

Практическая работа 4. Классы и объекты

Цель работы:

- Освоить принципы создания классов и объектов в C#
- Научиться определять поля, свойства, методы и конструкторы
- Применить полученные знания для решения практических задач

Ход работы:

1. Изучить теоретический материал по классам и объектам
2. Разработать консольное приложение согласно варианту задания
3. Реализовать необходимые классы с полями, свойствами, методами и конструкторами
4. Протестировать работу программы
5. Оформить отчет в репозитории

Теоретическая часть

Классы и объекты

Класс - это шаблон или чертеж для создания объектов. Он определяет:

- **Поля** - переменные, хранящие состояние объекта
- **Свойства** - механизм доступа к полям с контролем ввода
- **Методы** - функции, определяющие поведение объекта
- **Конструкторы** - специальные методы для инициализации объектов

Объект - это экземпляр класса, созданный в памяти.

Основные элементы класса:

```
public class Car
{
    // Поля (обычно private)
    private string brand;
    private int year;

    // Свойства (public с контролем доступа)
    public string Brand
    {
        get { return brand; }
        set { brand = value; }
    }

    public int Year
    {
        get { return year; }
        set
        {
```

```
        if (value > 1886 && value <= DateTime.Now.Year)
            year = value;
    }
}

// Конструктор по умолчанию
public Car()
{
    brand = "Unknown";
    year = DateTime.Now.Year;
}

// Параметризованный конструктор
public Car(string brand, int year)
{
    this.brand = brand;
    this.year = year;
}

// Методы
public void DisplayInfo()
{
    Console.WriteLine($"Автомобиль: {brand}, Год: {year}");
}

public int CalculateAge()
{
    return DateTime.Now.Year - year;
}
}
```

Практический пример

```
using System;

namespace ClassesExample
{
    public class Student
    {
        // Поля
        private string firstName;
        private string lastName;
        private int age;
        private double averageGrade;

        // Свойства
        public string FirstName
        {
            get { return firstName; }
            set
            {

```

```
        if (!string.IsNullOrEmpty(value))
            firstName = value;
    }
}

public string LastName
{
    get { return lastName; }
    set
    {
        if (!string.IsNullOrEmpty(value))
            lastName = value;
    }
}

public int Age
{
    get { return age; }
    set
    {
        if (value >= 16 && value <= 100)
            age = value;
    }
}

public double AverageGrade
{
    get { return averageGrade; }
    set
    {
        if (value >= 0 && value <= 5)
            averageGrade = value;
    }
}

// Конструкторы
public Student() { }

public Student(string firstName, string lastName, int age, double
averageGrade)
{
    FirstName = firstName;
    LastName = lastName;
    Age = age;
    AverageGrade = averageGrade;
}

// Методы
public void DisplayInfo()
{
    Console.WriteLine($"Студент: {LastName} {FirstName}");
    Console.WriteLine($"Возраст: {Age} лет");
    Console.WriteLine($"Средний балл: {AverageGrade:F2}");
}
```

```
        public string GetStatus()
        {
            if (averageGrade >= 4.5)
                return "Отличник";
            else if (averageGrade >= 3.5)
                return "Хорошист";
            else if (averageGrade >= 2.5)
                return "Удовлетворительно";
            else
                return "Неудовлетворительно";
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            // Создание объектов
            Student student1 = new Student("Иван", "Петров", 20, 4.7);
            Student student2 = new Student("Мария", "Сидорова", 19, 3.8);

            // Использование методов
            student1.DisplayInfo();
            Console.WriteLine($"Статус: {student1.GetStatus()}");
            Console.WriteLine();

            student2.DisplayInfo();
            Console.WriteLine($"Статус: {student2.GetStatus()}");
        }
    }
}
```

Варианты заданий

Вариант 1: Класс "Книга"

Создайте класс **Book** с полями: название, автор, год издания, количество страниц. Реализуйте методы для вывода информации и определения, является ли книга старой (старше 50 лет).

Вариант 2: Класс "Прямоугольник"

Создайте класс **Rectangle** с полями: длина, ширина. Реализуйте методы для вычисления площади, периметра и проверки, является ли прямоугольник квадратом.

Вариант 3: Класс "Банковский счет"

Создайте класс **BankAccount** с полями: номер счета, владелец, баланс. Реализуйте методы для пополнения, снятия средств и вывода информации о счете.

Вариант 4: Класс "Студент"

Создайте класс **Student** с полями: имя, фамилия, группа, средний балл. Реализуйте методы для вывода информации и определения стипендии (отличник - повышенная, хорошист - обычная, остальные - нет).

Вариант 5: Класс "Автомобиль"

Создайте класс **Car** с полями: марка, модель, год выпуска, пробег. Реализуйте методы для вывода информации и расчета износа (пробег / год).

Вариант 6: Класс "Товар"

Создайте класс **Product** с полями: название, цена, количество на складе. Реализуйте методы для расчета общей стоимости товара на складе и применения скидки.

Вариант 7: Класс "Круг"

Создайте класс **Circle** с полем: радиус. Реализуйте методы для вычисления площади, длины окружности и диаметра.

Вариант 8: Класс "Работник"

Создайте класс **Employee** с полями: имя, должность, оклад, стаж. Реализуйте методы для расчета премии (процент от оклада в зависимости от стажа).

Вариант 9: Класс "Телевизор"

Создайте класс **Television** с полями: марка, диагональ, разрешение, включен/выключен. Реализуйте методы для включения/выключения и изменения канала.

Вариант 10: Класс "Треугольник"

Создайте класс **Triangle** с полями: сторона А, сторона В, сторона С. Реализуйте методы для проверки существования треугольника, вычисления площади и периметра.

Вариант 11: Класс "Мобильный телефон"

Создайте класс **MobilePhone** с полями: модель, заряд батареи, баланс. Реализуйте методы для звонка, отправки SMS и зарядки телефона.

Вариант 12: Класс "Кошелек"

Создайте класс **Wallet** с полями: владелец, сумма денег, валюта. Реализуйте методы для пополнения, траты денег и конвертации в другую валюту.

Вариант 13: Класс "Дом"

Создайте класс **House** с полями: адрес, площадь, количество комнат, этаж. Реализуйте методы для расчета стоимости дома и вывода полной информации.

Вариант 14: Класс "Компьютер"

Создайте класс **Computer** с полями: процессор, оперативная память, жесткий диск, включен/выключен. Реализуйте методы для включения/выключения и проверки производительности.

Вариант 15: Класс "Задача"

Создайте класс **Task** с полями: название, описание, приоритет, выполнена/не выполнена. Реализуйте методы для отметки выполнения и изменения приоритета.

Критерии оценки

Оценка 5 (отлично):

- Полностью реализован класс согласно варианту
- Корректно работают все методы
- Грамотное использование свойств с контролем ввода
- Наличие нескольких конструкторов
- Чистый и хорошо оформленный код
- Полностью рабочая программа

Оценка 4 (хорошо):

- Реализован основной функционал класса
- Работают основные методы
- Использованы свойства вместо публичных полей
- Наличие конструкторов
- Незначительные недочеты в реализации

Оценка 3 (удовлетворительно):

- Реализована базовая структура класса
- Работают не все методы
- Использованы публичные поля вместо свойств
- Отсутствуют некоторые конструкторы
- Есть ошибки в логике программы

Оценка 2 (неудовлетворительно):

- Класс не реализован или реализован некорректно
- Программа не работает
- Грубые ошибки в понимании темы

Контрольные вопросы

1. Что такое класс и чем он отличается от объекта?
2. Какие модификаторы доступа вы знаете и для чего они используются?
3. Что такое конструктор и какие типы конструкторов существуют?
4. Чем свойство отличается от поля?
5. Как создать объект класса в C#?
6. Что такое перегрузка методов и конструкторов?

7. Для чего используется ключевое слово `this`?
 8. Какие преимущества дает использование свойств вместо публичных полей?
 9. Что такое инкапсуляция и как она реализуется в C#?
 10. Как организовать проверку входных данных в свойствах?
-

Структура репозитория для практической работы

```
PracticalWork-Classes/  
├── .gitignore  
├── README.md  
├── ClassesProject/  
│   ├── Program.cs  
│   ├── Student.cs (или другой класс по варианту)  
│   └── ClassesProject.csproj  
└── images/  
    ├── screenshot1.png  
    └── screenshot2.png
```

Содержание файла README.md для репозитория:

```
# Практическая работа № 3: Классы и объекты  
  
**Вариант:** [номер варианта]  
  
**Задание:** [текст задания по варианту]  
  
## Реализация  
  
### Код программы  
  
### Скриншоты работы программы  
  
![Результат работы программы](images/screenshot1.png)  
![Тестирование методов](images/screenshot2.png)
```