Практическая работа 11.

Принципы ООП. Абстракция и Полиморфизм

Задание 1

Создайте класс Animal с **абстрактным** методом Sound. Создайте производные от класса Animal классы Cat, Dog, Duck. Реализуйте метод Sound(). В Main() создайте по одному объекту каждого производного класса и вызовите их методы Sound().

Задание 2

Создайте класс Bank с **виртуальным** методом Use(), который печатает сообщение “Спасибо что пользуетесь нашими банком!”

Создайте производные от класса Bank классы SberBank, AlphaBank, VtbBank, которые будут переопределять метод Use(), выводя сообщение “Спасибо что пользуетесь услугами {название банка}!”

Создайте массив из 10 случайных банкоматов. Вызовите у всех Use() в отдельном цикле.

Задание 3

Создайте классы:

- ResourceStorage
    - Поля:

        - private _money = 500;
        - private _wood = 500;
        - private _iron = 500;
    - Свойства:
        - Money {get;set;} - ограничивает значение от 0 до 500.
        - Wood {get;set;} - ограничивает значение от 0 до 500;
        - Iron {get;set;} - ограничивает значение от 0 до 500;
    - Метод:
        - DisplayResource() - печатает в консоль информацию о ресурсах.
        - bool TryGetWood(int amount) - При попытке вычесть больше, чем имеется выдает специально сообщение и возвращает false, не изменяя значение ресурса. Если ресурсов хватает, то возвращает true изменяя значение ресурса.
        - bool TryGetMoney(int amount) - При попытке вычесть больше, чем имеется выдает специально сообщение и возвращает false, не изменяя значение ресурса. Если ресурсов хватает, то возвращает true изменяя значение ресурса.
        - bool TryGetIron(int amount) - При попытке вычесть больше, чем имеется выдает специально сообщение и возвращает false, не

изменяя значение ресурса. Если ресурсов хватает, то возвращает true изменяя значение ресурса.

- AbstractFactory
    - Хранит в поле ссылку на ResourceStorage и инкапсулирует в свойство только для чтения.
    - В конструкторе принимает ссылку на объект ResourceStorage.
    - Абстрактный метод Create();
- ArcherFactory : AbstractFactory
    - Реализует Create, отнимая из ResourceStorage 50 денег, 50 дерева.
    - Если ресурсов хватает, то вывести сообщение “Лучник был создан”.
- InfantryFactory : AbstractFactory
    - Реализует Create, отнимая из ResourceStorage 50 денег, 10 дерева 25 железа.
    - Если ресурсов хватает, то вывести сообщение “Пехотинец был создан”.
- KnightFactory : AbstractFactory
    - Реализует Create, отнимая из ResourceStorage 150 денег, 0 дерева, 100 железа.
    - Если ресурсов хватает, то вывести сообщение “Рыцарь был создан”.

Создайте меню, с помощью которого можно выбрать какую из Factory использовать. Зациклите меню и сделайте выход отдельным пунктом. Отдельный пункт должен вызвать DisplayResource().

Контрольные вопросы:

1. Что такое абстракция в объектно-ориентированном программировании?
2. Что такое полиморфизм в объектно-ориентированном программировании?
3. Что такое виртуальные методы/свойства?
4. Что такое абстрактный класс и какая его главная особенность?
5. Что такое абстрактные методы/свойства?
6. Чем отличаются абстрактные методы от виртуальных методов?
7. Каковы преимущества использования абстракции и полиморфизма?
8. Для чего используется модификатор sealed для виртуальных методов/свойств?
9. Для чего используется модификатор override?