

01. Razlike izmedju imperativnih i funkcionalnih jezika?

- modeli na kojima se zasnivaju
- programi (imperativni: niz instrukcija, funkcionalni: funkcija)
- ključna operacija (imperativni: dodeljivanje, funkcionalni: primena funkcije)
- dodela (imperativni: eksplicitna, funkcionalni: nema eksplicitne dodele)
- rad sa memorijom (imperativni: eksplicitni, funkcionalni: implicitan)
- tipovi (imperativni: tipiziran jezik, funkcionalni: bez tipova (samo 2 osnovna: atom i lista))

02. Za formalan opis sintakse programskih jezika se koriste?

- kontekstno-slobodne gramatike (CFG)

03. Da li interpreter prevodi program iz izvornog u izvršni?

da) **ne) (to radi kompajler)**

04. Bazne apstrakcije su:

- promenljive, elementarni tipovi podataka, operacija dodeljivanja

05. Sta su alijasi ?

- dve ili vise promenljivih koje pristupaju jednoj memorijskoj lokaciji

06. Pomocu cega se alijasi kreiraju ?

- pomocu pointera i referentnih promenljivih

07. Sta je doseg promenljive ?

- segment koda programa u kome je promenljiva poznata i moze da se koristi.

08. Dinamicki doseg?

- zasniva se na sekvenci poziva potprograma
- odredjuje su u vreme izvršenja programa

09. Interpretacija realnog tipa (double) u C# je:

- a) binarna **b) preko mantise i promenljive** c) BCD code

Dealokacija Garbage promenljive u C# je:

- a) implicitna

b) explicitna (ovo mi je rekao kolega da je tacno i da je proverio na uvidu)

Dealokacija promenljivih u C# se vrši:

a) implicitno (ovo bi trebalo da je tacno prema slajdovima)

- b) explicitno

10. Rezultat izvršavanja sledeceg koda je:

```
int brojac ( ) { static int
k = 0; return ++ k;
}
int main ( ) {
printf (" brojac = %d /n", brojac ( ) );
printf (" brojac = %d /n", brojac ( ) );
return 0;
}
```

Izlaz: brojac = 1 brojac = 2

11. Rezultat izvršavanja sledeceg koda je:

```
class ClassA {
public int ChangeValue ( int v )
{ v = v + 1; return v ; }
}
class PassByValueTest {
static void Main ()
{ ClassA ca = new ClassA() ;
int v1 = 8 ;
int v2 = ca.ChangeValue (v1);
System.Console.WriteLine ("v1 = {0}, v2 = {1}", v1, v2);
}
}
8, 9
```

12. Koja je bazna klasa u C#?

- System.Object

13. Da li u C# nasledjivanje moze da bude visestruko?

a) da **b) ne**

14 .Net frejmwork je:

a) **razvojno okruzenje** **b) izvrsno okruzenje** (oba)

15. Inkapsulaciona konstrukcija u programskom jeziku C# je:

a) package **b) klasa** c) namespace

16. Koji deo .NET Framework-a sadrzi predefinisane tipove podataka?

- CTS – Common Type System

17. Koji su referentni tipovi podataka?

- klase, nizovi, stringovi i delegati

18. Koji su to vrednosni tipovi podataka ?

- int, short, long, float, double, decimal, byte, char i bool, enum i struct (zapis)

19. Koji su to predefinisani vrednosni tipovi podataka ?

- int, short, long, float, double, decimal, byte, char i bool (najosnovnije)

20. Koji su to korisnicki definisani vrednosni tipovi podataka ?

- enum i struct (zapis)

21. C# kompajler vrsi translaciju programa u:

a) masinski jezik b) bytecode **c) IL(Intermediate Language)**

22. Mehanizam za inkapsulaciju implementacije apstraktnog tipa podataka u C# je

a) struktura podataka **b) klasa** c) package

Izvedena klasa nasleđuje sve osim konstruktora i destruktora bazne klase.

Koncepti objektno orijentisanog programiranja:

Učarenje, nasleđivanje, polimorfizam

Sta se koristi za "interfejs entiteta" u C++?

a) public klauzula b) public modifikator pristupa c) private klauzula

XSLT je:

a) deklarativan jezik b) proceduralan jezik c) logicki jezik

SGML je standard za definisanje i reprezentovanje strukture različitih tipova elektronskih dokumenta.

DTD (Document Type Definition): nasledjen od SGML-a, poseban jezik za opis strukture dokumenta, vrlo ogranicene mogucnosti za definisanje tipova.

Kardinalnost elemenata u xml semi specificira se preko:

a) seqence i choice konstruktora **b) minOccurs i maxOccurs** c) ...

Konstruktor all u XML semi definise:

a) grupu u kojoj svi elementi mogu maksimalno pojaviti jednom

b) Uredjenu grupu elemenata

c) Grupu u kojoj redosled elemenata nije bitan

XSLT Procesor za transformaciju elementa koristi:

a) XML semu b) HTML dokument **c) XSLT stylesheet dokument**

Atribut match u templejtu elementu <xsl:template match ="/">

- povezuje templejt sa **korenim elementom**

Rezultat izvršavanja XPath naredbe oblika /Poruka/Salje nad definisanim XML dokumentom je:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<Poruka>
```

```
  <Salje> Ivan Todorovic </Salje>
```

```
  <Prima> Milan Petrovic </Prima>
```

```
</Poruka>
```

a)Ivan Todorovic b)Milan Petrovic **c)<Salje> Ivan Todorovic </Salje>**

XML Reader reprezentuje:

a) DOM model XML parsera **b) Pull model XML Parsera** c) SAX model XML parsera

DOM parseri u .NET-u su:

- MSXML parser

- XML document

DOM parser se koristi kad se zahteva obrada vecine elemenata, kad se zahteva dinamicki pristup i manipulacija.

1. Osnovni delovi SOAP poruke su:

- a) Envelope, Header, Body,
- b) Root, Metadata
- c) methodcall, methodname.

a) Envelope, Header, Body.

2. W3C standardi za definisanje XML dokumenata su:

- a) XSLT
- b) DTD
- c) SAX
- d) XML schema
- e) ne znam

b) DTD i d) XML schema

3. Da li je XML dokument dobro oformljen?

- a) da
- b) ne

```
<?xml version="1.0" encoding="UTF-8"?>
<2006RezultatiTesta mesec="januar">
  <Polagalo> 200 </Polagalo>
  <Polozilo> 100 </Polozilo>
</2006RelultatiTesta>
```

b)ne

4. Da li je sledeci XML dokument validan?

```
<?xml version="1.0" standalone="yes"?>
<PORUKE>
  <PORUKA Datum="10.10.2005">
    <OD> Jovana </OD>
    <ZA> Nikola </ZA>
    <TEKST> Vidimo se sutra u 9 </TEKST>
  </PORUKA>
</PORUKE>
</xml>
```

- a) da
- b) ne

b) ne

5. Koja su od sledecih tvrdjenja tacna?

- a) XSLT koristi Xpath
- b) XSLT nije W3C standard
- c) XSLT je XML document

a) XSLT koristi Xpath

6. Kljucne tehnologije arhitekture Web servisa su:

- a) DCOM
- b) XML
- c) SOAP
- d) CORBA
- e) ne znam

b) XML i c) SOAP

7. Formulacija XML scheme se izvodi koriscenjem predefinisanih tagova.

- a) da
- b) ne

b) ne

8. Koje je ime protokola koji se koristi za prenos objekata preko Web-a?

- a) WSDL
- b) DCOM
- c) SOAP
- d) Ne znam

b) DCOM

9. Koji je redosled pristupa cvorovima pri citanju XML dokumenta koriscenjem klase XMLReader?

- a) prvi red 1
 drugi red 2 3 (2 je povazana sa 4 i 5)
 treci red 4 5 6
- b) prvi red 1**
drugi red 2 5 (2 je povazana sa 3 i 4)
treci red 3 4 6
- c) prvi red 6
 drugi red 4 5 (4 je povazana sa 1 i 2)
 treci red 1 2 3

PRVA GRUPA

1. Životni vek promenljive je

- A. segment koda programa u kome je promenljiva poznata i može se koristiti
- B. vreme vezivanja promenljive za određenu memorijsku lokaciju, u toku izvršavanja programa**
- C. Ne znam

2. BCD je

- A. 8-bitni kod
- B. 16-bitni kod
- C. 4-bitni kod**
- D. ne znam

3. Statički doseg promenljive zasniva se na

- A. Sekvenci poziva potprograma
- B. Leksičkoj strukturi programa**
- C. Ne znam

4. Vrednost stack_dinamičke promenljive je

- A. Perzistentna
- B. Reinicijalizuje se pri svakom pozivu funkcije**
- C. Ne znam

5. Za leksičku specifikaciju programskih jezika koristi se

- A. Binarno stablo
- B. Konteksno-slobodna gramatika
- C. Regularni izrazi**
- D. Ne znam

6. Mehanizam koji obezbeđuje vidljivost interfejsa ATP u C++ je:

- A. public modifikator pristupa
- B. public klauzula**
- C. protected klauzula
- D. Ne znam

7. Statička promenljiva alocira se u statičkom segmentu memorije

- A. Više puta
- B. Samo jednom**
- C. Ne znam

8. Koje su od navedenih imenovane inkapsulacije:

- A. Java packages**
- B. C# assemblies
- C. C# namespaces**
- D. Ne znam

9. Dat je fragment C# koda

```
class ClassA {
public int ChangeValue ( int v )
    {      v = v + 2; return v ; }
}
class PassByValueTest {
    static void Main ()
    {      ClassA ca = new ClassA() ;
        int v1 = 5 ;
        int v2 = ca.ChangeValue (v1);
        System.Console.WriteLine ("v1 = {0}, v2 = {1}", v1, v2);
    }
}
```

Koje će vrednosti imati promenljive v1 i v2 posle poziva metode ChangeValue?

v1 = _____ v2 = _____

DRUGA GRUPA

1. Doseg promenljive je

- A. **segment programskog koda u kome je promenljiva poznata i može se koristiti**
- B. vreme vezivanja promenljive za određenu memorijsku lokaciju, u toku izvršavanja programa
- C. Ne znam

2. Problem tzv. «garbage» dinamičke promenljive dele programski jezici u kojima se dinamička promenljiva dealocira

- A. Implicitno
- B. Eksplicitno**
- C. Ne znam

3. Inkapsulaciona konstrukcija u C# je:

- A. package
- B. assembly**
- C. private klauzula
- D. ne znam

4. Dinamičkii doseg promenljive zasniva se na

- A. Sekvenci poziva potprograma**
- B. Leksičkoj strukturi programa
- C. Ne znam

5. Interna reprezentacija decimalnog tipa (Decimal type) u C# je

- A. Binarna
- B. Preko mantise i eksponenta
- C. BCD code**
- D. Ne znam

6. Koja od sledećih situacija kreira "dangling pointer"

- A. Dealociranje bloka koji je sadržao pointer na drugi podtojeći memorijski blok
- B. Posle dealokacije dinamičke promenljive, postoji pointer koji sadrži adresu takve promenljive**
- C. Pointer koji pokazuje na dinamičku promenljivu, postavljen je na "null"
- D. ne znam

7. Životni vek C *static* promenljive koja je deklarirana u nekoj funkciji jednak je

- A. Vremenu izvršavanja programa**
- B. Vremenu izvršavanja funkcije
- C. Ne može se predvideti
- D. Ne znam

8. Šta razlikuje funkcionalni od imperativnog programskog jezika

- A. U funkcionalnom jeziku svi potprogrami se moraju deklarirati pomoću ključne reči *function*
- B. U funkcionalnom jeziku ne postoje promenljive i operacija dodeljivanja**
- C. Ne postoji razlika
- D. U funkcionalnom jeziku ne postoji kontrolna struktura selekcije
- E. Ne znam

9. Dat je fragment koda u C:

```
int brojac () {  
    static int i = 3;  
    return ++ i;  
}  
int main () {  
    printf (" brojac = %d /n", brojac ( ) );  
    printf (" brojac = %d /n", brojac ( ) );  
    return 0;  
}
```

Napisati koje će se vrednosti brojača prikazati na konzoli, posle prvog i drugog poziva funkcije *brojac*.

Posle I poziva vrednost brojaca je: **4**

Posle II poziva vrednost brojaca je: **5**