

Metode

Pisanje metod

Parametri metod

Klicanje metod

Razlogi za pisanje metod

- ❑ Dosedanji programi so bili kratki, enostavni
 - Tudi naprej bodo taki ;-)
- ❑ "Pravi" programi so večji
 - Tudi nekaj milijonov vrstic
 - Ne moremo napisati v enem kosu
 - Razdelitev v manjše kose
 - Vsak kos napišemo posebej
 - C# – metode
- ❑ Drugi programski jeziki: funkcije, podprogrami, procedure, subrutine, ...

Obstoječe metode

- ❑ Doslej smo uporabljali že cel kup metod
 - izpis na zaslon (`Console.WriteLine("bla")`)
 - Pretvorba niza v število: `Convert.ToInt32("12")` ali `int.Parse("12")`
 - Izračun sinusa (`Math.Sin(12.4)`)
 - ...
 - ❑ Metode lahko pišemo (sestavljamo) tudi sami
 - ❑ Pravzaprav jih že ves čas
 - `Main`
 - Lahko pa jih poimenujemo po svoje
 - ❑ `Main`
 - Metoda s točno predpisanim namenom / obliko / ...
-

Zakaj pišemo metode

- ❑ "skrivanje" postopka - metodo napišemo in jo potem lahko uporabimo, ne da bi skrbeli, kako je napisana
 - ❑ preglednost programa
 - ❑ program - zaporedje "neodvisnih" postopkov (branje podatkov, obdelava podatkov (lahko spet iz več neodvisnih postopkov), predstavitev rezultatov
 - ❑ Nariši diagram poteka
 - ❑ Lažje ažuriranje in popravljanje programa
-

Kdaj pisati metodo

- ❑ Pomembna odločitev
 - ❑ Navadimo se predvsem s prakso
 - ❑ Načela
 - Preglednost
 - Uporabnost v drugih programih
 - "samozadostnost"
 - V metodah NE izpisujemo ali beremo, če to ni izrecni namen metode
-

Zakaj v metodah ne izpisujemo/beremo

- ❑ Zakaj omejevati uporabnost metode?
 - ❑ Če npr. podatek preberemo, metodo ne bomo mogli uporabljati, če bomo podatek na primer izračunali!
 - ❑ Če rezultat izpišemo, metode ne bomo mogli uporabiti, če bomo hoteli z rezultatom še kaj početi.
 - ❑ Primer: Sestavimo metodo, ki bo izračunala mesečni račun za elektriko za nekega naročnika
 - Če bomo podatke o porabi prebrali znotraj metode, naše metode ne bomo mogli uporabiti v sistemu, ki podatke o porabi dobi avtomatsko
 - Če bomo rezultat izpisali, metode ne bomo mogli uporabiti kot del sistema, ki izračuna povprečno porabo skupine uporabnikov
-

Delitev – vrste metod

- ❑ Prva delitev:
 - Statične
 - Objektne
 - ❑ Najprej bomo obravnavali le statične

 - ❑ Druga delitev:
 - Metode, ki ne vračajo ničesar
 - Metode, ki vračajo rezultat
 - ❑ Tip metode = tip rezultata, ki ga metoda vrača
 - ❑ Metode lahko sprejmejo argumente: parametri metode
-

Primer metode in njenega klica

- ❑ Metoda tipa int z dvema vhodnima podatkom

```
public static int Max(int a, int b)
{
    /* metoda vrne večje od dveh celih števil */
    if (a > b) return a;
    return b; // sem smo prišli le, če b >= a
}
```

- ❑ Klici metode

```
x = Max(12, x);
...
Console.WriteLine("Večje med števili " + a +
    " in " + c + " je " + Max(a, c));
...
y = Max(Max(a, b) + Max(c, d), 2 + Max(100, x * x));
```

Razlaga posameznih stavkov in besed

- ❑ `public`
 - Vsi deli programa lahko uporabljajo to metodo
 - Do nadaljnega vse metode "public"
 - ❑ `static`
 - Statična metoda – razlaga v poglavju Objektno programiranje
 - Do nadaljnega vse metode "static"
 - ❑ `int` (za `public static`)
 - Metoda je tipa `int`, kar pomeni da bo rezultat tipa `int`
 - ❑ `Max`
 - Ime metode (poljubno! – nenapisano pravilo: začnemo z veliko črko)
 - ❑ `(int a, int b)`
 - Argumenta (parametra) metode
 - Oba sta tipa `int`
 - Formalni argument (dejanski argument bomo uporabili pri klicu metode)
 - ❑ `return`
 - Vračanje rezultata: tip rezultata mora biti enak tipu metode
 - Konec izvajanja metode
-

Kje pišemo metode

```
class MojProgram
{

    // tu napišemo metode, eno za drugo

    //glavni program - metoda main
    public static void Main(string[] parametri)
    {

    }

}
```

Kako je videti program

```
class MojProgram
{
    public static int Max(int a, int b)
    {
        /* vrne vecje od dveh celih stevil */
        if (a > b) return a;
        return b;
    }
}
```

Definicija metode

```
public static void Main(string[] parametri)
{
    // določi največje število med prebranimi //
    ...
    Console.WriteLine("Največje število med " +
        x + ", " + y + " in " + z + " je " + Max(x, Max(y, z)));
}
}
```

Klic metode

Klic metode

Formalni in dejanski parametri

```
public class MojProgram
```

```
{
```

```
    public static int Max(int a, int b)
    { /* vrne vecje od dveh celih stevil */
        if (a > b) return a;
        return b;
    }
```

Formalna parametra

```
    public static void Main(string[] parametri)
    { // določi največje število med prebranimi //
        Console.WriteLine("Največje število med " +
            x + ", " + y + " in " + z + " je " + Max(x, Max(y, z)));
```

Dejanska parametra

Dejanska parametra

```
        Console.WriteLine("Največje število med " +
            x + " in " + y + " je " + Max(x, y));
```

```
    }
```

```
}
```

Formalni in dejanski parametri / argumenti

- ❑ Formalni parameter/argument
 - Za opis delovanja metode
 - ❑ Dejanski parameter/argument oz vhodni podatek metode
 - Začetna vrednost argumenta metode
 - ❑ Klic metode `Max(12, y)`
 - Spremenljivka (parameter) `a` v metodi dobi vrednost 12
 - Parameter `b` v metodi dobi vrednost, ki je shranjena v spremenljivki `y`
-

Klic metode

- ❑ Metodo tipa void (ne vrača ničesar) kličemo tako da napišemo
 - Ime metode in njene dejanske parametre
`ime(dejanski parametri)`
 - ❑ Metodo, ki ni tipa void kličemo **vedno v stavku**, npr.
 - `Console.WriteLine(ime_metode(dejanski_parametri));`
 - `Podatk_tip spremenljivka=ime_metode(dejanski_parametri);`
 - ❑ Pri klicu metod moramo paziti na naslednje
 - tip metode
 - ujemanje parametrov
 - prenaša se vrednost parametra
-

Metode _ odgovori na vprašanja

```
public static int Vsota(int n)
{
    int r = 1;
    int i = 1;
    while (i < n) {
        i = i + 1;
        r = i * r;
    }
    return r;
}
```

- Kako je ime metodi?
 - Kakšen je tip rezultata, ki ga vrača metoda?
 - Koliko argumentov sprejme metoda?
 - Kakšni so tipi in imena argumentov?
 - Kaj metoda počne? Ali lahko predlagaš boljše ime za metodo?
 - Napiši vsaj dva klica metode!
 - Napiši program, ki demonstrira delovanje metode.
-

Zgled: Inicialke

- ❑ Napiši metodo `Inicialke`, ki sprejme dva niza znakov (ime in priimek) in vrne niz sestavljen iz inicialk. Na primer, `Inicialke("France", "Preseren")` naj vrne niz `"F.P."`.
 - Ime metode
 - `Inicialke`
 - Tip metode = tip rezultata, ki ga metoda vrne
 - `string`
 - Število argumentov
 - `2`
 - Tip in imena argumentov
 - `string ime`
 - `string priimek`
-

Zgled: Inicialke – metoda v C#

```
public static string Inicialke(string ime, string priimek)
{
    char zp, zi;
    zi = ime[0];
    zp = priimek[0];
    return zi + "." + zp + ".";
}
```

Vaje: tipi metod in argumenti

- ❑ Kakšni so tipi naslednjih metod, kakšni so argumenti?
 - `public static string PrvaBeseda(string besedilo)`
 - `public static double[] Produkt(double[] A, double[] B)`
 - `public static void Foo()`
 - `public static int Najdi(string geslo, string[] slovar)`
 - `public static int Produkt(int prvo, int drugo)`
-

Vaje: glave metod

- ❑ Za vsakega od naslednjih opisov zapišimo deklaracijo metode = glavo metode (kakšen tip metoda vrača in kakšne argumente sprejme)
 - Metoda, ki v podani tabeli nizov poišče najdaljši niz in ga vrne.
`public static string Najdaljsi(string[] a)`
 - Metoda, ki najde eno izmed obeh rešitev dane kvadratne enačbe (pod predpostavko, da so rešitve realna števila).
`public static double Kvadratna(double a, double b, double c)`
 - Metoda, ki dano tabelo nizov uredi po abecedi in vrne urejeno tabelo.
`public static string[] Uredi(string[] a)`
 - Metoda, ki na ekran izpiše "Kdor to bere, je osel!".
`public static void Osel()`
-

Vaja: Kaj naredi naslednji program

```
public static int Povecaj(int a)
{
    a = a + 1; // povecam a za 1
    return a; //ker je metoda tipa int, vrne rezultat
}
public static void Main(string[] parametri)
{
    string pod;
    int a;
    int naj;
    Console.Write("Številko: ");
    pod = Console.ReadLine();
    a = Convert.ToInt32(pod);
    naj = Povecaj(a);
    Console.WriteLine("naj = " + naj + "\na = " + a);
}
```

Parametri (argument) metod

- Parametre metod lahko kličemo na dva načina:
 - po vrednosti (tako smo delali v vseh dosedanjih primerih)
 - Ob klicu metode parameter dobi vrednost, kot je določena v dejanskem parametru
 - Potem ni nobene povezave med dejanskim in formalnim parametrom: metoda ustvari svoje lastne spremenljivke
 - Po referenci
 - Ob klicu metode parametri dobijo naslov spremenljivke, ki nastopa kot parameter metode: metoda tako dejansko dela s spremenljivkami iz “glavnega programa”
-

Zgled: Povprečna vrednost

- Napišimo metodo, ki dobi za vhodna podatka (parametra) dve decimalni števili, vrne pa njuno povprečno vrednost

```
public static double Povprecje(double x, double y)
{
    return (x + y) / 2;
}
```

Vaja: vsota dvomestnih števil, deljivih s 5

- Napišimo metodo, ki ne sprejme nobenih parametrov, izračuna in vrne pa vsoto vseh dvomestnih števil, ki so deljiva s 5!

```
public static int VsotaDvomestnihDeljivihs5() //glava funkcije
{
    int vsota = 0; //začetna vsota je 0
    for (int i = 1; i <= 100; i++)
    {
        if (i % 5 == 0) //če ostanek pri deljenju deljiv s 5
            vsota = vsota + i; //število prištejemo k vsoti
    }
    /*metoda vrne rezultat: vsoto vseh dvomestnih števil,
    deljivih s 5*/
    return vsota;
}
```

Vaja – število znakov v stavku

- Napišimo metodo, ki dobi dva parametra: poljuben STAVEK in poljuben ZNAK. Metoda naj ugotovi in vrne kolikokrat se v stavku pojavi izbrani znak.

```
static int Kolikokrat(string stavek, char znak) //glava metode
{
    int skupaj = 0; //Začetno število znakov je 0
    for( int i=0; i<stavek.Length; i++ )
    {
        if (stavek[i] == znak) /tekoči znak primerjamo z našim/*
            skupaj++;
    }
    return skupaj; //Metoda vrne skupno število najdenih znakov
}
```

```
static void Main(string[] args) //Glavni program
{
    Console.Write("Vnesi poljuben stavek: ");
    string stavek = Console.ReadLine();
    Console.Write("Vnesi znak, ki te zanima: ");
    char znak = Convert.ToChar(Console.Read());
    Console.WriteLine("V stavku je " + Kolikokrat(stavek, znak)+" znakov "+znak+".");
}
```

Vaja: Število Pi

- Število PI lahko izračunamo tudi kot vsoto vrste $4 - 4/3 + 4/5 - 4/7 + 4/9 - \dots$. Napiši metodo, ki ugotovi in vrne, koliko členov tega zaporedja moramo sešteti, da se bo tako dobljena vsota ujemala s konstanto `Math.PI` do npr. vključno devete decimalke. Metoda naj ima za parameter število ujemajočih se decimalk. Rešitev za npr. 9 decimalk: 1096634169

```
static long StClenov(int decimalk)
{
    double pi=4; //definiramo in inicializiramo začetno vrednost za pi
    double clen; //tekoči člen zaporedja
    long i=1; //definicija števca členov
    while (Math.Round(pi, decimalk) != Math.Round(Math.PI, decimalk))
    {
        clen = 4.00 / (i * 2 + 1); //izracun tekocega clena
        if(i % 2 != 0)
            pi = pi - clen; //lihe člene odštevamo
        else
            pi = pi + clen; //sode člene prištevamo
        i++;
    }
    return i;
}
```

Vaja: Število Pi

- ❑ Še klic metode

```
Console.WriteLine("Računam koliko členov zaporedja 4  
- 4/3 + 4/5 - 4/7 + 4/9 - ... je potrebno\sešteti, da bo tako dobljena vsota enaka konstanti  
Math.PI!");
```

```
Console.WriteLine("\nTrenutek .....\//ujemanje na 9 decimalk
```

```
Console.WriteLine("Število členov: " + StClenov(9));
```

Vaja: Najmanjši na začetek

- Napiši metodo `Najmanjsi_na_zacetek`, ki sprejme tabelo celih števil in v njej zamenja prvi (ničti- indeks 0) in najmanjši element te tabele. Na primer, ko poženemo

```
int[] b = {10, 4, 7, 18, 2, 7};
```

```
Najmanjsi_na_zacetek(b);
```

je tabela `b` enaka `{2, 4, 7, 18, 10, 7}`.

Vaja: Najmanjši na začetek

- ❑ Tip metode: void
 - POZOR: če tabela nastopa kot parameter metode, se vsaka sprememba v tej tabeli odrazi tudi v programu, kjer smo to metodo klicali.
 - Zakaj – trenutno še “skrivnost” (če tabela nastopi kot parameter metode, metoda dobi naslov te tabele in ne ustvari nove, lokalne oz. začasne tabele)
 - ❑ Poiščemo najmanjšega
 - ❑ Zamenjamo 0-tega in najmanjšega
 - Potrebujemo indeks!
 - `t[kje] = t[0];`
 - `t[0] = min;`
-

Vaja: Najmanjši na začetek

□ Kako pa poiščemo najmanjši element v tabeli

- Zapomnimo si tako najmanjšega, kot njegov indeks
- Kandidat za najmanjšega je 0-ti

```
int kje = 0; //začetni indeks
```

```
int min = t[0]; //začetni minimalni element
```

- V zanki pregledamo vse preostale

- Če je tekoči element manjši od trenutno najmanjšega, si njegovo vrednost in njegov indeks zapomnimo

```
if (t[i] < min)
```

```
{
```

```
    min = t[i]; //zapomnimo si minimalen element
```

```
    kje = i; //zapomnimo pa si tudi njegov indeks
```

```
}
```

Vaja: Papajščina

- Sestavimo metodo `public static string Papajscina(string s)`, ki dani niz `s` pretvori v "papajščino". To pomeni, da za vsak samoglasnikom, ki se pojavi v nizu, postavi črko `p` in samoglasnik ponovi.

```
public static string papajscina(string s)
{
    string sPapaj = "";
    for (int i=0;i<s.Length;i++)
    {
        //metoda bo delovala tudi če so v staku velike črke

        char znak=char.ToUpper(s[i]);
        if (znak=='A' || znak=='E' || znak=='I' || znak=='O' ||
znak=='U')
            sPapaj = sPapaj + s[i] + 'p'+s[i]; /*dodamo znak
            'p' in še samoglasnik se ponovi*/
        else sPapaj = sPapaj + s[i];
    }
    return sPapaj; //vračanje papajščine
}
```

Vaja: kakšen je izpis

- ❑ Kakšen je izpis naslednje metode, če jo pokličemo s stavkom *Vraca(10)*? Najprej reši "peš" in šele potem preizkusi z računalnikom.

```
public static void Vraca(int n)
{
    int i = 1;
    while (i <= n)
    {
        if (i % 2 != 0) Console.Write(i);
        else Console.Write('#');
        i++;
    }
    Console.WriteLine('#');
}
```

Vaja

- Dana je metoda

```
public static int Izracun(int n, int k)
{
    int r = 1;
    int i = 1;
    while (i < n)
    {
        r = n + 1;
        i = k * r;
    }
    return (r+i);
}
```

- Kako je ime metodi?
 - Kakšen je tip rezultata, ki ga vrača metoda?
 - Koliko argumentov sprejme metoda?
 - Kakšni so tipi in imena argumentov?
 - Napiši primer klica te metode v “glavnem programu”
-

Vaja: Tabela

- ❑ Sestavi tabelo naključnih celih števil med 1 in 100.
 - ❑ Prepiši jih v novo tabelo tako, da bodo v novi tabeli najprej elementi, ki imajo v prvi tabeli indekse 0, 2, 4, ..., potem pa še elementi z lihimi indeksi.
 - ❑ Če ime prvotna tabela elemente 2, 4, 23, 5, 45, 6, 8 so v novi tabeli elementi razporejeni kot 2, 23, 45, 8, 4, 5, 6.
 - ❑ Izpiši obe tabeli po 10 v vrsto.
-

Vaja: Tabela

□ Razbijemo na metode

- Generiraj – ustvari tabelo
- Preloži – preloži na zahtevan način
- Izpiši – izpiše po 10 v vrsto

□ `public static int[] Generiraj(int velikostTab, int sm, int zm)`

□ `public static int[] Prelozi(int[] tabela)`

□ `public static void Izpisi(int[] tabela)`

Vaja: Tabela – glavni program

```
public static void Main(string[] arg)
{

    int veltab = 50; //velikost tabele
    int[] tabOrig;
    int[] tabNova;
    int sM = 1; // spodnja meja števil v tabeli
    int zM = 100; // zgornja meja števil v tabeli

    // generiraj naključno tabelo
    tabOrig = Generiraj(veltab, sM, zM);
    /* prelozi v novo, . tako da so elementi s sodimi
       indeksi pred lihimi*/
    tabNova = Prelozi(tabOrig);
    Izpisi(tabOrig);
    Izpisi(tabNova);
}
```

Vaje

- Oglejte si spodnjo metodo in poskusite ugotoviti, kaj dela.

```
public static string PozdravljenSvet(int kolikokrat)
{
    string vrni = "";
    int stevec = 0;
    while(stevec < kolikokrat)
    {
        vrni = vrni + "Pozdravljen, svet! ";
        stevec = stevec + 1;
    }
    return vrni;
}
```

- Sestavi metodo Pravokotnik, ki sprejme naravno število n in m ter na zaslon izpiše pravokotnik iz znakov + višine n in dolžine m.

Primer: Pravokotnik(3,5);

```
+++++
+++++
+++++
```

Vaje

- Kakšen je izpis naslednje metode, če jo pokličemo s stavkom *Vraca(3)*? Najprej reši "peš" in šele potem preizkusi z računalnikom.

```
public static void Vraca(int n)
{
    int i = 1;
    while (i <= n)
    {
        if (i % 2 != 0) Console.Write(i);
        else Console.Write('#');
        i++;
    }
    Console.WriteLine('#');
}
```

Vaje

- ❑ Napiši metodo `PrvaStevka`, ki za dano *pozitivno* celo število vrne njegovo prvo številko (kot tip `int`).

Primer:

```
int stevka = PrvaStevka(12484);  
Vrednost spremenljivke stevka je 1.
```

- ❑ Napiši metodo `public static void Izpis(int[] tab)`, ki dano tabelo izpiše na zaslon.

Primer:

```
int[] tab = {1, 2, 3, 4, 5};  
Izpis(tab);  
Izpiše nam [1, 2, 3, 4, 5]
```
