

Odgovor:

- a) $a * b + a * c$,
- b) $3 * (2 * a - 8 * b) / c$,
- c) $(a + b) * (3 * k + 1) / (k - 3)$.

75. Šta je imenski prostor?

Odgovor:

Imenski prostor (engl. *namespace*) predstavlja skup povezanih klasa određen imenom. Imenski prostor eliminiše potrebu za posebnim imenima koja po pravilu znače uvođenje raznih prefiksa po kojima će se imena razlikovati od bilo kog drugog. Preporučuje se da se svaka klasa definiše u određenom imenskom prostoru. Za izradu imenskog prostora koristi se rezervisana reč **namespace** iza koje sledi ime.

76. Definirati imenski prostor *Osoblje* koji sadrži klasu *Radnik*.

Odgovor:

```
namespace Osoblje
{
    class Radnik
    {
    }
}
```

77. Dati opšti oblik pristupa elementima imenskog prostora.

Odgovor:

Pristup elementima imenskog prostora je oblika

ime_prostora.ime_elementa_pristupa

Na primer, klasa *Console* se nalazi u imenskom prostoru *System*. To znači da potpuna referenca klase *Console* glasi *System.Console*.

78. Da li postoji mogućnost ugnježdživanja imenskih prostora? Ako postoji, dati primer.

Odgovor:

Imenski prostori se mogu ugnezditi:

Primer:

```
namespace Osoblje
{
    namespace Radnici
    {
        class RadnoVreme
    }
}
```

```

    {
    }
}

```

79. Navesti klase tipova podataka koje su definisane u imenskom prostoru *System*.

Odgovor:

U tabeli 1.14 prikazani su tipovi podataka koji su definisani u imenskom prostoru *System*.

Tabela 1.14 Tipovi podataka u imenskom prostoru *System*

Tip podatka	Klasa
bool	Boolean
byte	Byte
sbyte	SByte
char	Char
decimal	Decimal
double	Double
float	Single
int	Int32
uint	UInt32
long	Int64
ulong	UInt64
object	Object
short	Int16
ushort	UInt16
string	String

80. Navesti osnovna pravila pisanja metode *Main*.

Odgovor:

Pravila pisanja metode *Main* su:

- korišćenje velikog slova „M“, tj. „Main“,
- određivanje jednog *Main* za ulaznu tačku programa,
- deklarisanje *Main* kao **static void** Main.

81. Čemu služi rezervisana reč **using**?

Odgovor:

Rezervisana reč **using** se koristi za uključivanje imenskih prostora neophodnih za izvršenje aplikacije. Na primer, imenski prostor *System* uključuje se naredbom **using System**;

82. Šta je *Console*?

Odgovor:

Console je klasa koja omogućava C# aplikacijama da obavljaju standardne ulazno/izlazne operacije. Koristi se samo u konzolnim aplikacijama, tj. aplikacijama koje se pokreću iz komandnog prozora (engl. *Command Prompt*). Dve osnovne metode ove klase su *Read* i *Write*.

83. Prikazati strukturu C# programa.

Odgovor:

C# program ima sledeću osnovnu strukturu:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Zbirka
{
    class Program83
    {
        // Opis glavnog programa aplikacije.
        static void Main(string[] args)
        {
            // Kôd koji pokreće aplikaciju.
        }
    }
}
```

84. Šta je izlazna sekvenca?

Odgovor:

Izlazna sekvenca (engl. *escape sequence*) specijalan je znak koji se ne prikazuje i koristi se pri izradi izlaznih izveštaja za specificiranje upravljačkih znakova. Predstavlja se obrnutom kosom crtom (\) iza koje sledi drugi znak ili simbol.

85. Navesti izlazne sekvence.

Izlazne sekvence prikazane su u tabeli 1.15.

Tabela 1.15 *Izlazne sekvence*

Izlazna sekvenca	ASCII vrednost	Opis
\a	007	Zvučni signal
\b	008	Vraća kursor unazad
\t	009	Pomera kursor za jedan tabulator unapred
\n	010	Pomera kursor na početak novog reda
\v	011	Vertikalni tabulator
\f	012	Pomera papir štampača na početak sledeće stranice
\r	013	Pomera kursor na početak reda
\"	034	Navodnik (")
\'	039	Apostrof (')
\?	063	Znak pitanja
\\	092	Obrnuta kosa crta (\)
\0	000	Null znak

86. Šta se dobija na izlazu sledećeg programa?

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Zbirka
{
    class Program86
    {
        static void Main(string[] args)
        {
            Console.Write("Zovem se \n\t Milan");
        }
    }
}

```

Odgovor:

Zovem se
Milan

87. Navesti razliku između opisnih i izvršnih naredbi.

Odgovor:

Izvršna naredba izražava određenu akciju koju treba računar da izvrši, a opisna naredba sadrži informacije koje mogu biti korisne u programu ili su neophodne za pravilno izvršavanje određenih izvršnih naredbi. Izvršne naredbe, prema vrsti akcije koju definišu, mogu biti: naredbe obrade, upravljanja ili prenosa.

88. Šta je regularni izraz? Dati primer.

Odgovor:

Regularni izraz (engl. *regular expression*) posebno je formatiran niz znakova koji se koristi za pronalaženje šablona u tekstu.

Primer:

Broj pošte se sastoji isključivo od 5 cifara, ni manje ni više. Prihvata se 18000, a ne prihvata 180, 180000, 18_00. Regularni izraz kojim se određuje broj pošte piše se u obliku `^\d{5}$`, što znači: pronađi bilo koju cifru „`\d`“ koja se ponavlja 5 puta `{5}` u celom znakovnom nizu „`^....$`“.

89. Šta je *Regex*?

Odgovor:

Regex je klasa u imenskom prostoru *System.Text.RegularExpressions*. Sadrži statičke metode koje dozvoljavaju korišćenje regularnih izraza bez eksplicitne izrade *Regex* objekta. Najpoznatija statička metoda klase *Regex* jeste *Match*, koja pretražuje ulazni znakovni niz radi provere da li se javlja određeni regularni izraz i vraća traženi rezultat u obliku pojedinačnog *Match* objekta.

90. Šta označava sledeći regularni izraz: `^[0-9]+\s+([a-zA-Z]+\s+[a-zA-Z]+\s+[a-zA-Z]+)\s+$`?

Odgovor:

Značenje regularnog izraza je sledeće:

- `[0-9]+` znači da string počinje s jednom ili više cifara,
- `\s+` cifre su praćene jednom ili više praznina,
- `([a-zA-Z]+\s+[a-zA-Z]+\s+[a-zA-Z]+)` znači da sledeća reč počinje s jednim ili više znakova bilo velikih ili malih slova, a da to nije broj ili može početi s jednim ili više znakova praćenih prazninom ili s jednim ili više znakova,
- operator ili označava se uspravnim crtom `( | )`

Primer: 11000 Beograd

91. Napisati nazive metoda koje sadrži klasa *Math* za izračunavanje matematičkih funkcija:

e^x	_____
$\sin x$	_____
$\sqrt{x}$	_____
$\ln x$	_____
$\arctg x$	_____
$ x $	_____
$\text{th} x$	_____
$\log_{10} x$	_____

Odgovor:

e^x	Exp(x)
$\sin x$	Sin(x)
$\sqrt{x}$	Sqrt(x)
$\ln x$	Log(x)
$\arctg x$	Atan(x)
$ x $	Abs(x)
$\text{th} x$	Tanh(x)
$\log_{10} x$	Log10(x)

92. Kodirati sledeće aritmetičke izraze:

a)
$$\frac{a - b}{c + \frac{a}{c + \frac{b}{c - b}}}$$

b)
$$(tg x + 1) \cdot (x^{4 \cdot n} + 4 \cdot x + 1).$$

Odgovor:

a) $(a - b) / (c + a / (c + b / (c - b)))$,

b) $(\text{Math.Tan}(x) + 1) * (\text{Math.Pow}(x, 4 * n) + 4 * x + 1.0).$

93. Kodirati sledeće izraze:

$$\text{a) } \frac{\ln \sqrt{u^2 + 2 \cdot \cos u}}{2,8 \cdot u - 1},$$

$$\text{b) } \ln \sqrt{\lg^2 x + 0,2 + \frac{\cos y}{e^{x \cdot y}}}.$$

Odgovor:

a) `Math.Log(Math.Sqrt(Math.Pow(u, 2) + 2 * Math.Cos(u))) / (2.8 * u - 1.0),`

b) `Math.Log(Math.Sqrt(Math.Pow(Math.Tan(x), 2.0) + 0.2 + Math.Cos(y) / Math.Exp(x * y))).`

94. Šta je aritmetička naredba?

Odgovor:

Naredba kojom se vrednost aritmetičkog izraza dodeljuje promenljivoj zove se aritmetička naredba.

95. Navesti opšti oblik aritmetičke naredbe dodele.

Odgovor:

Opšti oblik aritmetičke naredbe dodele je

$$p = a$$

gde su: p – ime promenljive,

a – aritmetički izraz.

96. Koje je dejstvo naredbe dodele?

Odgovor:

Izračunava se vrednost aritmetičkog izraza, izračunata vrednost se transformiše u tip podatka saglasno tipu promenljive kojoj se vrednost dodeljuje i dodeljuje se promenljivoj.

97. Kodirati sledeću aritmetičku naredbu

$$y = \sqrt{\frac{e^{\frac{x}{2}} + 1,2 \cdot \sin 2 \cdot x}{3,3 \cdot \cos x - 7,1 \cdot e^x}}.$$

Odgovor:

`y = Math.Sqrt((Math.Exp(x / 2) + 1.2 * Math.Sin(2 * x)) / (3.3 * Math.Cos(x) - 7.1 * Math.Exp(x))).`

98. Kodirati sledeće aritmetičke naredbe:

$$\text{a) } y = \frac{e^{\sin^3 x} + \ln(\operatorname{arctg} x)}{\sin x},$$

$$\text{b) } y = \frac{x}{x^5 \cdot \left(e^{\frac{1,4}{x}} - 1 \right)}.$$

Odgovor:

$$\text{a) } y = \text{Math.Exp(Math.Pow(Math.Sin(x), 3) + Math.Log(Math.Atan(x)))} / \text{Math.Sin}(x),$$

$$\text{b) } y = x / (\text{Math.Pow}(x, 5) * (\text{Math.Exp}(1.4 / x) - 1.0)).$$

99. Kodirati sledeću aritmetičku naredbu

$$y = \left(a \cdot x^3 + b \cdot \sqrt[5]{x^2} + \frac{c}{x} \right)^{\sin x}.$$

Odgovor:

$$y = \text{Math.Pow}(a * \text{Math.Pow}(x, 3) + b * \text{Math.Pow}(x, 2.0 / 5.0) + c / x, \text{Math.Sin}(x)).$$

100. Kodirati sledeću logičku naredbu

$$y = \sqrt{\frac{2 + x - a \cdot x^2}{x + 2}} > 0 \vee e^x < x - 1 \wedge \sqrt{x - 1} \leq 0.$$

Odgovor:

$$y = \text{Math.Sqrt}((2 + x - a * \text{Math.Pow}(x, 2)) / (x + 2)) > 0 \parallel \text{Math.Exp}(x) < (x - 1) \&\& \text{Math.Sqrt}(x - 1) <= 0.$$

101. Kodirati sledeće naredbe:

$$\text{a) } z = \sqrt{|x|} + e^x + \sin^2 x + \cos x^2 + \operatorname{th} x + \operatorname{arctg} x,$$

$$\text{b) } z = \neg(x \vee y) \wedge x \vee \neg y.$$

Odgovor:

$$\text{a) } z = \text{Math.Sqrt(Math.Abs}(x)) + \text{Math.Exp}(x) + \text{Math.Pow}(\text{Math.Sin}(x), 2) + \text{Math.Cos}(\text{Math.Pow}(x, 2)) + \text{Math.Tanh}(x) + \text{Math.Atan}(x),$$

$$\text{b) } z = !(x \parallel y) \&\& x \parallel !y.$$

102. Dati izlaznu listu sledećeg programa:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Zbirka
{
    class Program102
    {
        static void Main(string[] args)
        {
            int i = 3, k = 4, a, b, c;
            a = 10 * i / k;
            b = 10 / k * i;
            c = 10 / i / k;

            Console.WriteLine("a = {0}, b = {1}, c = {2}", a, b, c);
        }
    }
}
```

Rešenje:

a = 7, b = 6, c = 0

a = 7, b = 6, c = 0

103. Dati izlaznu listu sledećeg programa:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Zbirka
{
    class Program103
    {
        static void Main(string[] args)
        {
            double x, k, a, y, b;

            x = 3.75;
            k = 2 * x + 0.3;
            a = 3 * x + 0.75;
            y = k + 1.7;
            b = a + 2.75;

            Console.Write("y = {0}, b = {1} ", y, b);
        }
    }
}
```

Rešenje:

$y = 9.5$, $b = 14.75$

104. Dati izlaznu listu sledećeg programa:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Zbirka
{
    class Program104
    {
        static void Main(string[] args)
        {
            bool x, y, z;
            x = false;
            y = true;

            z = !(x && y);

            Console.WriteLine("z = {0}", z);
        }
    }
}
```

Rešenje:

$z = \text{True}$

105. Šta je algoritam?

Odgovor:

Reč algoritam je arapskog porekla (po imenu matematičara Alhorezmija iz IX veka) kojom se u najširem smislu označava skup pravila koji vodi rešenju nekog matematičkog problema.

Algoritam je potpuno i precizno definisana procedura (postupak) koja uzima jednu ili više ulaznih vrednosti i daje jednu ili više izlaznih vrednosti. Te vrednosti se zovu ulazni i izlazni podaci algoritma. Algoritam je konačan niz logičnih koraka potrebnih da se reši neki problem, dok je program algoritam zapisan na nekom programskom jeziku.

To znači da se algoritmom definiše metoda (procedura) i redosled izvršavanja određenih operacija, tako da se svaki algoritam, kao i program, može rastaviti na module. Različito povezivanje modula dovodi do nastanka različitih struktura algoritama.

U praksi se sreću tri strukture: linijska, razgranata i ciklična.