Практическая работа №2

МДК 0104: Системное программирование на .NET

Тема: Условные выражения в C#

Цель работы Сформировать практические навыки применения условных операторов (`if-else`, `switch`, тернарный оператор) для управления потоком выполнения программы на C#.

Задачи

1. Закрепить синтаксис условных конструкций.
2. Научиться выбирать оптимальную конструкцию для решения конкретной задачи.
3. Развить навыки анализа условий и проектирования логики программы.
4. Приобрести опыт работы с системой контроля версий Git и платформой Gogs.

Краткая справочная информация

Основные конструкции для управления условиями в C#

1. **Оператор `if` / `else if` / `else`**: Основная конструкция для ветвления логики.

```
if (condition1) // Если условие 1 истинно (true)
{
    // Выполняется этот блок кода
}
else if (condition2) // Иначе, если условие 2 истинно
{
    // Выполняется этот блок кода
}
else // Во всех остальных случаях
{
    // Выполняется этот блок кода
}
```

2. **Тернарный условный оператор (`? :`)**: Краткая форма для простых условий, возвращающая одно из двух значений.

```
variable = (condition) ? expressionIfTrue : expressionIfFalse;
string result = (age >= 18) ? "Взрослый" : "Не взрослый";
```

3. **Оператор выбора `switch`**: Идеален для множественного ветвления, когда одна переменная или выражение сравнивается с множеством постоянных значений.

```
switch (expression)
{
    case constant1:
        // Код для constant1
        break;
    case constant2:
        // Код для constant2
        break;
    ...
    default:
        // Код, если ни один case не подошел
        break;
}
```

4. `switch` **на основе шаблонов (Pattern Matching)** (C# 7.0+): Более мощная версия `switch`, которая позволяет использовать не только константы, но и типы, условия (`when`) и т.д.

```
switch (obj)
{
    case int i when i > 0:
        Console.WriteLine($"Положительное число: {i}");
        break;
    case string s:
        Console.WriteLine($"Это строка: {s}");
        break;
    case null:
        Console.WriteLine("Объект null");
        break;
    default:
        Console.WriteLine("Неизвестный объект");
        break;
}
```

Ход работы

1. Получить вариант задания у преподавателя.
2. На локальной машине создать новое консольное приложение .NET с именем `ConditionalPractice`.
3. Реализовать логику согласно своему варианту. Код должен быть хорошо читаемым и содержать комментарии.
4. Протестировать программу на различных входных данных, включая граничные случаи (например, отрицательные числа, ноль, максимальные значения).
5. Сделать не менее 3-х скриншотов работы программы с разными входными данными.
6. Создать новый приватный репозиторий на предоставленном Git-сервере **Gogs** с названием `conditional-practice`.
7. Настроить файл `.gitignore` для проекта .NET.

8. Добавить в репозиторий исходный код проекта (файлы `.cs`), папку `images` со скриншотами и оформить файл `README.md`.
9. Предоставить ссылку на репозиторий преподавателю для проверки.

Требования к репозиторию на Gogs

Структура репозитория должна выглядеть следующим образом:

```
conditional-practice/
├── ConditionalPractice/
│   ├── Program.cs
│   └── ConditionalPractice.csproj
├── images/
│   ├── test_1.png
│   ├── test_2.png
│   └── test_3.png
├── .gitignore
└── README.md
```

Содержание `README.md`:

- Название работы и вариант.
- Цель и задачи.
- Описание задания.
- Инструкция по сборке и запуску (например, `dotnet run`).
- Описание примеров, приведенных на скриншотах.

Варианты заданий

Вариант 1: Калькулятор оценок Напишите программу, которая запрашивает у пользователя балл (от 0 до 100). Используя конструкцию `if-else if-else`, определите и выведите буквенную оценку по шкале: A (90-100), B (80-89), C (70-79), D (60-69), F (<60).

Вариант 2: Определение времени года Напишите программу, которая запрашивает у пользователя номер месяца (1-12). Используя оператор `switch`, выведите название времени года (Зима, Весна, Лето, Осень), к которому этот месяц относится.

Вариант 3: Проверка простого числа Напишите программу, которая проверяет, является ли введенное пользователем целое число простым. Используйте `if-else` для вывода результата. (Простое число - делится без остатка только на 1 и на себя).

Вариант 4: Калькулятор с операцией Напишите программу-калькулятор, которая запрашивает два числа и символ операции (`+`, `-`, `*`, `/`). Используя `switch`, выполните соответствующую арифметическую операцию и выведите результат. Учтите возможность деления на ноль с помощью `if`.

Вариант 5: Определение типа треугольника Запросите у пользователя длины трех сторон треугольника. Используя вложенные условия `if-else`, определите и выведите тип треугольника: равносторонний, равнобедренный, разносторонний или "не существует".

Вариант 6: Конвертер валют Напишите программу, которая предлагает пользователю выбрать направление конвертации: из USD в EUR или из EUR в USD. Затем запросите сумму. Используя `switch` для выбора операции и `if` для проверки положительности суммы, выполните конвертацию по фиксированному курсу.

Вариант 7: Определение високосного года Напишите программу, которая проверяет, является ли введенный год високосным. Используйте `if-else` и логические операторы для проверки условий: год високосный, если он делится на 4, но не делится на 100, либо делится на 400.

Вариант 8: Игра "Угадай число" Программа загадывает число (например, от 1 до 10). Пользователь пытается его угадать. Используя `if-else`, дайте подсказки "Больше" или "Меньше". С помощью тернарного оператора в конце выведите сообщение "Победа!" или "Попробуйте еще раз".

Вариант 9: Проверка логина и пароля Запросите у пользователя логин и пароль. Используя конструкцию `if-else` с логическими операторами, проверьте, соответствуют ли они заранее заданным значениям (например, `admin` и `password`). Выведите соответствующее сообщение.

Вариант 10: Определение дня недели Напишите программу, которая по введенному номеру дня недели (1-7) выводит его название. Реализуйте решение двумя способами: с помощью `if-else if` и с помощью `switch`. Сравните подходы в комментариях к коду.

Вариант 11: Подсчет стоимости разговора Напишите программу, которая вычисляет стоимость международного телефонного разговора. Используйте `switch` для выбора кода страны (например, США - 1, Германия - 49) и `if` для применения скидки, если разговор длился более 10 минут.

Вариант 12: Классификация возраста Запросите возраст пользователя. Используя `if-else if-else`, классифицируйте его: Младенец (0-1), Ребенок (2-12), Подросток (13-19), Взрослый (20-65), Пенсионер (65+). Для каждой категории выведите описание.

Вариант 13: Калькулятор ИМТ (Индекс Массы Тела) Рассчитайте ИМТ по введенным росту и весу. Используя `if-else if-else`, определите категорию: "Недостаточный вес", "Норма", "Избыточный вес", "Ожирение" и выведите ее вместе с числовым значением ИМТ.

Вариант 14: Проверка строки Попросите пользователя ввести строку. Используя условные операторы и свойства строк, проверьте:

1. Является ли строка пустой или `null` (`if`).
2. Содержит ли строка число (используйте `int.TryParse(...)`).
3. Длиннее ли она 5 символов (тернарный оператор для короткого сообщения). Выведите результаты проверок.

Вариант 15: Машина состояний (State Machine) Смоделируйте простой светофор. Программа поочередно выводит в консоль "Красный", "Желтый", "Зеленый" с задержкой. Используйте `switch` для выбора текущего состояния и `if` для перехода к следующему состоянию (например, по нажатию клавиши Enter). Цикл `while` будет основным, условием выхода - ввод слова "stop".

Контрольные вопросы

1. В чем основное различие между операторами `if` и `switch`? Когда целесообразно использовать каждый из них?

2. Как с помощью тернарного оператора предотвратить деление на ноль?
3. Что такое "защитное программирование" (defensive programming) и как условные операторы помогают в его реализации?
4. Объясните назначение ключевого слова `break` внутри оператора `switch`.
5. Какие преимущества дает использование `switch` с сопоставлением шаблонов (pattern matching) по сравнению с классическим `switch`?