

## ПЗ\_6 Создание классов, реализация наследования.

**Цель:** закрепить знания по работе с классами в контексте структуры класса, иерархии наследования

**Задание:**

В соответствии с вариантом реализовать приложение:

1. Текстовый / графический интерфейс на ваш выбор.
2. Должны быть отработаны различные варианты создания объектов классов при наличии перегруженных конструкторов.

Критерии оценивания	Количество начисляемых баллов:
Создан родительский класс	1
Определены свойства родительского класса	1
Разработаны статические поля/методы родительского класса	1
Реализован унаследованный класс	1
Переопределена функциональность унаследованного класса	1

### 1 вариант

Создать класс FractionalNumber описывающий сущность дробного числа:

Член класса	название	особенности
поле	sign	- +
поле	dividend	Целое число
поле	divider	Целое число
метод	GetNumber()	выводит значение дроби в формате $\pm x/n$

**Добавить свойства**, которые проверяют числа на корректность (делимое строго целое число, делитель строго целое число  $\neq 0$ )

**Добавить статические поля**, которые содержат количество положительных дробных чисел и отрицательных по отдельности

Создать класс NegativeFN, унаследованный от FractionalNumber.

- **Добавить поле** capacity (разрядность). Поле знака сделать недоступным для изменения. Он всегда отрицателен.
- **Переопределить метод** GetNumber так, чтоб он выводил дробь в соответствии с разрядностью. Например:

*Capacity = 4*

*Результат работы GetNumber для дроби 8/10: -0008/0010*

## 2 вариант

Создать класс Address описывающий почтовый адрес как сущность в приложении.

Член класса	название	особенности
поле	country	Строка
поле	city	строка
поле	street	строка
поле	house	Число
поле	flat	Число
метод	GetAddress()	Выводит адрес единой строкой

**Добавить свойства**, которые проверяют на пустые вводимые значения строковых полей, значения дома и квартиры в диапазоне 1..100

**Добавить статические поля**, которые содержат количество объектов класса и количество четных домов

Создать класс RemoteAddress, унаследованный от Address

- Добавить свойство Reason (причина удаления адреса из базы).
- Переопределить метод GetAddress() так чтоб он выводил значения всех доступных полей класса RemoteAddress

## 3 вариант

Создать класс Package описывающий посылку как сущность в приложении.

Член класса	название	особенности
поле	nameDest	Имя отправителя
поле	nameSor	Имя получателя
поле	address	Адрес назначения
Поле	receiptDate	Дата отправки (DateTime)
поле	type	Тип посылки
метод	GetPackageInfo()	выводит описание текущего объекта

**Добавить свойства**, которые проверяют на пустые имена отправителя, получателя и адреса

**Добавить свойство** веса посылки (диапазон от 0,01 .. 10 кг)

**Добавить статические поля**, которые считают общий вес всех посылок и количество посылок

Создать класс UndeliveredPackage, унаследованный от Package.

- Добавить свойство Reason (причина недоставки посылки).
- Переопределить метод GetPackageInfo() так чтоб он выводил значения всех доступных полей класса NotDeliveryPackage

#### 4 вариант

Создать класс BookCard описывающий карточку книги в библиотеке как сущность в приложении

Член класса	название	особенности
поле	ID	Уникальный номер
поле	author	автор
поле	title	название
поле	year	Год издания экземпляра
поле	condition	Состояние книги
метод	GetBookInfo()	выводит описание текущего объекта

**Добавить свойства**, содержащие проверку на непустые имя автора, название

**Добавить свойства**, проверяющие корректный год издания (не раньше 1990 г)

**Создать перечисление** для состояния книги (3 позиции: на списание, хорошее, новая)

**Создать статические поля**, которые считают отдельно количество книг на списание, в хорошем состоянии и в новом состоянии

Создать класс ScientificBookCard (карточка научной литературы), унаследованной от BoolCard

- Добавить свойство Direction (направление – химия, медицина, и т.д.) и свойство ReceiptDate (добавить проверку на свежесть литературы – не старше 5 лет)
- Переопределить метод для вывода обновленной информации

#### 5 вариант

Создать класс Atom описывающий характеристики химического вещества как сущность в приложении

Член класса	название	особенности
поле	name	Имя хим элемента
поле	countElectrons	Количество электронов
поле	mass	Масса атома
поле	valence	валентность
метод	GetInfo()	выводит описание текущего объекта

**Добавить свойство**, проверяющее вводимые данные на непустое имя элемента

**Добавить свойство**, гарантирующее что количество электронов >0

Валентность реализовать через **перечисление** (пусть будут только значения I II III IV V)

**С помощью статических полей** подсчитать количество элементов с 1 и 2 валентностью по отдельности

Создать класс RadioactiveAtom (радиоактивный атом) унаследованный от класса Atom

- Добавить свойство – период полураспада (проверка на неотрицательные значения)
- Переопределить метод для вывода обновленной информации