

PRIMJER PITANJA IZ PRVE KONTROLNA ZADAĆA IZ "PROGRAMIRANJA U PROGRAMSKOM JEZIKU JAVA"

STUDENT: _____

Čemu služi JVM? (TEORETSKO PITANJE - 2 BODA)

A	JVM predstavlja kompajler za <i>bytecode</i> .
B	JVM predstavlja interpreter za <i>bytecode</i> .
C	JVM provjerava ispravnost <i>bytecode</i> -a i utvrđuje krše li se sigurnosna ograničenja.
D	JVM dolazi s instalacijom Eclipse-a i služi za pisanje Java programa.
E	JVM prevodi "java" datoteke u "class" datoteke.
F	JVM prevodi "class" datoteke u "java" datoteke.

Pomoću kojeg od navedenih načina je moguće ispisati poruku "Kontrolna zadaća" na konzolu? (TEORETSKO PITANJE - 2 BODA)

A	System.out("Kontrolna zadaća");
B	System("Kontrolna zadaća");
C	System.out.println("Kontrolna zadaća");
D	Console("Kontrolna zadaća");
E	Scanner.out("Kontrolna zadaća");
F	Console.out("Kontrolna zadaća");

Što od navedenog **vrijedi** za apstraktne klase? (TEORETSKO PITANJE - 2 BODA)

A	Apstraktne klase se ne mogu nasljeđivati.
B	Programski jezik Java ne podržava apstraktne klase.
C	Iz apstraktnih klasa nije moguće kreirati objekte.
D	Iz apstraktnih klasa je moguće kreirati objekte.
E	Apstraktne klase ne mogu imati implementacije metoda.
F	Apstraktne klase ne mogu imati privatne varijable.

Koju klasu je u Javi potrebno naslijediti kod definiranja klase za novu iznimku A, ako iznimka A mora spadati u kategoriju iznimaka koje se bacaju u slučaju pogreške programera? (TEORETSKO PITANJE - 2 BODA)

A	java.lang.Exception
B	java.lang.Error
C	java.lang.CheckedException
D	java.lang.RuntimeException
E	java.lang.Throwable
F	java.lang.ThrowableException

Koja vrsta iznimaka mora biti obrađena unutar "try-catch" bloka, pri čemu se u suprotnom Java program ne može ni prevesti? (TEORETSKO PITANJE - 2 BODA)

A	Označene iznimke.
B	Neoznačene iznimke.
C	Iznimke kod izvođenja.
D	Iznimke kod prevođenja.
E	Iznimke kod interpretiranja.
F	Sve vrste iznimaka.

Kako mora izgledati metoda "setBroj" koja postavlja vrijednost "private int broj" u nekoj klasi? (ZADATAK - 3 BODA)

A	<pre>public static setBroj(int broj) { broj = broj; }</pre>	D	<pre>private setBroj(int broj) { broj = broj; }</pre>
B	<pre>public static final void setBroj(int broj) { this.broj = broj; }</pre>	E	<pre>public void setBroj(int broj) { this.broj = broj; }</pre>
C	<pre>private int setBroj(int broj) { super.broj = broj; }</pre>	F	Nijedno od navedenih rješenja ne odgovara traženoj metodi "setBroj".

Kako mora izgledati "prototip" apstraktne metode "izracunajPlacuRadnika" koja prima objekt klase "Radnik" i vraća objekt klase "BigDecimal"? (ZADATAK - 3 BODA)

A	<code>public abstract Radnik izracunajPlacuRadnika(BigDecimal placa);</code>
B	<code>public Radnik izracunajPlacuRadnika(BigDecimal placa);</code>
C	<code>public izracunajPlacuRadnika();</code>
D	<code>public Radnik BigDecimal izracunajPlacuRadnika();</code>
E	<code>public BigDecimal izracunajPlacuRadnika(Radnik radnik);</code>
F	<code>public abstract BigDecimal izracunajPlacuRadnika(Radnik radnik);</code>

Što od navedenog je potrebno ubaciti na mjesto označeno s "??", kako bi metoda "provjera" mogla baciti označenu iznimku pod nazivom "PrvaIznimka"? (ZADATAK - 3 BODA)

```
public void provjera(int i) ?? {  
    if (i < 0) {  
        throw new PrvaIznimka("Prva greška!");  
    }  
}
```

```
public class PrvaIznimka extends  
Exception {  
    public PrvaIznimka(String message) {  
        super(message);  
    }  
}
```

A	throws PrvaIznimka
B	throw Exception
C	throw PrvaIznimka
D	throws Exception
E	throw RuntimeException
F	throws RuntimeException

Kako treba izgledati implementacija sučelja "Pozdravljanje"? (ZADATAK - 3 BODA)

```
public interface Pozdravljanje {  
    String pozdravi(String ime);  
}
```

A	B
<pre>public class DobarDan implements Pozdravljanje { public void pozdravi(String ime) { return "Dobar dan, " + ime + "!"; } }</pre>	<pre>public class DobarDan implements Pozdravljanje { public String pozdravi(String ime); }</pre>
C	D
<pre>public class DobarDan implements Pozdravljanje { public String pozdravi(String ime) { return "Dobar dan, " + ime + "!"; } }</pre>	<pre>public class DobarDan implements Pozdravljanje { public String pozdravi(String ime) { new String("Dobar dan, " + ime + "!"); } }</pre>
E	F
<pre>public class DobarDan implements Pozdravljanje { public void pozdravi(String ime); }</pre>	<pre>public class DobarDan implements Pozdravljanje { public String pozdravi(); }</pre>

Na koji od načina je moguće ispraviti pogrešku u navedenom programskom kodu koji koristi klasu "DobarDan" iz devetog zadatka? (ZADATAK - 3 BODA)

```
public class Glavna {  
    public static void main(String[] args) {  
        DobarDan pozdrav; //prva linija  
        pozdrav.pozdravi("Student");//druga linija  
    }  
}
```

A	Prvu liniju potrebno je zamijeniti s "DobarDan pozdrav = new DobarDan();".
B	Prvu liniju potrebno je zamijeniti s "pozdrav = new DobarDan();".
C	Prvu liniju potrebno je zamijeniti s "DobarDan pozdrav = new DobarDan("Student");".
D	Drugu liniju potrebno je zamijeniti s "pozdrav.pozdravi();".
E	Drugu liniju potrebno je zamijeniti s "pozdrav.pozdrav("Student");".
F	Drugu liniju potrebno je zamijeniti s "pozdrav.dobarDan("Student");".