

Лабораторна робота № 13

Тема: Обробка винятків (exceptions) у C#.

Мета: Ознайомитися з механізмом обробки винятків у C#. Навчитися використовувати конструкції try, catch, finally.

Теоретичні відомості

Виняток (Exception) — це помилка, яка виникає під час виконання програми.

Для обробки виняткових ситуацій використовують інструкцію try. Вона дозволяє позначити блок коду, в якому відслідковуватимуться виняткові ситуації, а також забезпечує код для їх обробки. Блок try містить код, у якому відслідковуватимуться винятки. Секція catch містить одну чи кілька інструкцій catch, кожна з яких має свій блок коду для обробки винятків. Інструкції catch називають ще оброблювачами виняткових ситуацій. Блок finally містить код, який виконається при будь-яких умовах: трапиться виняток, чи ні.

Основні конструкції:

```
try
{
    // Код, де може виникнути помилка
}
catch (Exception ex)
{
    // Обробка помилки
}
finally
{
    // Виконується завжди
```

- try — блок ризикованого коду
- catch — перехоплення винятку
- finally — виконується незалежно від результату

Актуалізація опорних знань.

1. Що таке виняток?
2. Для чого використовується try-catch?
3. Яка роль finally?
4. Як створити власний виняток?
5. Чим відрізняється Parse() від TryParse()?
1. Чим відрізняється throw ex; від throw;?

2. Чи можна виконати блок `catch` без блоку `finally`, і навпаки?
3. Що таке `InnerException` і для чого він потрібен?
4. Яка різниця між перевіркою через `if` та обробкою через `try-catch` з точки зору продуктивності?

Зміст роботи.

Завдання 1.

Написати програму, яка: зчитує два числа з консолі; виконує ділення; обробляє винятки: ділення на нуль (`DivideByZeroException`); неправильний формат введення (`FormatException`). Вивести зрозуміле повідомлення користувачу .

Завдання 2.

Написати програму, яка: Запрошує у користувача число. Перетворює його через `int.Parse()`. Обробляє можливі винятки.

Завдання 3.

Написати програму, яка: Створіть масив із 5 елементів. Дайте користувачу можливість ввести індекс. Обробіть `IndexOutOfRangeException`.

Завдання 4.

Реалізуйте консольний калькулятор: підтримка операцій (+, -, *, /); обробка всіх можливих помилок введення.

Завдання 5.

Створіть клас `BankAccount`: метод зняття грошей; якщо сума більша за баланс — генеруйте виняток.