

## Домашнє завдання

### Обробка винятків у C#.

#### Завдання 1.

Написати клас *Employee* (Працівник), що описує працівника, містить дані про

- ім'я, ○ прізвище, ○ посаду, ○ заробітну плату
- номер договору про прийом на роботу ○ властивості доступу до полів

Визначити у класі методи

- вводу імені та прізвища,
- вводу заробітної плати.

Методи повинні викидати винятки

- при введенні у ім'я чи прізвище недопустимих символів(цифр, чи ін),
- введенні символьних даних в числові поля(заробітна плата)

Створити клас *Department*(відділ), що міститиме масив(або список *List<>*) з працівників(*Employee*) та методи

- додавання працівників,
- видалення працівника(за номером договору чи прізвищем та іменем) ○ перегляд відділу

До класу *Department* додати обробку виняткових ситуацій ○ перевищення розміру масиву працівників відділу(або деякої кількості, якщо обрано колекцію *List<>*)

- видалення даних з пустого масиву(списку), тобто видалення працівника з відділу, якщо в відділі ще немає жодного працівника)

У програмі надати користувачу можливість здійснити повторну спробу введення даних.

Можна визначити свої класи винятків `class`

`FullDepartmentException` : `ApplicationException`

```
{    public FullDepartmentException(string message = "Department is full") : base(message)
{ }
}
```

#### Завдання 2.

Створити базовий абстрактний клас *Shape*.

- Визначити у класі абстрактну властивість для читання *Area* (обчислення площі фігури) ○ Визначити у класі віртуальний метод *Print()* , вивести назву фігури(*this.GetType().Name*) та її площу(*Area*).

Створити клас *Circle*, похідний від класу *Shape*.

- Визначити у класі поле *radius*

- ○ Реалізувати у класі віртуальну властивість для читання *Area* (повернути  $\pi R^2$ ).

Створити клас *Rectangle*, похідний від класу *Shape*.

- Визначити у класі поля висота та ширина.
- Реалізувати у класі віртуальну властивість для читання *Area* (ширина \* висота).

Створити для перевірки список( *List<>* ) з фігур. Помістити у список різного типу фігури.

- Вивести інформацію про всі фігури
- Впорядкувати список фігур за зростанням площ фігур
- Зібрати фігури, що є колами у окремий список(*FindAll*)

### **Завдання 3.**

Написать класс *Money*, предназначенный для хранения денежной суммы (в гривнах и копейках). Для класса реализовать перегрузку операторов + (сложение денежных сумм), – (вычитание сумм), / (деление суммы на целое число), \* (умножение суммы на целое число), ++ (сумма увеличивается на 1 копейку), -- (сумма уменьшается на 1 копейку), <, >, ==, !=.

Класс не может содержать отрицательную сумму. В случае если при исполнении какой-либо операции получается отрицательная сумма денег, то класс генерирует исключительную ситуацию «Банкрот».

Программа должна с помощью меню продемонстрировать все возможности класса *Money*.

Обработка исключительных ситуаций производится в программе.