Пријемни испит – Мастер 2019
Студијски програм: Софтверско инжењерство и рачунарске науке

Шифра задатака | 7 | 7 | 4 | 6 | 8 | 2 |

1.	**Када се нит прави реализацијом интерфејса *Runnable*, тада се имплементира (реализује):**
a.	*run*() метода интерфејса *Runnable*
b.	*start*() метода интерфејса *Runnable*
c.	*begin*() метода интерфејса *Runnable*
d.	*execute*() метода интерфејса *Runnable*
e.	*perform*() метода интерфејса *Runnable*
n.	Не знам
2.	**Која се метода позива да одблокира нит која је блокирана *wait*() методом:**
a.	*inform*() метода
b.	*stop*() метода
c.	*notify*() метода
d.	*sleep*() метода
e.	*check*() метода
n.	Не знам
3.	**Уколико се жели утврдити, да ли се нит још извршава, користи се метода:**
a.	*isRun*() класе *Thread*
b.	*isAlive*() класе *Thread.*
c.	*isExecute*() класе *Thread*
d.	*isActive*() класе *Thread*
e.	*isVisible*() класе *Thread*
n.	Не знам
4.	**Уколико се јави потреба да две или више нити деле заједнички ресурс и имају искључиви (ексклузивни) приступ до заједничког ресурса, који се мора обезбедити механизам:**
a.	перзистенција
b.	активација
c.	пасивизација
d.	синхронизација
e.	материјализација
n.	Не знам
5.	**Уколико се желе видети све *IP* адресе које су везане за изабрано симболичко име користи се метода:**
a.	*getAllIpAdress*() класе *InetAddress*
b.	*getAllByPort*() класе *InetAddress*
c.	*getAllByName*() класе *InetAddress.*
d.	*getAllByClass*() класе *InetAddress*
e.	*getAllIpObject*() класе *InetAddress*
n.	Не знам
6.	**Сокет, у ширем смислу, је механизам који омогућава комуникацију:**
a.	између објеката који се извршавају на истом рачунару
b.	између нити које се извршавају на истом рачунару
c.	између екранских форми које се извршавају на различитим рачунарима у мрежи
d.	између програма и базе података
e.	између програма који се извршавају на различитим рачунарима у мрежи
n.	Не знам
7.	**Укључивање *JDBC API*-а у Јава програм ради се преко следеће наредбе:**
a.	import java.jdbc.*;
b.	import java.db.*;
c.	import java.swing.*;
d.	import java.sql.*;
e.	import java.oracle.*;
n.	Не знам
8.	**Шта је резултат фазе анализе код упрошћене Ларманове методе развоја софтвера:**
a.	база података
b.	пословна логика
c.	архитектура софтверског система
d.	софтверски патерни

e.	екранске форме
n.	Не знам
9.	**Актор (учесник) представља:**
a.	тестера софтверског система
b.	аналитичара софтверског система
c.	доменског експерта
d.	спољног корисника система
e.	пројектанта базе података
n.	Не знам
10.	**Предуслови код системских операција у фази анализе код упрошћене Ларманове методе:**
a.	указује на то шта ће се десити, пре него што се системска операција изврши
b.	указују на то шта је требало да се деси, како би системска операција могла да се изврши, а не како се то десило
c.	указује на то шта је и како је требало да се деси, како би системска операција могла да се изврши
d.	указује на то шта ће се десити, пошто се изврши системска операција
e.	указује на то како се нешто десило, како би системска операција могла да се изврши, а не шта је требало да се деси
n.	Не знам
11.	**При извршењу операција убаци и промени објекат (слог) проверавају се:**
a.	ограничења меморијског простора
b.	ограничења клијентског и серверског програма
c.	системска ограничења
d.	ограничења оперативног система
e.	вредносна и структурна ограничења
n.	Не знам
12.	**Перзистентни оквири су засновани на:**
a.	холивудском принципу
b.	принципу независности системских операција
c.	принципу слабе повезаности класа
d.	принципу отворено-затворено
e.	принципу компатибилности класа
n.	Не знам
13.	**Систем за управљање базама података (database management system, DBMS) поседује механизме за:**
a.	управљање трансакцијама
b.	заштиту од неовлашћеног приступа подацима и заштиту од уништења података
c.	обезбеђење ефикасног коришћења базе података
d.	управљање дистрибуираним деловима базе података
e.	све што је претходно наведено
n.	Не знам
14.	**Интерна репрезентација базе података је:**
a.	физички модел базе података који приказује начини похрањивања и приступа подацима на екстерној меморији
b.	општа логичка структура базе података која приказује концептуални модел разматраног системаа
c.	колекција логички повезаних бинарних и текстуалних датотека
d.	све што је претходно наведено
e.	ништа од претходно наведеног
n.	Не знам
15.	**Нарушавање референцијалног интегритета може се десити у следећем случају:**
a.	у релацији која садржи спољни кључ (референцирајућа релација) вредности атрибута које чине спољни кључ нису одговарајуће због убацивања нове н-торке у релацију
b.	у релацији која садржи спољни кључ (референцирајућа релација) вредности атрибута које чине спољни кључ нису одговарајуће због измене вредности атрибута који чине спољни кључ
c.	у релацији која садржи примарни кључ (референцирана релација) вредности атрибута које чине примарни кључ нису одговарајуће због избацивања н-торке из релације
d.	у релацији која садржи примарни кључ (референцирана релација) вредности атрибута које чине примарни кључ нису одговарајуће због измене вредности атрибута који чине примарни кључ
e.	све што је претходно наведено
n.	Не знам
16.	**Ако су дате две некомпатибилне релације P1 и P2, које су повезане својим примарним и спољним кључевима, применом које операције ћемо добити резултујућу релацију која садржи све н-торке које су појављују и у P1 и у P2:**
a.	спајање
b.	унија

c.	лево спољно спајање
d.	пројекција
e.	централно спољно спајање
n.	Не знам
17.	**Којом *DDL* наредбом се мења дефиниција табеле у *SQL*-у:**
a.	CREATE TABLE
b.	ALTER TABLE
c.	MODIFY TABLE
d.	DROP TABLE
e.	ADD COLUMN
n.	Не знам

18.	**Следећим *SQL* наредбама креирана је, а потом је попуњена подацима табела *ISPLATA*:**
	
	`CREATE TABLE ISPLATA(`
	`zaposlen_id tinyint identity,`
	`satnica decimal NULL,`
	`plata decimal NULL,`
	`provizija decimal NULL,`
	`broj_prodaja tinyint NULL);`
	
	`INSERT ISPLATA (satnica, plata, provizija, broj_prodaja) VALUES`
	`(10.00, NULL, NULL, NULL),`
	`(20.00, NULL, NULL, NULL),`
	`(30.00, NULL, NULL, NULL),`
	`(40.00, NULL, NULL, NULL),`
	`(NULL, 10000.00, NULL, NULL),`
	`(NULL, 20000.00, NULL, NULL),`
	`(NULL, 30000.00, NULL, NULL),`
	`(NULL, 40000.00, NULL, NULL),`
	`(NULL, NULL, 15000, 3),`
	`(NULL, NULL, 25000, 2),`
	`(NULL, NULL, 20000, 6),`
	`(NULL, NULL, NULL, NULL),`
	`(NULL, NULL, 14000, 4);`
	
	Упит:
	`SELECT zaposlen_id,`
	`CAST(`
	`COALESCE(`
	`satnica * 40 * 52, -- 40= br. rad. sati u sedmici; 52= br. sedmica u god.`
	`plata,`
	`provizija * broj_prodaja`
	`) as money`
	`) as 'Isplata'`
	`FROM ISPLATA`
	`ORDER BY 'Isplata';`
	У првом реду приказује (не рачунајући заглавље):

a.	12 NULL
b.	5 10000,00
c.	13 56000,00
d.	6 20000,00
e.	ништа од претходно наведеног
n.	Не знам

19.	**Погледати код који је дат испод. Која од наредних изјава је тачна:**
	`class DevizniPenzioner {`
	`}`
	
	`class Devizni Penzioner {`
	`}`

a.	назив `DevizniPenzioner` није дозвољен у програмском језику Јава
b.	назив `Devizni Penzioner` није дозвољен у програмском језику Јава
c.	обе класе су приватне класе
d.	ове две класе Јава посматра као исту класу

e.	обе класе су јавне класе
n.	Не знам

20.	**Погледати код који је дат испод (број линије кода је дат уз леву ивицу). Шта представља друга линија кода?**
	`1 public class Osoba {`
	`2 //public String ime`
	`3 }`

a.	јавни атрибут име
b.	јавну методу име
c.	једнолинијски коментар
d.	вишелинијски коментар
e.	синтаксну грешку
n.	Не знам

21.	**Погледати кôд који је дат испод. Која од наредних изјава је тачна:**
	`class Metode {`
	`double vratiApsolutnuNulu () {`
	`double rezultat = -273.15;`
	`}`
	`}`

a.	метода враћа -273.15 као повратну вредност
b.	метода има један параметар
c.	метода има два параметра
d.	тип повратне вредности методе је воид
e.	кôд не може да се компајлира
n.	Не знам

22.	**Погледати код који је дат испод. Шта ће се десити кад се покрене *main*() метода?**
	`public class Osoba {`
	`public String ime;`
	`}`
	
	`public class Test {`
	`public static void main(String[] args) {`
	`Osoba o1 = new Osoba();`
	`Osoba o2 = new Osoba();`
	`o1.ime = "Pera";`
	`o2.ime = "Mika";`
	`o2 = o1;`
	`System.out.println(o2.ime);`
	`}`
	`}`

a.	на екрану ће се исписати “Pera”
b.	на екрану ће се исписати “Mika”
c.	на екрану ће се исписати “Pera Mika”
d.	на екрану ће се исписати “Mika Pera”
e.	Јава пријављује грешку
n.	Не знам

23.	**Погледати кôд који је дат испод. Шта ће метода `provera()` исписати на екрану ако се позове и проследе јој се као аргументи бројеви 15 и 2, респективно?**
	`class Provera {`
	`void provera (int a, int b) {`
	`if (a > b) System.out.println("A je vece od B");`
	`if (a < b) System.out.println("A je manje od B");`
	`else System.out.println("A je jednako B");`
	`}`
	`}`

a.	“A je vece od B”
b.	“A je vece od B” и “A je manje od B”
c.	“A je vece od B” и “A je manje od B” и “A je jednako B”
d.	“A je manje od B”
e.	“A je veće od B” и “A je jednako B”
n.	Не знам

24.	**Погледати код који је дат испод. Шта ће метода вратити као повратну вредност ако јој се као аргумент проследи вредност 300?**
	`class Provera {`
	`  boolean proveraIntervala (int a){`
	`if (a < 100 && a > 500)`
	`  return true;`
	`else`
	`  return false;`
	`  }`
	`}`

a.	метода ништа не враћа као повратну вредност
b.	*true*
c.	*false*
d.	a
e.	0
n.	Не знам

25.	**Погледати кôд који је дат испод. Шта ће метода исписати на екрану ако се позове и проследи јој се као аргументи број 150?**
	`class Provera {`
	`  void proveraVisine (int v){`
	`if (v < 153) System.out.println("Nizak");`
	`else if (v < 185) System.out.println("Srednje visine");`
	`else System.out.println("Visok");`
	`  }`
	`}`

a.	"Nizak"
b.	"Srednje visine"
c.	"Visok"
d.	"Nizak" и "Srednje visine"
e.	"Nizak" и "Visok"
n.	Не знам

26.	**Погледати кôд који је дат испод. Шта ће метода вратити као повратну вредност ако се позове са аргументима 3 (као a) и 4 (као b)?**
	`class Petlje {`
	`  int izracunaj(int a, int b) {`
	`  int r = 1;`
	`  for (int i=0; i<b; i++)`
	`  r = r * a;`
	`  return r;`
	`  }`
	`}`

a.	ништа (бесконачна петља)
b.	1
c.	3
d.	12
e.	81
n.	Не знам

27.	**Погледати кôд који је дат испод. Шта ће метода исписати на екрану ако се позове?**
	`class Petlje {`
	`void ispisiBrojeve() {`
	`  for (int i=0; i>100; i++)`
	`  System.out.println(i);`
	`  }`
	`}`

a.	ништа
b.	бројеве од 0 до 100
c.	бројеве од 0 до 99
d.	бројеве од 1 до 99
e.	бесконачна петља
n.	Не знам

28.	**Погледати кôд који је дат испод. Шта ће метода исписати на екрану ако се позове?**
	`class Petlje {`
	`  void ispisiBrojeve(){`
	`  for (int i=0; i<100; i--)`
	`  System.out.println(i);`
	`  }`
	`}`

a.	ништа
b.	бројеве од 0 до 100
c.	бројеве од 0 до 99
d.	бројеве од 1 до 99
e.	бесконачна петља
n.	Не знам

29.	**Погледати кôд који је дат испод. Шта ће метода `ispisi()` исписати на екрану ако се позове са аргументом 22?**
	`class Petlje {`
	`  void ispisi(int n) {`
	`int i = 1;`
	`while (i < n) {`
	`  if (i % 3 == 0 && i % 7 == 0) {`
	`  System.out.println(i + " ");`
	`  i++;`
	`  }`
	`}`
	`  }`
	`}`

a.	ништа (бесконачна петља)
b.	3
c.	3 7
d.	21
e.	3 6 7 9 12 14 15 18 21
n.	Не знам

30.	**Погледати кôд који је дат испод. Шта ће се исписати на екрану кад се позове `main()` метода?**
	`class GFigura {`
	`  void ispisi(){System.out.println("geometrijska figura");}`
	`}`
	`class Trougao extends GFigura{`
	`  void ispisi(){super.ispisi(); System.out.println("trougao");}`
	`}`
	`class PravougliTrougao extends Trougao{`
	`}`
	`public class Test{`
	`  public static void main(String[] args) {`
	`  GFigura gf = new PravougliTrougao();`
	`  gf.ispisi();`
	`  }`
	`}`

a.	ништа
b.	geometrijska figura
c.	trougao
d.	geometrijska figura trougao
e.	trougao geometrijska figura
n.	Не знам