1 Kopiowanie

1.1 Operator przypisania

1.1.1 Operator przypisania zastosowany do typu wartościowego.

Kod programu `https://github.com/Piotrekk16/Kopiowanie1`:

```
int a = 6; //deklaracja i nadanie wartości a na 6
int b = a; //skopiowanie za pomocą operatora przypisania

// wyświetlenie wartości a i b po skopiowaniu
Console.WriteLine("a={0}, b={1}", a, b);

b++; //inkrementacja zmiennej b

//wyświetlenie wartości a i b po zmianie
Console.WriteLine("Po zmianie");
Console.WriteLine("a={0}, b={1}", a, b);

Console.ReadKey();
```

Na konsoli otrzymamy:

```
a=6, b=6
Po zmianie
a=6, b=7
```

1.1.2 Operator przypisania zastosowany do typu referencyjnego.

W naszym projekcie dodajemy sobie klasę `Osoba`.

```
class Osoba
{
    private int wiek; //deklaracja pola
    public Osoba() { } //konstruktor domyślny
    public Osoba(int wiek) //konstruktor parametryczny
    {
        this.wiek = wiek;
    }
```

```
        public void UstawWiek(int wiek) //metoda na
           ustawienie wieku
        {
            this.wiek = wiek;
        }
        public int PobierzWiek() //metoda na pobranie wieku
        {
            return wiek;
        }
    }
```

Kod programu w `Main`:

```
            Osoba o1 = new Osoba(30); //tworzy pierwszy
               obiekt i nadajemy mu wiek 30
            Osoba o2 = new Osoba(); //tworzymy drugi obiekt
            o2 = o1; //kopiowanie za pomocą operatora
               przypisania

            //wartości po kopiowaniu
            Console.WriteLine("Wiek pierwszej osoby {0}",
               o1.PobierzWiek());
            Console.WriteLine("Wiek drugiej osoby
               {0}",o2.PobierzWiek());

            o1.UstawWiek(31); //zmiana wieku pierwszej osoby

            //wartości po zmianie
            Console.WriteLine("Po zmianie");
            Console.WriteLine("Wiek pierwszej osoby {0}",
               o1.PobierzWiek());
            Console.WriteLine("Wiek drugiej osoby {0}",
               o2.PobierzWiek());

            //sprawdzenie referencji
            if (Object.ReferenceEquals(o1, o2))
                Console.WriteLine("Referencje odwołują się
                   do tego samego obiektu");
            else
```

```
            Console.WriteLine("Referencje nie odwołują
               się do tego samego obiektu");
```

Repozytorium na githubie: `https://github.com/Piotrekk16/Kopiowanie2`.

```
Wiek pierwszej osoby 30
Wiek drugiej osoby 30
Po zmianie
Wiek pierwszej osoby 31
Wiek drugiej osoby 31
Referencje odwołują się do tego samego obiektu
```

2 Płytka kopia

Płytką kopią używamy, kiedy klasa nie zawiera pól referencyjnych. Klasa dziedziczy po interfejsie `ICloneable` i implementuje metodę `Clone()`, w której wykorzystamy metodę `System.Object.MemberwiseClone()`.

Modyfikujemy klasę `Osoba`:

```
class Osoba : ICloneable
{
    private int wiek; //deklaracja pola
    public Osoba() { } //konstruktor domyślny
    public Osoba(int wiek) //konstruktor parametryczny
    {
        this.wiek = wiek;
    }
    public void UstawWiek(int wiek) //metoda na
       ustawienie wieku
    {
        this.wiek = wiek;
    }
    public int PobierzWiek() //metoda na pobranie wieku
    {
        return wiek;
    }

    public Object Clone() //implementacja metody Clone
    {
        return MemberwiseClone();
    }
```

Zmodyfikowany kod w `Main`:

```
Osoba o1 = new Osoba(30); //tworzy pierwszy
   obiekt i nadajemy mu wiek 30
Osoba o2 = new Osoba(); //tworzymy drugi obiekt
o2 = (Osoba)o1.Clone(); //kopiowanie za pomocą
   metody Clone(), potem rzutujemy na klasę
   Osoba

//wartości po kopiowaniu
Console.WriteLine("Wiek pierwszej osoby {0}",
   o1.PobierzWiek());
Console.WriteLine("Wiek drugiej osoby
   {0}",o2.PobierzWiek());

o1.UstawWiek(31); //zmiana wieku pierwszej osoby

//wartości po zmianie
Console.WriteLine("Po zmianie");
Console.WriteLine("Wiek pierwszej osoby {0}",
   o1.PobierzWiek());
Console.WriteLine("Wiek drugiej osoby {0}",
   o2.PobierzWiek());

//sprawdzenie referencji
if (Object.ReferenceEquals(o1, o2))
    Console.WriteLine("Referencje odwołują się
       do tego samego obiektu");
else
    Console.WriteLine("Referencje nie odwołują
       się do tego samego obiektu");
```

Kod na githubie: `https://github.com/Piotrekk16/Kopiowanie3`.

```
Wiek pierwszej osoby 30
Wiek drugiej osoby 30
Po zmianie
Wiek pierwszej osoby 31
Wiek drugiej osoby 30
Referencje nie odwołują się do tego samego obiektu
```

2.1 Płytka kopia dla klasy z polem referencyjnym

Tworzy klasę `Urodziny` na przechowywanie pola referencyjnego w klasie `Osoba`.

```
class Urodziny
{
    private string miejsceUr; //deklaracja pola
    public Urodziny() { } //konstruktor domyślny

    //konstruktor parametryczny
    public Urodziny(string miejsceUr)
    {
        this.miejsceUr = miejsceUr;
    }

    //metoda na ustawienie miejsca urodzin
    public void UstawMiejsceUr(string miejsceUr)
    {
        this.miejsceUr = miejsceUr;
    }

    //metoda na pobranie miejsca urodzenia
    public string PobierzMiejsceUr()
    {
        return miejsceUr;
    }
```

Zmodyfikowana klasa `Osoba`:

```
class Osoba : ICloneable
{
    private int wiek; //deklaracja pola
    public Urodziny ur; //pole referencyjne

    //konstruktor domyślny (uzupełniamy o tworzenie
       obiektu)
    public Osoba()
    {
        ur = new Urodziny();
```

```
        }

        //konstruktor parametryczny
        public Osoba(int wiek,string miejsce)
        {
            this.wiek = wiek;
            ur = new Urodziny();
            ur.UstawMiejsceUr(miejsce);
        }

        public void UstawWiek(int wiek) //metoda na
           ustawienie wieku
        {
            this.wiek = wiek;
        }

        public int PobierzWiek() //metoda na pobranie wieku
        {
            return wiek;
        }

        public Object Clone() //implementacja metody Clone
        {
            return MemberwiseClone();
        }

    }
```

Kod w `Main` pozostawiamy bez zmian. Projekt `https://github.com/Piotrekk16/Kopiowanie4`. Wyjście poniżej. Modyfikacja samego wieku daje pożądany "efekt kopiowania".

```
Wiek pierwszej osoby 30
Wiek drugiej osoby 30
Po zmianie
Wiek pierwszej osoby 31
Wiek drugiej osoby 30
Referencje nie odwołują się do tego samego obiektu
```

Jeśli w Main zmodyfikujemy również miejsce urodzin, nie będzie to co chcemy.

```
Osoba o1 = new Osoba(30,"Olsztyn"); //tworzy
   pierwszy obiekt i nadajemy mu wiek 30
Osoba o2 = new Osoba(); //tworzymy drugi obiekt
o2 = (Osoba)o1.Clone(); //kopiowanie za pomocą
   metody Clone(), potem rzutujemy na klasę
   Osoba

//wartości po kopiowaniu
Console.WriteLine("Wiek pierwszej osoby {0}",
   o1.PobierzWiek());
Console.WriteLine("Miejsce urodzin pierwszej
   osoby {0}", o1.ur.PobierzMiejsceUr());
Console.WriteLine("Wiek drugiej osoby
   {0}",o2.PobierzWiek());
Console.WriteLine("Miejsce urodzin drugiej
   osoby {0}", o2.ur.PobierzMiejsceUr());

o1.UstawWiek(31); //zmiana wieku pierwszej osoby
o1.ur.UstawMiejsceUr("Gdańsk"); //zmiana
   miejsca urodzin pierwszej osoby

//wartości po zmianie
Console.WriteLine("Po zmianie");
Console.WriteLine("Wiek pierwszej osoby {0}",
   o1.PobierzWiek());
Console.WriteLine("Miejsce urodzin pierwszej
   osoby {0}", o1.ur.PobierzMiejsceUr());
Console.WriteLine("Wiek drugiej osoby {0}",
   o2.PobierzWiek());
Console.WriteLine("Miejsce urodzin drugiej
   osoby {0}", o2.ur.PobierzMiejsceUr());

//sprawdzenie referencji
if (Object.ReferenceEquals(o1.ur, o2.ur))
    Console.WriteLine("Referencje odwołują się
       do tego samego obiektu");
else
    Console.WriteLine("Referencje nie odwołują
```

```
                    się do tego samego obiektu");
```

Kod na githubie: `https://github.com/Piotrekk16/Kopiowanie5`.

Wyjście:

```
Wiek pierwszej osoby 30
Miejsce urodzin pierwszej osoby Olsztyn
Wiek drugiej osoby 30
Miejsce urodzin drugiej osoby Olsztyn
Po zmianie
Wiek pierwszej osoby 31
Miejsce urodzin pierwszej osoby Gdańsk
Wiek drugiej osoby 30
Miejsce urodzin drugiej osoby Gdańsk
Referencje odwołują się do tego samego obiektu
```

3 Głęboka kopia

Głęboką kopię stosujemy do klas z polami referencyjnymi.

Modyfikujemy kod klasy `Osoba`:

```
class Osoba
{
    private int wiek; //deklaracja pola
    public Urodziny ur; //pole referencyjne

    //konstruktor domyślny (uzupełniamy o tworzenie
       obiektu)
    public Osoba()
    {
        ur = new Urodziny();
    }

    //konstruktor parametryczny
    public Osoba(int wiek,string miejsce)
    {
        this.wiek = wiek;
        ur = new Urodziny();
        ur.UstawMiejsceUr(miejsce);
    }
```

```
        public void UstawWiek(int wiek) //metoda na
            ustawienie wieku
        {
            this.wiek = wiek;
        }

        public int PobierzWiek() //metoda na pobranie wieku
        {
            return wiek;
        }

        public Osoba GlebokaKopia() //implementacja
            głębokiej kopii
        {
            Osoba tempOsoba = new Osoba();
            tempOsoba.wiek = this.wiek;
            tempOsoba.ur.UstawMiejsceUr(this.ur.PobierzMiejsceUr());
            return tempOsoba;

        }

    }
```

Kod w `Main`:

```
            Osoba o1 = new Osoba(30,"Olsztyn"); //tworzy
                pierwszy obiekt i nadajemy mu wiek 30
            Osoba o2 = new Osoba(); //tworzymy drugi obiekt
            o2 = o1.GlebokaKopia(); //kopiowanie z uzyżciem
                głębokiej kopii

            //wartości po kopiowaniu
            Console.WriteLine("Wiek pierwszej osoby {0}",
                o1.PobierzWiek());
            Console.WriteLine("Miejsce urodzin pierwszej
                osoby {0}", o1.ur.PobierzMiejsceUr());
            Console.WriteLine("Wiek drugiej osoby
                {0}",o2.PobierzWiek());
            Console.WriteLine("Miejsce urodzin drugiej
```

```
        osoby {0}", o2.ur.PobierzMiejsceUr());

    o1.UstawWiek(31); //zmiana wieku pierwszej osoby
    o1.ur.UstawMiejsceUr("Gdańsk"); //zmiana
        miejsca urodzin pierwszej osoby

    //wartości po zmianie
    Console.WriteLine("Po zmianie");
    Console.WriteLine("Wiek pierwszej osoby {0}",
        o1.PobierzWiek());
    Console.WriteLine("Miejsce urodzin pierwszej
        osoby {0}", o1.ur.PobierzMiejsceUr());
    Console.WriteLine("Wiek drugiej osoby {0}",
        o2.PobierzWiek());
    Console.WriteLine("Miejsce urodzin drugiej
        osoby {0}", o2.ur.PobierzMiejsceUr());

    //sprawdzenie referencji
    if (Object.ReferenceEquals(o1.ur, o2.ur))
        Console.WriteLine("Referencje odwołują się
            do tego samego obiektu");
    else
        Console.WriteLine("Referencje nie odwołują
            się do tego samego obiektu");
```

Kod na githubie: `https://github.com/Piotrekk16/Kopiowanie6`.
Wyjście:

```
Wiek pierwszej osoby 30
Miejsce urodzin pierwszej osoby Olsztyn
Wiek drugiej osoby 30
Miejsce urodzin drugiej osoby Olsztyn
Po zmianie
Wiek pierwszej osoby 31
Miejsce urodzin pierwszej osoby Gdańsk
Wiek drugiej osoby 30
Miejsce urodzin drugiej osoby Olsztyn
Referencje nie odwołują się do tego samego obiektu
```