. Это сообщение будет выводиться при возникновении исключения.	public	-	-
MyDivideByZeroException		Конструктор с параметром для сообщения об ошибке позволяет задать пользовательское сообщение при создании объекта исключения.	public	string message-	-

5. Описание характеристик танков

Описание характеристик танков, использующихся при моделировании танкового сражения, приведено в табл. 17. Дополнительные выстрелы происходят с заданной вероятностью. Как правило, они наносят количество урона, которое отличается от базового урона.

Таблица 17. Описание характеристик танков

Танк	Базовый урон за выстрел	Количество выстрелов с базовым уроном	Дополнительный выстрел (для танка 3 и 4 уровня) в соответствии с особой способности танка.		Дополнительный специальный выстрел (для танков 4 уровня)	
			Шанс	Урон	Шанс	Урон
БТ-7	100	От 1 до 3				
А-20	160	От 1 до 5				
Т-34	160	От 1 до 5				
КВ-1	200	От 1 до 4				
МТ-25	200	От 1 до 6	50%	180		
Т3485	250	От 1 до 5	50%	270		
Т43	250	От 1 до 5	33%	280		
КВ-2	560	От 1 до 2	25%	1680		
КВ-1С	400	От 1 до 3	33%	800		
ЛТТБ	310	От 1 до 7	50%	290	100%	300 или 310
ЛТГ	310	От 1 до 7	50%	290	100%	280, 290, 300, 310, 320 или 330
КВ-3	450	От 1 до 4	33%	900	100%	900 или 1350
ИС	450	От 1 до 4	33%	900	100%	450, 900, 1350 или 1800

6. Описание пользовательского интерфейса

Пользовательский интерфейс приведен на рис. 4 – 6.

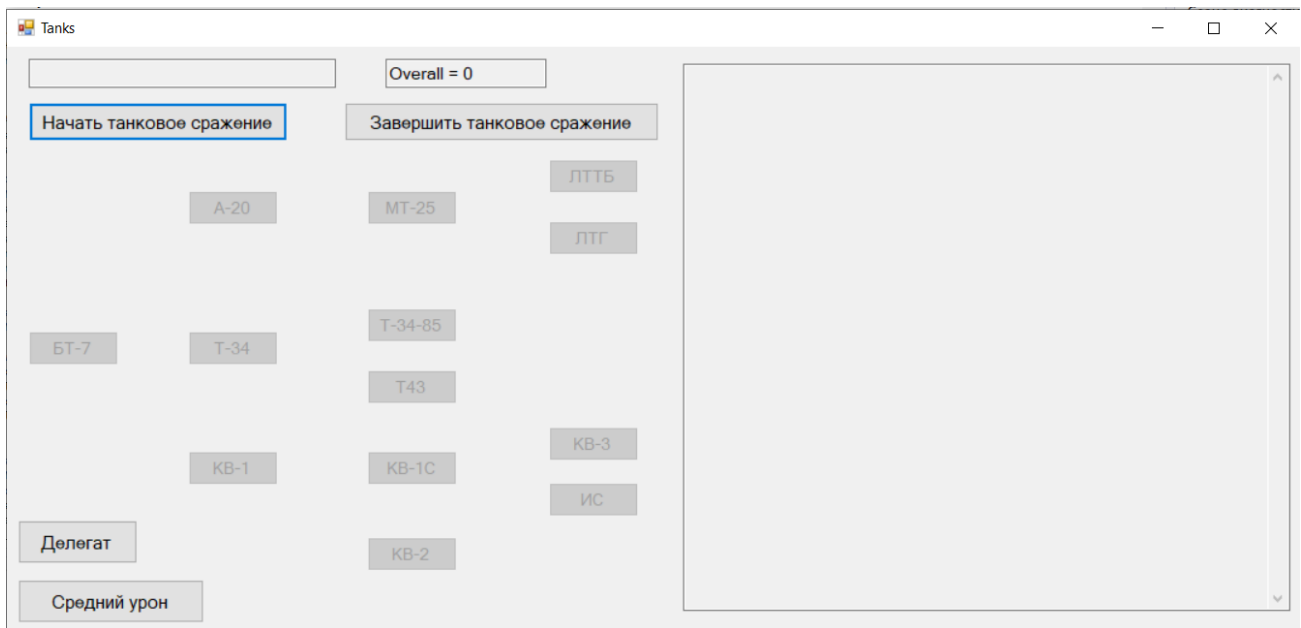


Рис. 4. Пользовательский интерфейс после запуска программы

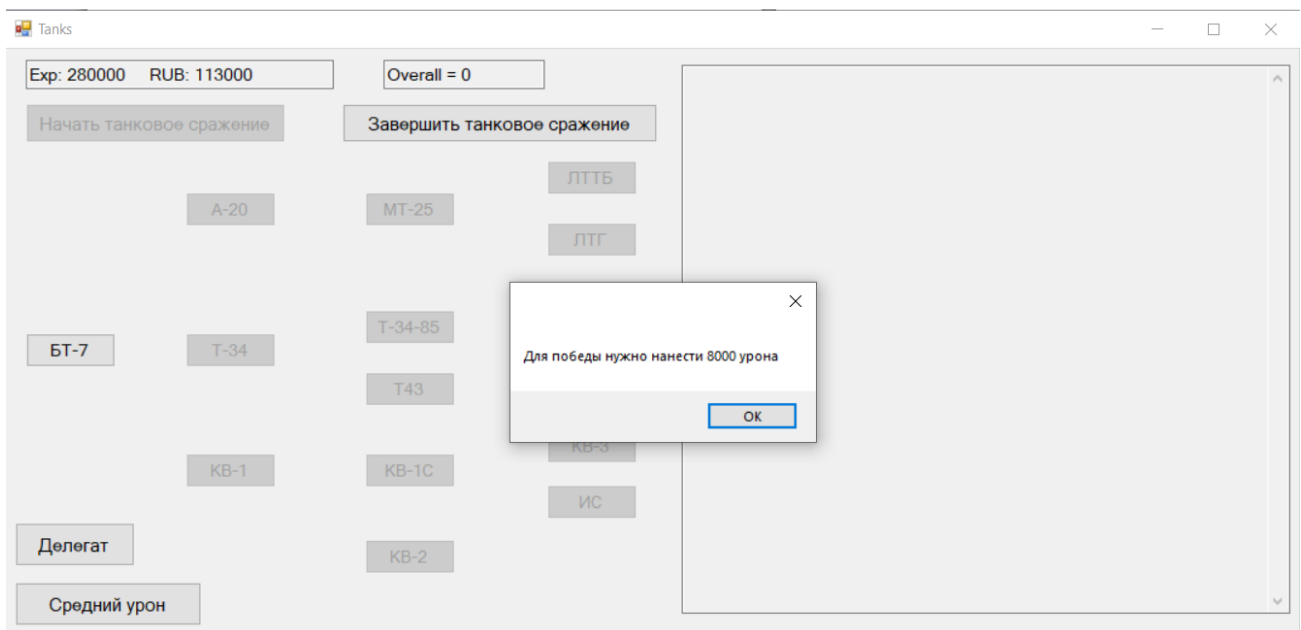


Рис. 5. Пользовательский интерфейс после начала танкового сражения

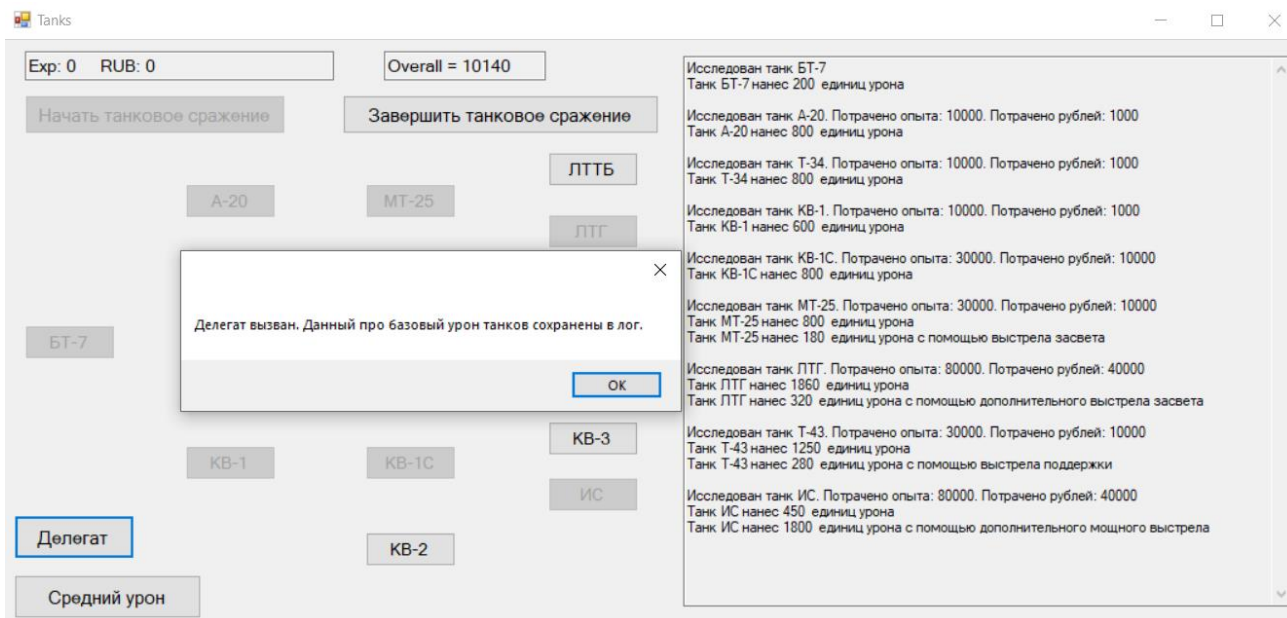


Рис. 6. Пользовательский интерфейс во время танкового сражения

Пользователь может исследовать танки с помощью нажатия соответствующей кнопки с названием танка. После этого у него спишется количество ресурсов в соответствии с ценой танка.

Пользователь может завершить танковое сражения и узнать результат с помощью нажатия на кнопку «Завершить танковое сражение».

Пользователь может посмотреть средний урон за выстрел с помощью кнопки «Средний урон». Если количество выстрелов равно нулю, появится пользовательское исключение при делении на ноль.

Пользователь может вызвать многоадресный делегат с помощью нажатия на кнопку «Делегат». Будут вызваны соответствующие методы, а их результат отобразится в логе программы.

В верхних TextBox написано количество ресурсов пользователя и суммарный нанесенный урон. Эти показатели обновляются после исследования танков. В правом TextBox выводится информация об эффективности исследованных танков.

7. Тестирование программы

Тестирование программы приведено на рис. 7 – 17. В тестах заметно, что в каждом случае результат моделирования различный, так как вносится вероятностный случайный параметр

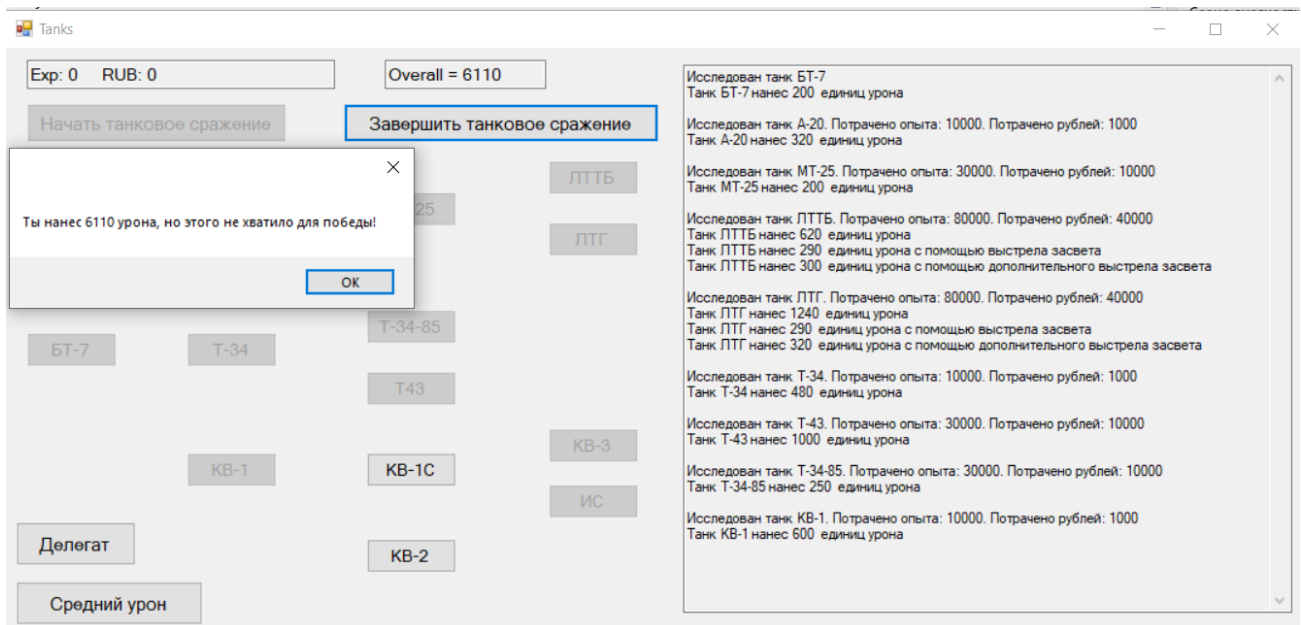


Рис. 7. Тест 1. Поражение при исследовании всех легких и средних танков

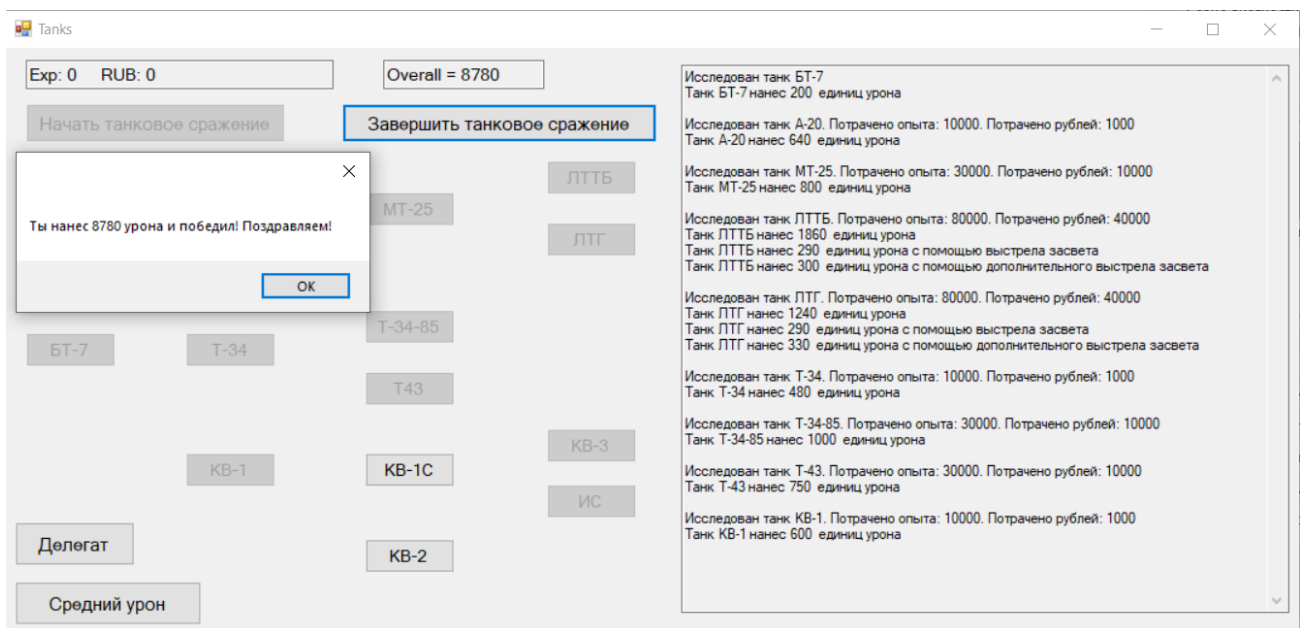


Рис. 8. Тест 2. Победа при исследовании всех легких и средних танков

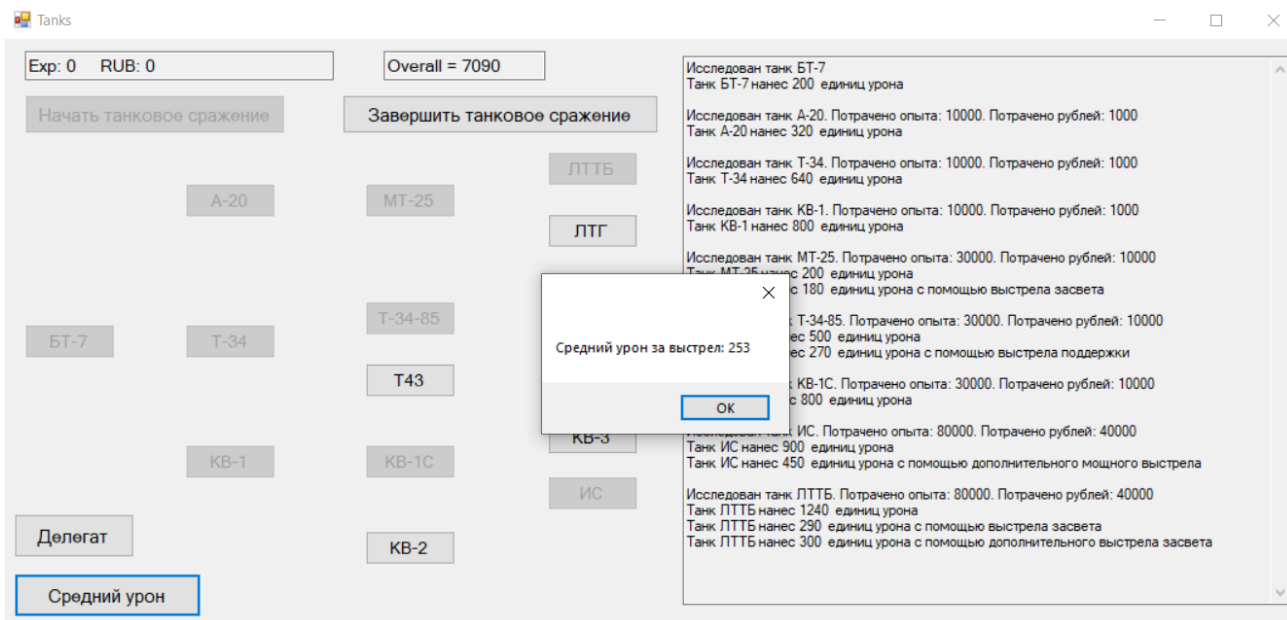


Рис. 9. Тест 3. Проверка среднего урона

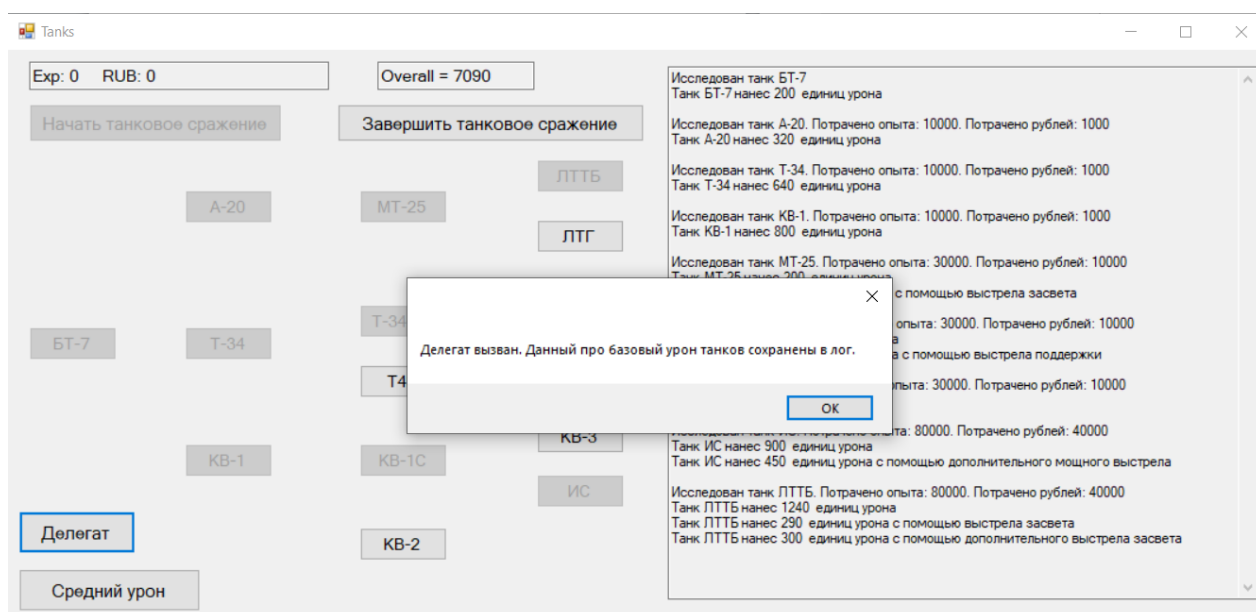


Рис. 10. Тест 4. Вызов делегата

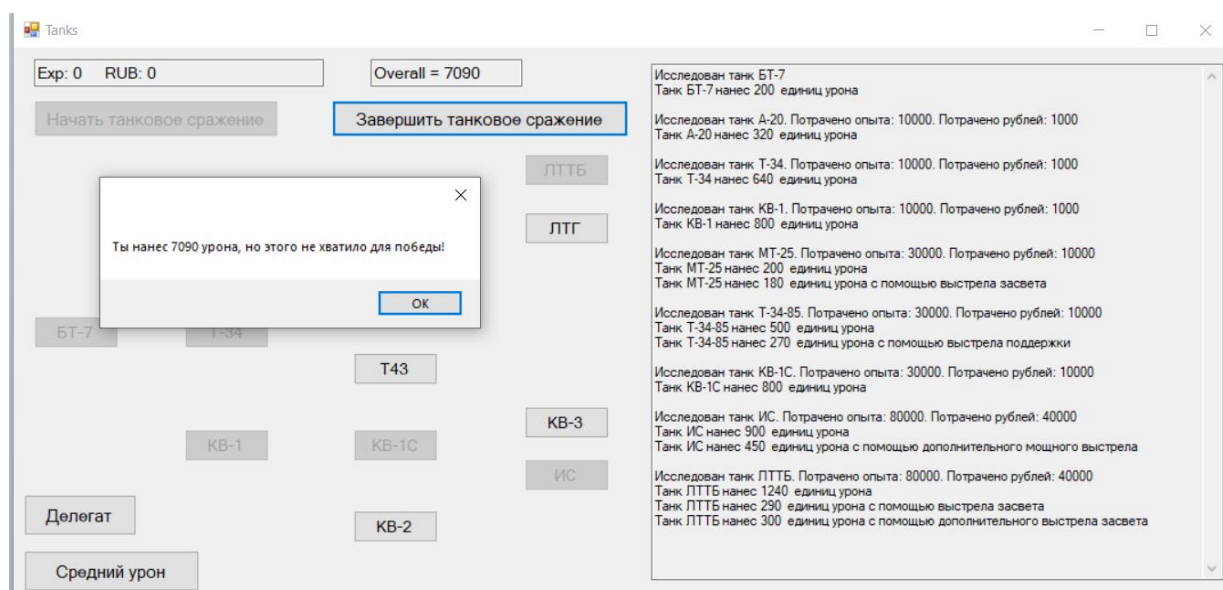


Рис. 11. Тест 5. Поражение при произвольном исследовании танков

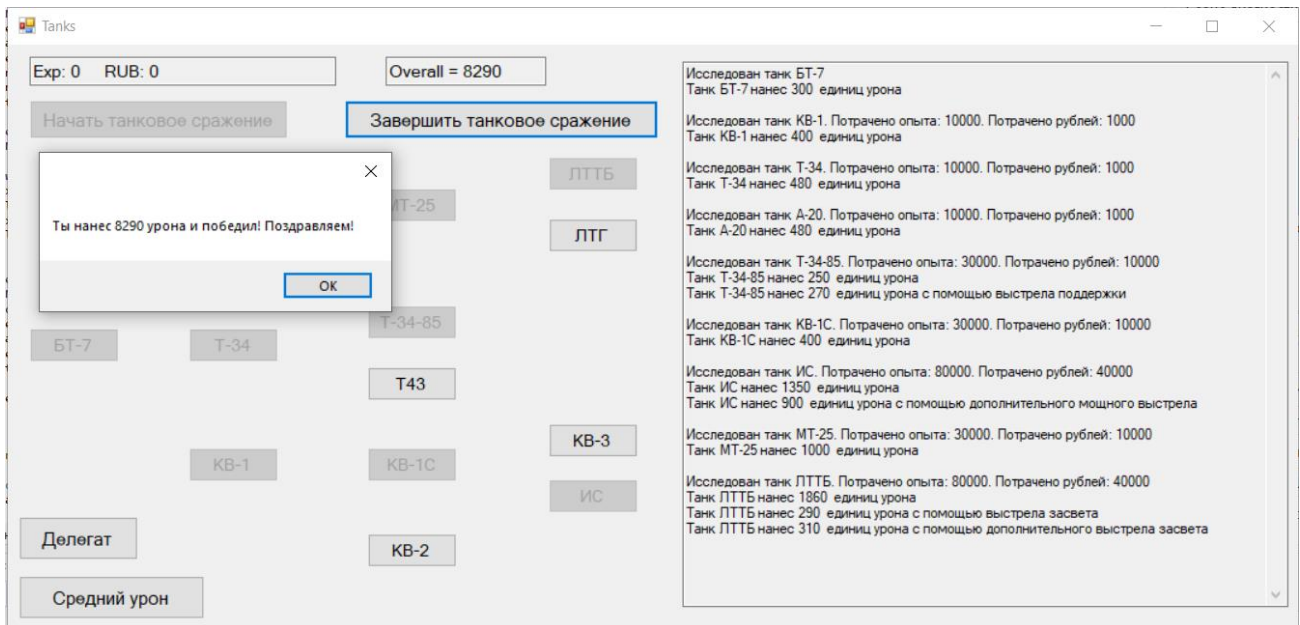


Рис. 12. Тест 6. Победа при таком же исследовании танков

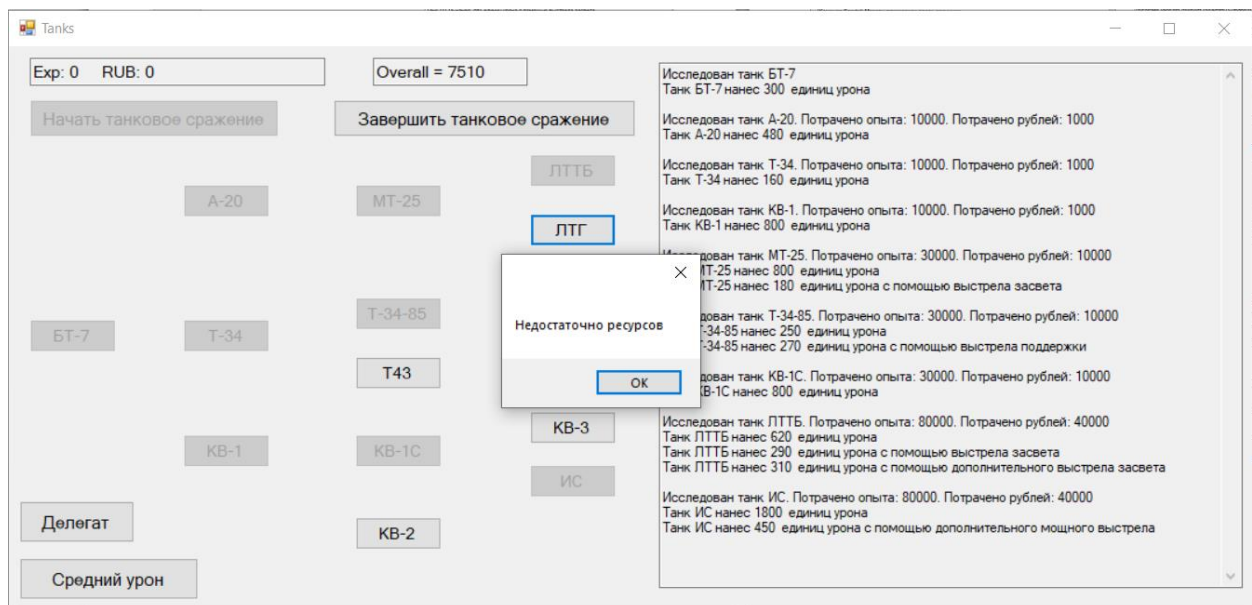


Рис. 13. Тест 7. Попытка исследовать танк при нехватке ресурсов

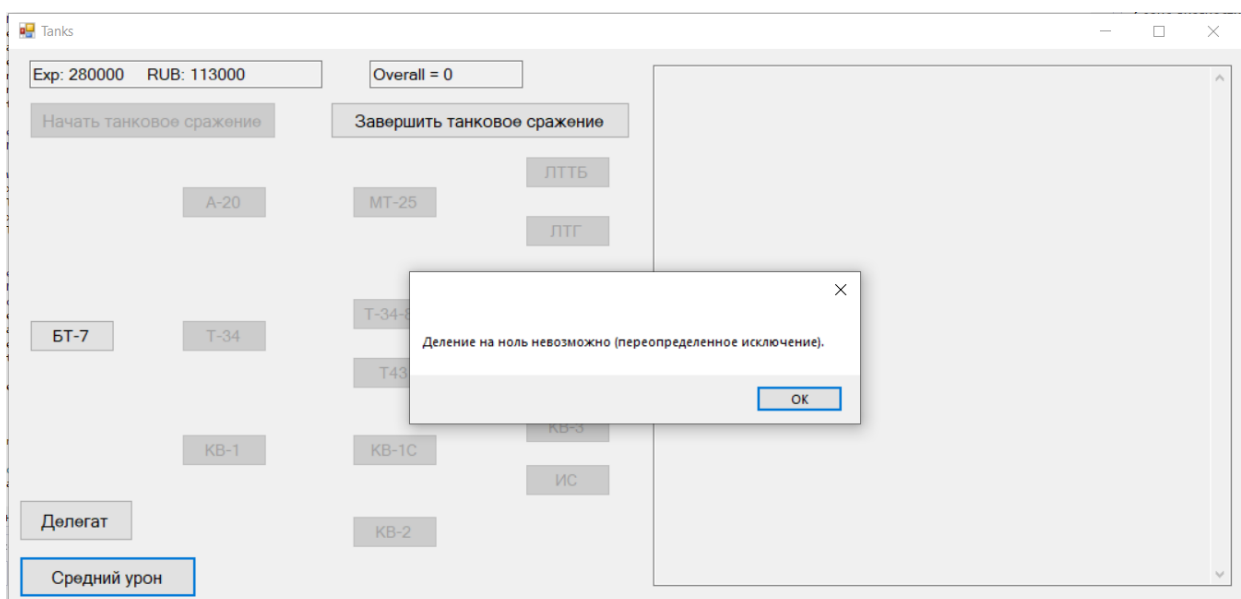


Рис. 14. Тест 8. Попытка деления на ноль

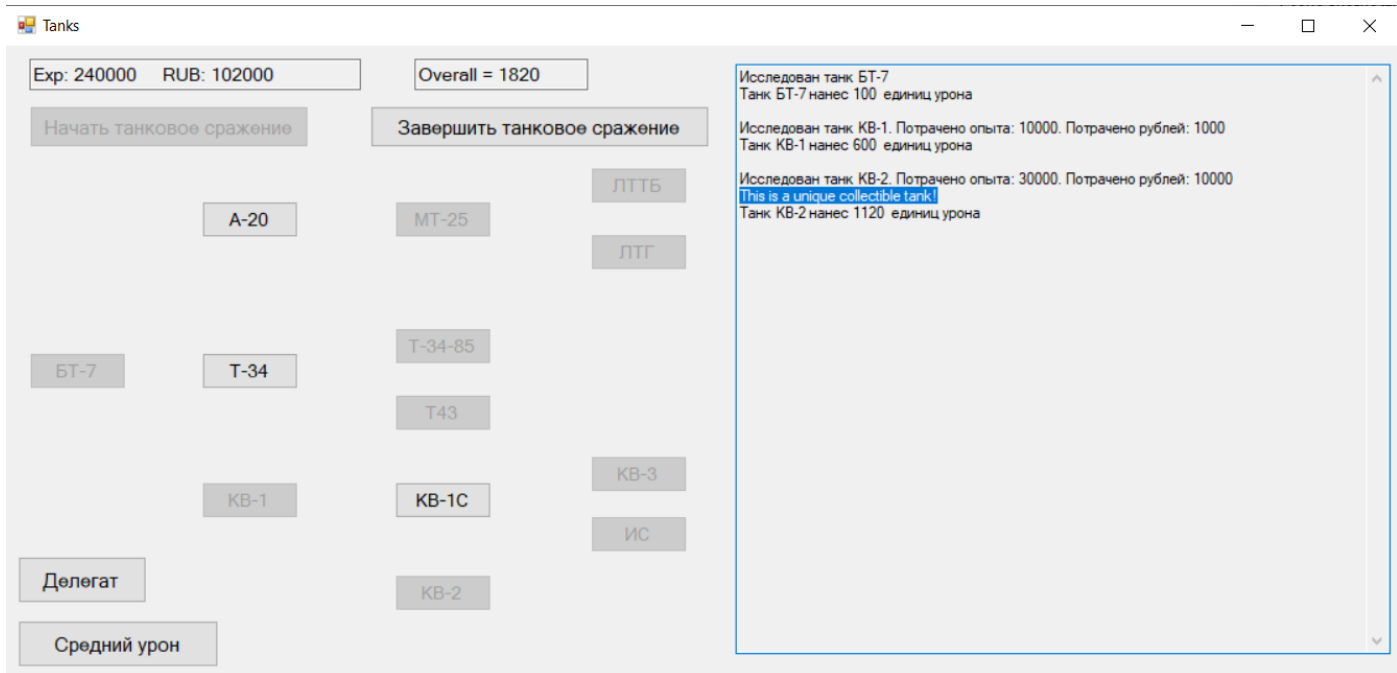


Рис. 15. Тест 9. Исследование коллекционного танка КВ-2

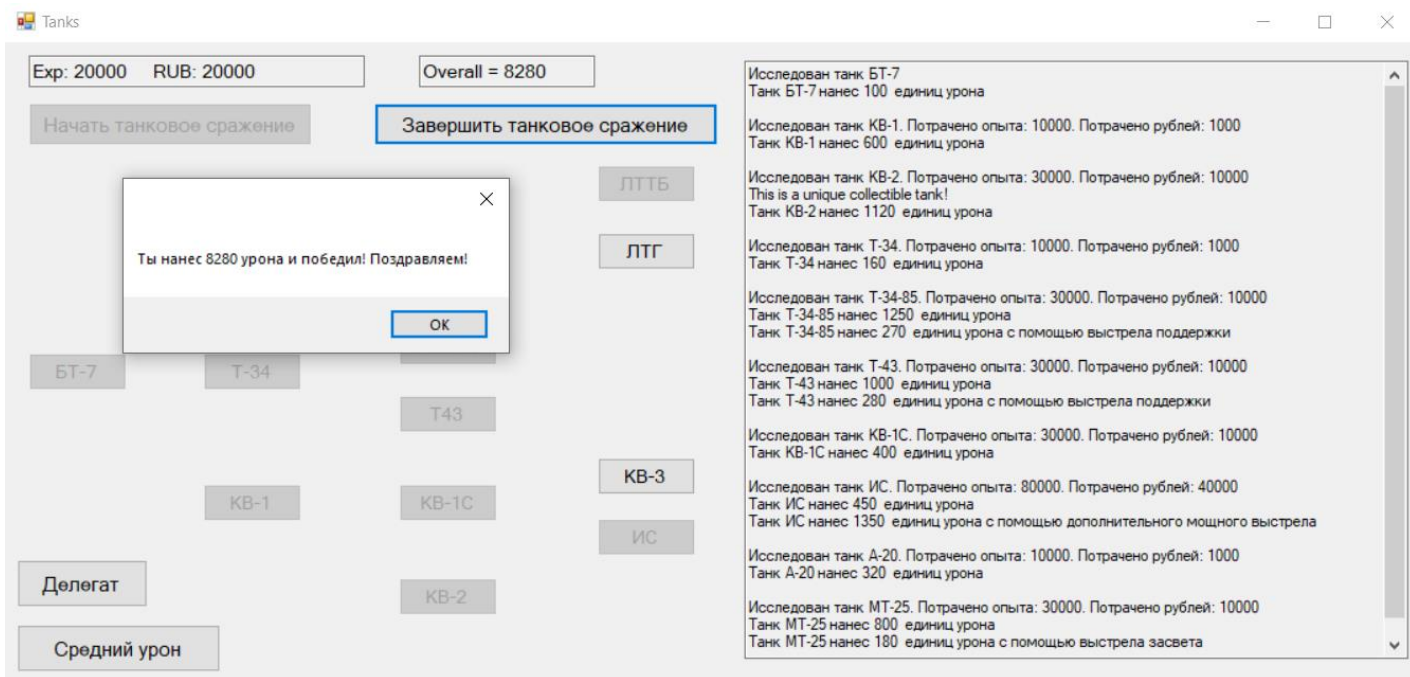


Рис. 16. Тест 10. Победа в бою с КВ-2 и оставшимися ресурсами

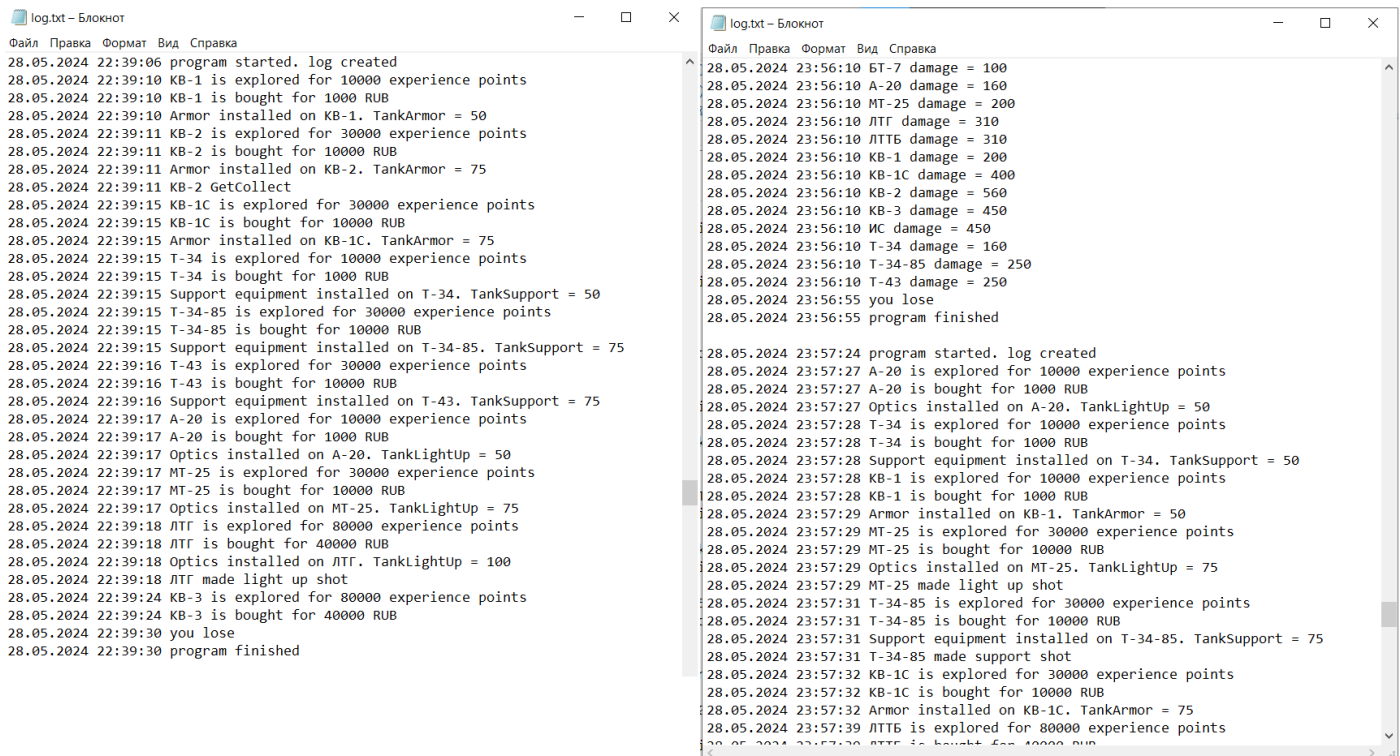


Рис. 17. Пример логирования программы. В правой части присутствует вызов многоадресного делегата

8. Выводы

Эта программа моделирует танковое сражение, результат которого зависит как от пользователя, так и от вероятности происхождения событий. Также программа включает логирование действий.

1. Программа демонстрирует использование интерфейсов для определения общих методов, которые должны быть реализованы классами.

2. Созданные классы демонстрируют использование наследования для организации иерархии объектов.

3. В классе KV1, T34, A20 переопределен метод ExploreTank, что позволяет каждому танку этих классов иметь свою уникальную реализацию в отличие от родительского класса.

4. Программа использует логирование с помощью класса Log, чтобы записывать различные события.

5. Программа содержит пользовательское исключение MyDivideByZeroException, которое наследует от стандартного исключения DivideByZeroException и используется для обработки ситуаций деления на ноль.

6. Программа генерирует случайные значения для моделирования количества выстрелов на каждом танке, что позволяет добавить непредсказуемости в результат танкового сражения.

7. Программа демонстрирует использование многоадресного делегата для вызова нескольких методов.

Приложение А. Код иерархии классов.

```
using System;
using System.IO;

namespace Modeling
{
    // класс для логирования программы
    public class Log
    {
        private static Log log;
        private Log() { }
        public static Log getLog()
        {
            if (log == null)
                log = new Log();
            return log;
        }
        public async void AddLog(string msg)
        {
            string path = "log.txt";
            using (StreamWriter writer = new StreamWriter(path, true))
            {
                await writer.WriteLineAsync(DateTime.Now.ToString() + " " + msg);
            }
        }
    }

    // Определение интерфейсов
    // =====
    // Способности танка
    interface ITank
    {
        void Shoot();
    }
    // Исследование танка (для 1 - 4 уровня)
    interface IExplore
    {
        int ExploreTank();
    }
    // Покупка танка (для 2 - 4 уровня)
    interface IBuy
    {
        int BuyTank();
    }
    // Броня - характеристика тяжелого танка
    interface IArmor
    {
        void InstallArmor();
    }
    // Поддержка - характеристика среднего танка
    interface ISupport
    {
```



```

        void InstallSupportEquipment();
    }
    // Засвет - характеристика легкого танка
    interface ILightUp
    {
        void InstallOptics();
    }
    // =====

    // Уровень 1
    // =====
    public class BT7 : IExplore, ITank
    {
        public string TankName;
        public int TankDamage;

        public BT7(string name, int damage)
        {
            TankName = name;
            TankDamage = damage;
        }

        public virtual int ExploreTank()
        {
            var logger = Log.getLog();
            logger.AddLog(TankName + " is explored");
            return 0;
        }

        public void Shoot()
        {
            var logger = Log.getLog();
            logger.AddLog(TankName + " damage = " + TankDamage);
        }
    }
    // =====

    // Уровень 2
    // =====
    public class KV1 : BT7, IBuy, IArmor
    {
        public int TankExperience;
        public int TankPrice;
        public int TankArmor;
        public KV1(string name, int damage, int experience, int price, int armor) :
base(name, damage)
        {
            TankName = name;
            TankDamage = damage;
            TankExperience = experience;
            TankPrice = price;
            TankArmor = armor;
        }
    }

```

```

    public override int ExploreTank()
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " is explored for " + TankExperience + " experience
points");
        return TankExperience;
    }

    public int BuyTank()
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " is bought for " + TankPrice + " RUB");
        return TankPrice;
    }

    public void InstallArmor()
    {
        var logger = Log.getLog();
        logger.AddLog("Armor installed on " + TankName + ". TankArmor = " +
TankArmor);
    }
}

public class T34 : BT7, IBuy, ISupport
{
    public int TankExperience;
    public int TankPrice;
    public int TankSupport;
    public T34(string name, int damage, int experience, int price, int support) :
base(name, damage)
    {
        TankName = name;
        TankDamage = damage;
        TankExperience = experience;
        TankPrice = price;
        TankSupport = support;
    }

    public override int ExploreTank()
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " is explored for " + TankExperience + " experience
points");
        return TankExperience;
    }

    public int BuyTank()
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " is bought for " + TankPrice + " RUB");
        return TankPrice;
    }
}

```

```

        public void InstallSupportEquipment()
        {
            var logger = Log.getLog();
            logger.AddLog("Support equipment installed on " + TankName + ". TankSupport = " + TankSupport);
        }
    }

    public class A20 : BT7, IBuy, ILightUp
    {
        public int TankExperience;
        public int TankPrice;
        public int TankLightUp;
        public A20(string name, int damage, int experience, int price, int light_up) :
        base(name, damage)
        {
            TankName = name;
            TankDamage = damage;
            TankExperience = experience;
            TankPrice = price;
            TankLightUp = light_up;
        }

        public override int ExploreTank()
        {
            var logger = Log.getLog();
            logger.AddLog(TankName + " is explored for " + TankExperience + " experience points");
            return TankExperience;
        }

        public int BuyTank()
        {
            var logger = Log.getLog();
            logger.AddLog(TankName + " is bought for " + TankPrice + " RUB");
            return TankPrice;
        }

        public void InstallOptics()
        {
            var logger = Log.getLog();
            logger.AddLog("Optics installed on " + TankName + ". TankLightUp = " + TankLightUp);
        }
    }

    // =====

    // Уровень 3
    // =====
    public class KV1S : KV1
    {
        public KV1S(string name, int damage, int experience, int price, int armor) :
        base(name, damage, experience, price, armor)
        {

```

```

        TankName = name;
        TankDamage = damage;
        TankExperience = experience;
        TankPrice = price;
        TankArmor = armor;
    }

    public int MakePowerfulShot()
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " made powerful shot");
        return TankDamage * 2;
    }
}

public class KV2 : KV1
{
    public KV2(string name, int damage, int experience, int price, int armor) :
base(name, damage, experience, price, armor)
    {
        TankName = name;
        TankDamage = damage;
        TankExperience = experience;
        TankPrice = price;
        TankArmor = armor;
    }

    public int MakePowerfulShot()
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " made powerful shot");
        return TankDamage * 3;
    }

    public string GetCollect()
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " GetCollect");
        return "This is a unique collectible tank!";
    }
}

public class T3485 : T34
{
    public T3485(string name, int damage, int experience, int price, int support) :
base(name, damage, experience, price, support)
    {
        TankName = name;
        TankDamage = damage;
        TankExperience = experience;
        TankPrice = price;
        TankSupport = support;
    }
}

```

```

    public int MakeSupportShot()
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " made support shot");
        return TankDamage + 20;
    }
}

public class T43 : T34
{
    public T43(string name, int damage, int experience, int price, int support) :
base(name, damage, experience, price, support)
    {
        TankName = name;
        TankDamage = damage;
        TankExperience = experience;
        TankPrice = price;
        TankSupport = support;
    }

    public int MakeSupportShot()
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " made support shot");
        return TankDamage + 30;
    }
}

public class MT25 : A20
{
    public MT25(string name, int damage, int experience, int price, int light_up) :
base(name, damage, experience, price, light_up)
    {
        TankName = name;
        TankDamage = damage;
        TankExperience = experience;
        TankPrice = price;
        TankLightUp = light_up;
    }

    public int MakeLightUpShot()
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " made light up shot");
        return TankDamage - 20;
    }
}

// =====

// Уровень 4
// =====
public class KV3 : KV1S
{

```

```

    public KV3(string name, int damage, int experience, int price, int armor) :
base(name, damage, experience, price, armor)
    {
        TankName = name;
        TankDamage = damage;
        TankExperience = experience;
        TankPrice = price;
        TankArmor = armor;
    }

    public int MakePowerfulShot(int chance)
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " made powerful shot");
        return TankDamage * (2 + chance);
    }
}

public class IS : KV1S
{
    public IS(string name, int damage, int experience, int price, int armor) :
base(name, damage, experience, price, armor)
    {
        TankName = name;
        TankDamage = damage;
        TankExperience = experience;
        TankPrice = price;
        TankArmor = armor;
    }

    public int MakePowerfulShot(int chance1, int chance2)
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " made powerful shot");
        return TankDamage * (2 + chance1 - chance2);
    }
}

public class LTTB : MT25
{
    public LTTB(string name, int damage, int experience, int price, int light_up) :
base(name, damage, experience, price, light_up)
    {
        TankName = name;
        TankDamage = damage;
        TankExperience = experience;
        TankPrice = price;
        TankLightUp = light_up;
    }

    public int MakeLightUpShot(int chance)
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " made light up shot");

```

```

        return TankDamage - 20 + chance;
    }
}

public class LTG : MT25
{
    public LTG(string name, int damage, int experience, int price, int light_up) :
base(name, damage, experience, price, light_up)
    {
        TankName = name;
        TankDamage = damage;
        TankExperience = experience;
        TankPrice = price;
        TankLightUp = light_up;
    }

    public int MakeLightUpShot(int chance1, int chance2)
    {
        var logger = Log.getLog();
        logger.AddLog(TankName + " made light up shot");
        return TankDamage - 20 + chance1 - chance2;
    }
}
// =====

public class MyDivideByZeroException : DivideByZeroException
{
    public MyDivideByZeroException() : base("Деление на ноль невозможно
(переопределенное исключение).")
    {
        // Конструктор без параметров вызывает базовый конструктор с сообщением об
ошибке
    }

    public MyDivideByZeroException(string message) : base(message)
    {
        // Конструктор с параметром для сообщения об ошибке
    }
}
}

```

Приложение Б. Код основной программы

```
using System;
using System.Windows.Forms;

namespace Modeling
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            var logger = Log.getLog();
            logger.AddLog("program started. log created");
            InitializeComponent();
            this.StartPosition = FormStartPosition.CenterScreen;
            button2.Enabled = false;
            button3.Enabled = false;
            button4.Enabled = false;
            button5.Enabled = false;
            button6.Enabled = false;
            button7.Enabled = false;
            button8.Enabled = false;
            button9.Enabled = false;
            button10.Enabled = false;
            button11.Enabled = false;
            button12.Enabled = false;
            button13.Enabled = false;
            button14.Enabled = false;
        }

        int Exp = 280000;
        int Money = 113000;

        // Создание экземпляров классов
        BT7 bt7 = new BT7("БТ-7", 100);

        // Тяжелые танки
        // -----
        KV1 kv1 = new KV1("KB-1", 200, 10000, 1000, 50);

        KV1S kv1s = new KV1S("KB-1C", 400, 30000, 10000, 75);
        KV2 kv2 = new KV2("KB-2", 560, 30000, 10000, 75);

        KV3 kv3 = new KV3("KB-3", 450, 80000, 40000, 100);
        IS is = new IS("ИС", 450, 80000, 40000, 100);
        // -----

        // Средние танки
        // -----
        T34 t34 = new T34("Т-34", 160, 10000, 1000, 50);

        T3485 t3485 = new T3485("Т-34-85", 250, 30000, 10000, 75);
        T43 t43 = new T43("Т-43", 250, 30000, 10000, 75);
        // -----
```



```

// Легкие танки
// -----
A20 a20 = new A20("A-20", 160, 10000, 1000, 50);

MT25 mt25 = new MT25("MT-25", 200, 30000, 10000, 75);

LTTB lttb = new LTTB("ЛТТБ", 310, 80000, 40000, 100);
LTG ltg = new LTG("ЛТГ", 310, 80000, 40000, 100);
// -----

int overallDamage = 0;
int overallShot = 0;
int winDamage = 8000;
private void button1_Click(object sender, EventArgs e)
{
    textBox1.Text = "Exp: " + Exp.ToString() + "      RUB: " + Money.ToString();
    button2.Enabled = true;
    button1.Enabled = false;
    MessageBox.Show($"Для победы нужно нанести {winDamage} урона");
}

private void button2_Click(object sender, EventArgs e)
{
    button3.Enabled = true;
    button4.Enabled = true;
    button5.Enabled = true;
    textBox2.Text += $"Исследован танк {bt7.TankName}\r\n";

    Random random = new Random();
    int randomNumber = random.Next(1, 4);
    textBox2.Text += $"Танк {bt7.TankName} нанес " + randomNumber * bt7.TankDamage
+ " единиц урона\r\n";
    overallDamage += randomNumber * bt7.TankDamage;
    textBox3.Text = "Overall = " + overallDamage;
    overallShot += randomNumber;

    button2.Enabled = false;
}

private void button3_Click(object sender, EventArgs e)
{
    int exp = a20.ExploreTank();
    int money = a20.BuyTank();
    Exp = Exp - exp;
    Money = Money - money;
    if (Exp < 0 || Money < 0)
    {
        MessageBox.Show("Недостаточно ресурсов");
        Exp = Exp + exp;
        Money = Money + money;
    }
    else
    {

```

```

        a20.InstallOptics();
        textBox1.Text = "Exp: " + Exp.ToString() + "      RUB: " +
Money.ToString();
        textBox2.Text += $"\"r\nИсследован танк {a20.TankName}. Потрачено опыта: "
+ exp + ". Потрачено рублей: " + money + "\"r\n";
        button6.Enabled = true;

        Random random = new Random();
        int randomNumber = random.Next(1, 6);
        textBox2.Text += $"Танк {a20.TankName} нанес " + randomNumber *
a20.TankDamage + " единиц урона\"r\n";
        overallDamage += randomNumber * a20.TankDamage;
        textBox3.Text = "Overall = " + overallDamage;
        overallShot += randomNumber;
    }
    button3.Enabled = false;
}

private void button4_Click(object sender, EventArgs e)
{
    int exp = t34.ExploreTank();
    int money = t34.BuyTank();
    Exp = Exp - exp;
    Money = Money - money;
    if (Exp < 0 || Money < 0)
    {
        MessageBox.Show("Недостаточно ресурсов");
        Exp = Exp + exp;
        Money = Money + money;
    }
    else
    {
        t34.InstallSupportEquipment();
        textBox1.Text = "Exp: " + Exp.ToString() + "      RUB: " +
Money.ToString();
        textBox2.Text += $"\"r\nИсследован танк {t34.TankName}. Потрачено опыта: "
+ exp + ". Потрачено рублей: " + money + "\"r\n";
        button7.Enabled = true;
        button8.Enabled = true;

        Random random = new Random();
        int randomNumber = random.Next(1, 6);
        textBox2.Text += $"Танк {t34.TankName} нанес " + randomNumber *
t34.TankDamage + " единиц урона\"r\n";
        overallDamage += randomNumber * t34.TankDamage;
        textBox3.Text = "Overall = " + overallDamage;
        overallShot += randomNumber;
    }
    button4.Enabled = false;
}

private void button5_Click(object sender, EventArgs e)
{
    int exp = kv1.ExploreTank();

```

```

int money = kv1.BuyTank();
Exp = Exp - exp;
Money = Money - money;
if (Exp < 0 || Money < 0)
{
    MessageBox.Show("Недостаточно ресурсов");
    Exp = Exp + exp;
    Money = Money + money;
}
else
{
    kv1.InstallArmor();
    textBox1.Text = "Exp: " + Exp.ToString() + "      RUB: " +
Money.ToString();
    textBox2.Text += $"\\r\\nИсследован танк {kv1.TankName}. Потрачено опыта: "
+ exp + ". Потрачено рублей: " + money + "\\r\\n";
    button9.Enabled = true;
    button10.Enabled = true;

    Random random = new Random();
    int randomNumber = random.Next(1, 5);
    textBox2.Text += $"Танк {kv1.TankName} нанес " + randomNumber *
kv1.TankDamage + " единиц урона\\r\\n";
    overallDamage += randomNumber * kv1.TankDamage;
    textBox3.Text = "Overall = " + overallDamage;
    overallShot += randomNumber;
}
button5.Enabled = false;
}

private void button6_Click(object sender, EventArgs e)
{
    int exp = mt25.ExploreTank();
    int money = mt25.BuyTank();
    Exp = Exp - exp;
    Money = Money - money;
    if (Exp < 0 || Money < 0)
    {
        MessageBox.Show("Недостаточно ресурсов");
        Exp = Exp + exp;
        Money = Money + money;
    }
    else
    {
        mt25.InstallOptics();
        textBox1.Text = "Exp: " + Exp.ToString() + "      RUB: " +
Money.ToString();
        textBox2.Text += $"\\r\\nИсследован танк {mt25.TankName}. Потрачено опыта: "
+ exp + ". Потрачено рублей: " + money + "\\r\\n";
        button11.Enabled = true;
        button12.Enabled = true;

        Random random = new Random();
        int randomNumber = random.Next(1, 7);

```

```

        textBox2.Text += $"Танк {mt25.TankName} нанес " + randomNumber *
mt25.TankDamage + " единиц урона\r\n";
        overallDamage += randomNumber * mt25.TankDamage;
        textBox3.Text = "Overall = " + overallDamage;
        overallShot += randomNumber;

        randomNumber = random.Next(1, 3);
        if (randomNumber == 1)
        {
            int lightShot = mt25.MakeLightUpShot();
            textBox2.Text += $"Танк {mt25.TankName} нанес " + lightShot +
" единиц урона с помощью выстрела засвета\r\n";
            overallDamage += lightShot;
            textBox3.Text = "Overall = " + overallDamage;
            overallShot++;
        }
    }
    button6.Enabled = false;
}

private void button7_Click(object sender, EventArgs e)
{
    int exp = t3485.ExploreTank();
    int money = t3485.BuyTank();
    Exp = Exp - exp;
    Money = Money - money;
    if (Exp < 0 || Money < 0)
    {
        MessageBox.Show("Недостаточно ресурсов");
        Exp = Exp + exp;
        Money = Money + money;
    }
    else
    {
        t3485.InstallSupportEquipment();
        textBox1.Text = "Exp: " + Exp.ToString() + "      RUB: " +
Money.ToString();
        textBox2.Text += $" \r\nИсследован танк {t3485.TankName}. Потрачено опыта:
" + exp + ". Потрачено рублей: " + money + " \r\n";

        Random random = new Random();
        int randomNumber = random.Next(1, 6);
        textBox2.Text += $"Танк {t3485.TankName} нанес " + randomNumber *
t3485.TankDamage + " единиц урона\r\n";
        overallDamage += randomNumber * t3485.TankDamage;
        textBox3.Text = "Overall = " + overallDamage;
        overallShot += randomNumber;

        randomNumber = random.Next(1, 3);
        if (randomNumber == 1)
        {
            int supportShot = t3485.MakeSupportShot();
            textBox2.Text += $"Танк {t3485.TankName} нанес " + supportShot +
" единиц урона с помощью выстрела поддержки\r\n";

```

```

        overallDamage += supportShot;
        textBox3.Text = "Overall = " + overallDamage;
        overallShot++;
    }
}
button7.Enabled = false;
}

private void button8_Click(object sender, EventArgs e)
{
    int exp = t43.ExploreTank();
    int money = t43.BuyTank();
    Exp = Exp - exp;
    Money = Money - money;
    if (Exp < 0 || Money < 0)
    {
        MessageBox.Show("Недостаточно ресурсов");
        Exp = Exp + exp;
        Money = Money + money;
    }
    else
    {
        t43.InstallSupportEquipment();
        textBox1.Text = "Exp: " + Exp.ToString() + "      RUB: " +
Money.ToString();
        textBox2.Text += $"\\r\\nИсследован танк {t43.TankName}. Потрачено опыта: "
+ exp + ". Потрачено рублей: " + money + "\\r\\n";

        Random random = new Random();
        int randomNumber = random.Next(1, 6);
        textBox2.Text += $"Танк {t43.TankName} нанес " + randomNumber *
t43.TankDamage + " единиц урона\\r\\n";
        overallDamage += randomNumber * t43.TankDamage;
        textBox3.Text = "Overall = " + overallDamage;
        overallShot += randomNumber;

        randomNumber = random.Next(1, 4);
        if (randomNumber == 1)
        {
            int supportShot = t43.MakeSupportShot();
            textBox2.Text += $"Танк {t43.TankName} нанес " + supportShot +
" единиц урона с помощью выстрела поддержки\\r\\n";
            overallDamage += supportShot;
            textBox3.Text = "Overall = " + overallDamage;
            overallShot++;
        }
    }
    button8.Enabled = false;
}

private void button9_Click(object sender, EventArgs e)
{
    int exp = kv2.ExploreTank();
    int money = kv2.BuyTank();

```

```

Exp = Exp - exp;
Money = Money - money;
if (Exp < 0 || Money < 0)
{
    MessageBox.Show("Недостаточно ресурсов");
    Exp = Exp + exp;
    Money = Money + money;
}
else
{
    kv2.InstallArmor();
    textBox1.Text = "Exp: " + Exp.ToString() + "      RUB: " +
Money.ToString();
    textBox2.Text += $"\\r\\nИсследован танк {kv2.TankName}. Потрачено опыта: "
+ exp + ". Потрачено рублей: " + money + "\\r\\n";
    textBox2.Text += kv2.GetCollect() + "\\r\\n";

    Random random = new Random();
    int randomNumber = random.Next(1, 3);
    textBox2.Text += $"Танк {kv2.TankName} нанес " + randomNumber *
kv2.TankDamage + " единиц урона\\r\\n";
    overallDamage += randomNumber * kv2.TankDamage;
    textBox3.Text = "Overall = " + overallDamage;
    overallShot += randomNumber;

    randomNumber = random.Next(1, 5);
    if (randomNumber == 1)
    {
        int powerfulShot = kv2.MakePowerfulShot();
        textBox2.Text += $"Танк {kv2.TankName} нанес " + powerfulShot +
" единиц урона с помощью мощного выстрела\\r\\n";
        overallDamage += powerfulShot;
        textBox3.Text = "Overall = " + overallDamage;
        overallShot++;
    }
}
button9.Enabled = false;
}

private void button10_Click(object sender, EventArgs e)
{
    int exp = kv1s.ExploreTank();
    int money = kv1s.BuyTank();
    Exp = Exp - exp;
    Money = Money - money;
    if (Exp < 0 || Money < 0)
    {
        MessageBox.Show("Недостаточно ресурсов");
        Exp = Exp + exp;
        Money = Money + money;
    }
    else
    {
        kv1s.InstallArmor();

```

```

        textBox1.Text = "Exp: " + Exp.ToString() + "      RUB: " +
Money.ToString();
        textBox2.Text += $"\"r\nИсследован танк {kv1s.TankName}. Потрачено опыта: "
+ exp + "\". Потрачено рублей: " + money + "\"r\n";

        Random random = new Random();
        int randomNumber = random.Next(1, 4);
        textBox2.Text += $"Танк {kv1s.TankName} нанес " + randomNumber *
kv1s.TankDamage + " единиц урона\"r\n";
        overallDamage += randomNumber * kv1s.TankDamage;
        textBox3.Text = "Overall = " + overallDamage;
        button13.Enabled = true;
        button14.Enabled = true;
        overallShot += randomNumber;

        randomNumber = random.Next(1, 4);
        if (randomNumber == 1)
        {
            int powerfulShot = kv1s.MakePowerfulShot();
            textBox2.Text += $"Танк {kv1s.TankName} нанес " + powerfulShot +
" единиц урона с помощью мощного выстрела\"r\n";
            overallDamage += powerfulShot;
            textBox3.Text = "Overall = " + overallDamage;
            overallShot++;
        }
    }
    button10.Enabled = false;
}

private void button11_Click(object sender, EventArgs e)
{
    int exp = lttb.ExploreTank();
    int money = lttb.BuyTank();
    Exp = Exp - exp;
    Money = Money - money;
    if (Exp < 0 || Money < 0)
    {
        MessageBox.Show("Недостаточно ресурсов");
        Exp = Exp + exp;
        Money = Money + money;
    }
    else
    {
        lttb.InstallOptics();
        textBox1.Text = "Exp: " + Exp.ToString() + "      RUB: " +
Money.ToString();
        textBox2.Text += $"\"r\nИсследован танк {lttb.TankName}. Потрачено опыта: "
+ exp + "\". Потрачено рублей: " + money + "\"r\n";

        Random random = new Random();
        int randomNumber = random.Next(1, 8);
        textBox2.Text += $"Танк {lttb.TankName} нанес " + randomNumber *
lttb.TankDamage + " единиц урона\"r\n";
        overallDamage += randomNumber * lttb.TankDamage;

```

```

        textBox3.Text = "Overall = " + overallDamage;
        overallShot += randomNumber;

        randomNumber = random.Next(1, 3);
        if (randomNumber == 1)
        {
            int lightShot = lttb.MakeLightUpShot();
            textBox2.Text += $"Танк {lttb.TankName} нанес " + lightShot +
" единиц урона с помощью выстрела засвета\r\n";
            overallDamage += lightShot;
            textBox3.Text = "Overall = " + overallDamage;
            overallShot++;
        }

        randomNumber = random.Next(1, 3);
        int extrLightShot = lttb.MakeLightUpShot(10 * randomNumber);
        textBox2.Text += $"Танк {lttb.TankName} нанес " + extrLightShot +
" единиц урона с помощью дополнительного выстрела засвета\r\n";
        overallDamage += extrLightShot;
        textBox3.Text = "Overall = " + overallDamage;
        overallShot++;
    }
    button11.Enabled = false;
}

private void button12_Click(object sender, EventArgs e)
{
    int exp = ltg.ExploreTank();
    int money = ltg.BuyTank();
    Exp = Exp - exp;
    Money = Money - money;
    if (Exp < 0 || Money < 0)
    {
        MessageBox.Show("Недостаточно ресурсов");
        Exp = Exp + exp;
        Money = Money + money;
    }
    else
    {
        ltg.InstallOptics();
        textBox1.Text = "Exp: " + Exp.ToString() + " RUB: " +
Money.ToString();
        textBox2.Text += $"Исследован танк {ltg.TankName}. Потрачено опыта: "
+ exp + ". Потрачено рублей: " + money + "\r\n";

        Random random = new Random();
        int randomNumber = random.Next(1, 8);
        textBox2.Text += $"Танк {ltg.TankName} нанес " + randomNumber *
ltg.TankDamage + " единиц урона\r\n";
        overallDamage += randomNumber * ltg.TankDamage;
        textBox3.Text = "Overall = " + overallDamage;
        overallShot += randomNumber;

        randomNumber = random.Next(1, 3);

```



```

        if (randomNumber == 1)
        {
            int lightShot = ltg.MakeLightUpShot();
            textBox2.Text += $"Танк {ltg.TankName} нанес " + lightShot + " единиц
урона с помощью выстрела засвета\r\n";
            overallDamage += lightShot;
            textBox3.Text = "Overall = " + overallDamage;
            overallShot++;
        }

        randomNumber = random.Next(1, 5);
        int randomNumber1 = random.Next(0, 3);
        int extrLightShot = ltg.MakeLightUpShot(10 * randomNumber, 10 *
randomNumber1);
        textBox2.Text += $"Танк {ltg.TankName} нанес " + extrLightShot + " единиц
урона с помощью дополнительного выстрела засвета\r\n";
        overallDamage += extrLightShot;
        textBox3.Text = "Overall = " + overallDamage;
        overallShot++;
    }
    button12.Enabled = false;
}

private void button13_Click(object sender, EventArgs e)
{
    int exp = kv3.ExploreTank();
    int money = kv3.BuyTank();
    Exp = Exp - exp;
    Money = Money - money;
    if (Exp < 0 || Money < 0)
    {
        MessageBox.Show("Недостаточно ресурсов");
        Exp = Exp + exp;
        Money = Money + money;
    }
    else
    {
        kv3.InstallArmor();
        textBox1.Text = "Exp: " + Exp.ToString() + " RUB: " +
Money.ToString();
        textBox2.Text += $" \r\nИсследован танк {kv3.TankName}. Потрачено опыта: "
+ exp + ". Потрачено рублей: " + money + " \r\n";

        Random random = new Random();
        int randomNumber = random.Next(1, 5);
        textBox2.Text += $"Танк {kv3.TankName} нанес " + randomNumber *
kv3.TankDamage + " единиц урона\r\n";
        overallDamage += randomNumber * kv3.TankDamage;
        textBox3.Text = "Overall = " + overallDamage;
        button13.Enabled = true;
        button14.Enabled = true;
        overallShot += randomNumber;

        randomNumber = random.Next(1, 4);

```

```

        if (randomNumber == 1)
        {
            int powerfulShot = kv3.MakePowerfulShot();
            textBox2.Text += $"Танк {kv3.TankName} нанес " + powerfulShot +
" единиц урона с помощью мощного выстрела\r\n";
            overallDamage += powerfulShot;
            textBox3.Text = "Overall = " + overallDamage;
            overallShot++;
        }

        randomNumber = random.Next(0, 2);
        int extrShot = kv3.MakePowerfulShot(randomNumber);
        textBox2.Text += $"Танк {kv3.TankName} нанес " + extrShot + " единиц
урона с помощью дополнительного мощного выстрела\r\n";
        overallDamage += extrShot;
        textBox3.Text = "Overall = " + overallDamage;
        overallShot++;
    }
    button13.Enabled = false;
}

private void button14_Click(object sender, EventArgs e)
{
    int exp = iS.ExploreTank();
    int money = iS.BuyTank();
    Exp = Exp - exp;
    Money = Money - money;
    if (Exp < 0 || Money < 0)
    {
        MessageBox.Show("Недостаточно ресурсов");
        Exp = Exp + exp;
        Money = Money + money;
    }
    else
    {
        iS.InstallArmor();
        textBox1.Text = "Exp: " + Exp.ToString() + " RUB: " +
Money.ToString();
        textBox2.Text += $" \r\nИсследован танк {iS.TankName}. Потрачено опыта: " +
exp + ". Потрачено рублей: " + money + " \r\n";

        Random random = new Random();
        int randomNumber = random.Next(1, 5);
        textBox2.Text += $"Танк {iS.TankName} нанес " + randomNumber *
iS.TankDamage + " единиц урона\r\n";
        overallDamage += randomNumber * iS.TankDamage;
        textBox3.Text = "Overall = " + overallDamage;
        button13.Enabled = true;
        button14.Enabled = true;
        overallShot += randomNumber;

        randomNumber = random.Next(1, 4);
        if (randomNumber == 1)
        {

```

```

        int powerfulShot = iS.MakePowerfulShot();
        textBox2.Text += $"Танк {iS.TankName} нанес " + powerfulShot +
" единиц урона с помощью мощного выстрела\r\n";
        overallDamage += powerfulShot;
        textBox3.Text = "Overall = " + overallDamage;
        overallShot++;
    }

    randomNumber = random.Next(0, 3);
    int randomNumber1 = random.Next(0, 2);
    int extrShot = iS.MakePowerfulShot(randomNumber, randomNumber1);
    textBox2.Text += $"Танк {iS.TankName} нанес " + extrShot + " единиц урона
с помощью дополнительного мощного выстрела\r\n";
    overallDamage += extrShot;
    textBox3.Text = "Overall = " + overallDamage;
    overallShot++;
}
button14.Enabled = false;
}

private void button15_Click(object sender, EventArgs e)
{
    var logger = Log.getLog();
    if (overallDamage > winDamage)
    {
        MessageBox.Show($"Ты нанес {overallDamage} урона и победил!
Поздравляем!");
        logger.AddLog("you win");
    }
    else
    {
        MessageBox.Show($"Ты нанес {overallDamage} урона, но этого не хватило для
победы!");
        logger.AddLog("you lose");
    }

    logger.AddLog("program finished\n");
    Close();
}

private void button16_Click(object sender, EventArgs e)
{
    try
    {
        // Попытка деления на ноль
        double result = DivideNumbers(overallDamage, overallShot);
        MessageBox.Show($"Средний урон за выстрел: {Math.Round(result)}");
    }
    catch (MyDivideByZeroException ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

```

    }
    static double DivideNumbers(int numerator, int denominator)
    {
        if (denominator == 0)
        {
            throw new MyDivideByZeroException(); // Генерация переопределенного
исключения
        }
        return (double)numerator / denominator;
    }

    public delegate void MyDelegate();
    private void button17_Click(object sender, EventArgs e)
    {
        MyDelegate multiDelegate = null;
        multiDelegate += bt7.Shoot;
        multiDelegate += a20.Shoot;
        multiDelegate += mt25.Shoot;
        multiDelegate += ltg.Shoot;
        multiDelegate += lttb.Shoot;

        multiDelegate += kv1.Shoot;
        multiDelegate += kv1s.Shoot;
        multiDelegate += kv2.Shoot;
        multiDelegate += kv3.Shoot;
        multiDelegate += iS.Shoot;

        multiDelegate += t34.Shoot;
        multiDelegate += t3485.Shoot;
        multiDelegate += t43.Shoot;

        // Вызываем делегат, что вызовет все добавленные методы
        multiDelegate?.Invoke();
        MessageBox.Show("Делегат вызван. Данный про базовый урон танков сохранены в
лог.");
    }
}

```