

Лабораторная работа № 2

Тема: Управляющие конструкции (условные операторы `if - else`, `switch`, циклы `for`, `while`, `do while` + контроль ввода с помощью метода `TryParse`)

Цель работы:

Научиться использовать управляющие конструкции языка C# для создания логики программ, включая условные операторы, циклы и контроль ввода, а также взаимодействие с пользователем через меню.

Задание:

Создайте новое консольное приложение на языке C#, которое выполняет следующие действия:

1. Меню выбора действий:
 - Реализуйте меню с тремя пунктами:
 1. Отгадай ответ
 2. Об авторе (вывод на экран информации о ФИО студента и его группа)
 3. Выход
 - При выборе пункта меню 3 запрашивайте подтверждение на выход из программы (при нажатии «д» выйти из программы, при нажатии «н» – остаться в программе). Если пользователь ввел значение, отличное от «д» и «н», должно появляться сообщение об ошибке и повторный запрос ввода.
 - Рекомендуется использовать оператор **switch** для реализации меню в приложении, а для создания цикличности использовать цикл **while**.
2. Контроль ввода:
 - Реализуйте проверку на корректность ввода: используйте циклы **while** и метод **TryParse** для проверки корректности ввода и вывода сообщения об ошибке, если пользователь вводит буквы вместо цифр. Обязательно использовать подробный вариант конструкции **while + TryParse** для объяснения работы конструкции.
 - Предусмотрите корректную работу программы в случае деления на 0, взятия корня из отрицательного числа и другие возможные исключительные ситуации (определить область допустимых значений для a и b).
3. Игровая логика "Отгадай ответ":
 - Перенесите логику вычислений из лабораторной работы № 1.
 - Запросите у пользователя ввести значения для a и b и вычислите результат математической функции. Обязательно должен быть контроль ввода, описанный во втором пункте.
 - Запросите у пользователя ввести предполагаемый ответ математической функции при заданных a и b . Обязательно должен быть контроль ввода, описанный во втором пункте.

- Сравните ответ пользователя с вычисленным результатом и сообщите пользователю результат (ответ верный, ответ неверный). Для выполнения сравнения округлите ответ до второго знака после запятой.
 - Дайте пользователю 3 попытки ввода ответа. Если пользователь ввёл правильный ответ, сообщите ему о выигрыше и выйдите в меню. При неправильном вводе ответа более трех раз сообщите пользователю о проигрыше, выведите на экран правильный ответ и выйдите в меню. После каждого ввода ответа сообщайте пользователю, сколько попыток ввода у него осталось.
4. Ограничения:
- Запрещено использование операторов **break**, **goto**, **continue** для выхода из циклов.
5. Дополнительное задание (его делать **необязательно**):
- Добавьте в меню пункт "Простой математический тест", где программа задает пользователю 3 простых математических вопроса. После каждого вопроса программа будет сообщать, верен ли ответ, и в конце покажет количество правильных ответов.
 - Генерация вопросов. Программа случайным образом выбирает одно из трех математических действий: сложение, вычитание или умножение. Например, "Сколько будет $3 + 2$?" или "Сколько будет $7 * 5$?".
 - Контроль ввода. Ввод ответа пользователя проверяется с использованием **TryParse**. Если ввод некорректен, программа выводит сообщение об ошибке и запрашивает ввод снова.
 - Подсчет правильных ответов. Программа подсчитывает количество правильных ответов пользователя из трех возможных.
 - Отображение результата. После того как пользователь ответит на все 3 вопроса, программа выводит количество правильных ответов и предлагает вернуться в главное меню.
 - Используйте класс **Random** для генерации чисел и выбора математической операции.

Теоретические вопросы:

1. Напишите пример конструкции **if - else**. Является ли блок **else** обязательным в этой конструкции? В каких случаях выполняется код в блоке **if**, а в каком случае код в блоке **else**?
2. Напишите пример конструкции **if - else if - else**. Чем эта команда отличается от конструкции **if - else**? Является ли блок **else** обязательным в этой конструкции? Сколько блоков **else if** может быть в этой конструкции?
3. Что такое область видимости? В какой момент в C# уничтожается переменная? Можно ли в одной области видимости создать 2 переменные разных типов с одинаковыми именами? Можно ли в разных областях видимости создать переменные с одинаковыми именами? Если да, то в каких случаях?
4. Напишите пример конструкции "тернарный оператор ?". Покажите, как бы выглядел этот код, если бы он был записан с помощью конструкции **if - else**. В каких случаях можно использовать тернарный оператор?

5. Напишите пример конструкции **switch**. С какими типами данных работает конструкция **switch**? Является ли блок **default** обязательным? За что отвечает инструкция **break**? В каких случаях стоит применять конструкцию **switch**, а в каких **if - else if - else**?
6. Для чего и как работает команда **TryParse**? В чем отличие команд **Parse** и **TryParse**? Привести пример её использования.
7. Конструкция **try - catch - finally**. Для чего предназначена эта конструкция и привести пример её использования. Является ли обязательным блок **catch**? Является ли обязательным блок **finally**?
8. Опишите работу и синтаксис цикла **for**. В каких случаях используется цикл **for**? Привести пример использования цикла.
9. Для чего предназначены инструкции **break** и **continue** в цикле **for**? Привести примеры их использования.
10. Опишите работу и синтаксис цикла **while**. В каких случаях используется цикл **while**? Привести пример использования цикла **while**. Чем цикл **while** отличается от цикла **do while**?
11. Опишите работу и синтаксис цикла **do while**. В каких случаях используется цикл **do while**? Привести пример использования цикла **do while**. Чем цикл **do while** отличается от цикла **while**?
12. Приведите пример цикла, который ни разу не выполнится.
13. Приведите пример цикла, который выполнится ровно один раз.
14. Покажите пример бесконечного цикла. Какие есть способы принудительного выхода из цикла **for**, **while** и **do while**?
15. Расскажите, в чем разница между постфиксной (**a++**) и префиксной (**++a**) формами записи сложения? Решить и объяснить пример.

Практические задачи:

1. Напишите программу, которая запрашивает у пользователя число и проверяет, является ли оно положительным, отрицательным или нулем. Выведите соответствующее сообщение.
2. Напишите программу, которая принимает на вход оценку студента (число от 0 до 100) и выводит его буквенный эквивалент: A (90-100), B (80-89), C (70-79), D (60-69), F (0-59). Используйте конструкцию **if - else if - else**.
3. Напишите программу, которая запрашивает у пользователя название дня недели (например, "понедельник") и выводит его порядковый номер (например, 1 для понедельника). Используйте конструкцию **switch**.
4. Напишите программу, которая выводит все четные числа от 1 до 20 с использованием цикла **for**.
5. Напишите программу, которая запрашивает у пользователя число и выводит таблицу умножения для этого числа от 1 до 10. Используйте цикл **while**.
6. Напишите программу, которая запрашивает у пользователя число и считает сумму всех чисел от 1 до этого числа включительно. Используйте цикл **do-while**.
7. Напишите программу, которая запрашивает у пользователя два числа и выводит их сумму. Используйте **TryParse** для проверки корректности ввода.
8. Напишите программу, которая запрашивает у пользователя возраст и проверяет, является ли он допустимым числом. Если пользователь вводит

некорректные данные (например, буквы), программа должна запросить ввод повторно.

9. Напишите программу, которая запрашивает у пользователя длину и ширину прямоугольника и выводит его площадь. Проверьте корректность ввода с помощью **TryParse**.
10. Напишите программу, которая запрашивает у пользователя два числа и выводит большее из них. Реализуйте решение **с использованием тернарного оператора**.
11. Напишите программу, которая запрашивает у пользователя оценку (число от 0 до 100) и выводит, сдал ли он экзамен (60 и выше – сдал, меньше 60 – не сдал). **Используйте тернарный оператор**.
12. Напишите программу, в которой переменная объявлена внутри цикла. Попробуйте обратиться к этой переменной вне цикла и объясните, почему это приводит к ошибке.
13. Напишите программу, в которой одна и та же переменная используется в разных методах. Объясните, как область видимости влияет на работу программы.
14. Напишите программу, которая запрашивает у пользователя два числа и делит одно на другое. Обработайте возможную ошибку деления на ноль с использованием конструкции **try - catch**.
15. Напишите программу, которая запрашивает у пользователя числовое значение и преобразует его в **int** с помощью **Parse**. Обработайте возможное исключение, если ввод некорректен, используя **try - catch - finally**.

Дополнительные материалы для подготовки:

- <https://metanit.com/sharp/tutorial/2.24.php>
- <https://metanit.com/sharp/tutorial/2.5.php>
- <https://metanit.com/sharp/tutorial/2.6.php>
- <https://metanit.com/sharp/tutorial/3.45.php>
- <https://metanit.com/sharp/tutorial/2.14.php>