

# Minta a zh elméleti részére

**Határidő** Nincs megadva határidő**Pont** 10**Kérdések** 10**Időkorlát** Nincs

## Próbálkozások naplója

	Próbálkozás	Idő	Eredmény
LEGUTOLSÓ	<a href="#">1. próbálkozás</a>	13 perc	5 az összesen elérhető 10 pontból

Beadva ekkor: máj 20, 01:34

### 1. kérdés

1 / 1 pont

Miért kapunk fordítási hibát erre az osztályra?

```
abstract class Strange {  
    private int position = 0;  
    public int position( int offset ){  
        position += offset;  
        return position;  
    }  
    public static Strange make(){  
        return new Strange();  
    }  
}
```

- ☐ Egy absztrakt osztályban lennie kell absztrakt metódusnak.
- ☐ Absztrakt osztályban a metódusoknak absztraktnak kell lenniük.
- ☐ Absztrakt osztályban nem lehet adattag.
- ☒ Az absztrakt osztály nem példányosítható.

**Helyes!**

**2. kérdés****1 / 1 pont**

Hányfélt írhatunk az `Object`, `Hi` és `Hello` típusok közül a `???` helyére, hogy a program leforduljon, és a kimenetén *Hello* jelenjen meg?

```
class Hi {  
    public void say(){ System.out.println("Hi"); }  
}  
class Hello extends Hi {  
    @Override public void say(){ System.out.println("Hello"); }  
}  
class Main {  
    public static void greet( ??? with ){ with.say(); }  
    public static void main( String[] args ){  
        Hi greeting = new Hello();  
        greet(greeting);  
    }  
}
```

☐ 3☐ 0☐ 2☒ 1**Helyes!****3. kérdés****0 / 1 pont**

Mi lesz az alábbi program hatása?

```
class First {  
    public void report(){  
        System.out.print("First");  
    }  
}  
class Second extends First {  
    @Override public void report(){  
        System.out.print("Second");  
    }  
}  
class Main {
```

```
public static void main( String[] args ){
    Second second = new Second();
    First first = second;
    first.report();
    second.report();
}
```

☐

Futási hiba: a `first` referenciának nem adható értékül a `second` referencia.

legadott válasz

☒

Fordítási hiba: a `first` referenciának nem adható értékül a `second` referencia.

helyes válasz

☐

Kiírja, hogy `SecondSecond`.

☐

Kiírja, hogy `FirstSecond`.

#### 4. kérdés

0 / 1 pont

Mit ír ki az alábbi program?

```
interface Durable {}
class House implements Durable {}
class Castle extends House {}
class Main {
    public static void main( String[] args ){
        Durable d = new Castle();
        Castle c = new Castle();
        System.out.print( d instanceof House );
        System.out.print( c instanceof House );
    }
}
```

☐

truefalse

helyes válasz

☐

truetrue

Megadott válasz

☐ falsefalse☒ falsetrue

## 5. kérdés

1 / 1 pont

Meghívható-e a `store` metódus a `"John"` paraméterrel?

```
public void store( Object value ){  
    file.println(value);  
}
```

☐ Igen, mert a paraméter dinamikus típusa a statikus típusnak nem altípusa.☐ Igen, mert a paraméter statikus típus a dinamikus típusnak altípusa.☒ Igen, mert a paraméter dinamikus típusa a statikus típusnak altípusa.☐ Nem, mert a paraméter statikus típus a dinamikus típusnak nem altípusa.

Helyes!

## 6. kérdés

1 / 1 pont

Hol használható az `Owen.minTemperature` kifejezés? A `warm.Stove` osztályban vagy a `cool.Fridge` osztályban?

```
package warm;  
class Owen {  
    public static final int minTemperature = -273; // absolute zero  
}
```

☐ Egyikben sem.☐ Mindkettőben.

**Helyes!**☒ Csak a `warm.Stove` osztályban.☐ Csak a `cool.Fridge` osztályban.**7. kérdés****0 / 1 pont**

Mi történik, ha a `Table` osztály default konstruktorával létrehozunk egy `Table` objektumot?

```
public class Table {  
    int legs = 4;  
    public void Legs( int legs ){ this.legs = legs; }  
}
```

**Ielyes válasz**☐ A `legs` mező értéke 4 lesz.☐ Fordítási hiba: hibásan definiáltuk a konstruktort.☐ A `legs` mező értéke annyi lesz, amennyit paraméterként átadunk neki.**Iegadott válasz**☒ Nincs default konstruktora a `Table` osztálynak**8. kérdés****0 / 1 pont**

Melyik állítás igaz az alábbi programra?

```
class Little {  
    private int size = 1;  
    public static void main( String[] args ){  
        Little Richard = new Little();  
    }  
}
```

## helyes válasz

☐

A **Richard** egy lokális változó, mely a végrehajtási vermen tárolódik, míg a **size** a heapen.

## legadott válasz

☒

A **Richard** egy lokális változó, mely a heapen tárolódik, míg a **size** a végrehajtási vermen.

☐

A **size** egy lokális változó, mely a heapen tárolódik, míg a **Richard** a végrehajtási vermen.

☐

A **size** egy lokális változó, mely a végrehajtási vermen tárolódik, míg a **Richard** a heapen.

## 9. kérdés

1 / 1 pont

Melyik fajta változókat szabadítja fel a szemétyűjtő (garbage collector)?

☐

az inicializálatlan változókat

☐

a lokális változókat

☒

a dinamikus változókat

☐

az automatikus változókat

Helyes!

## 10. kérdés

0 / 1 pont

Mi az eredmény?

```
class Mixer {  
    Mixer(Mixer m) { m1 = m; }  
}
```

```
Mixer m1;  
  
public static void main(String[] args) {  
    Mixer m2 = new Mixer();  
    Mixer m3 = new Mixer(m2); m3.go();  
    Mixer m4 = m3.m1;  
    m4.go();  
    Mixer m5 = m2.m1;  
    m5.go();  
}  
void go() { System.out.print("hi"); }  
}
```

☐ hi, majd kivételt kapunk

☐ hihi, majd kivételt kapunk

☒ hihhi

☐ hihi

Ielyes válasz

Iegadott válasz