

Objektumorientált szoftvertervezés - BMEVIIIAC00

2020/21/1

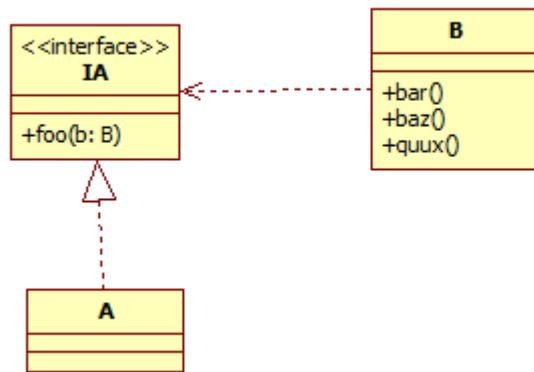
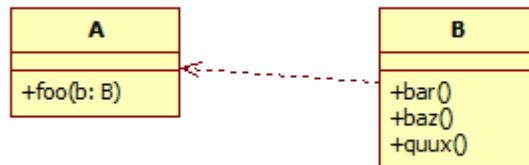
Question 1

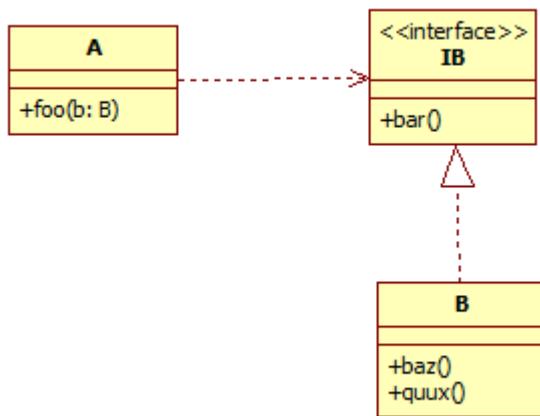
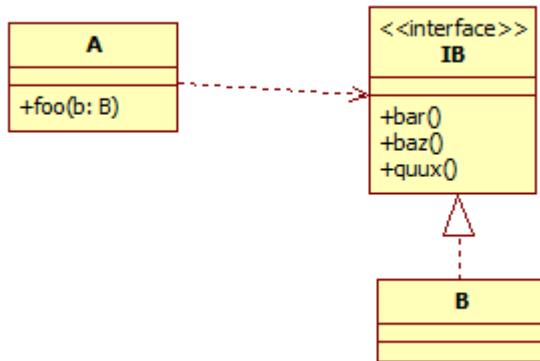
Complete

Mark 0.00 out of 1.00

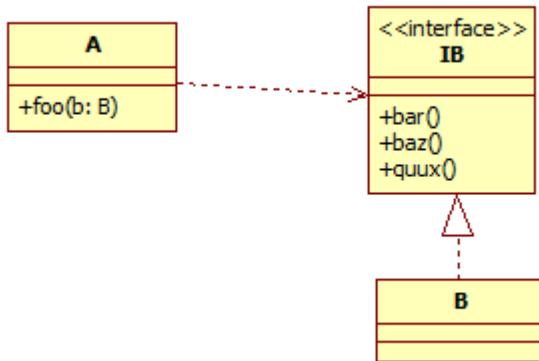
 Flag question

Rajzoljuk át az alábbi osztálydiagramot a DIP elvnek megfelelően!





The correct answer is:



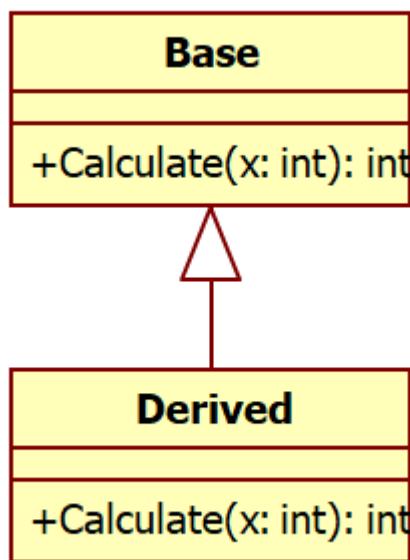
Question 2

Complete

Mark 1.00 out of 1.00

Flag question

Tekintsük az alábbi osztálydiagramot:



Teljesítik-e az alábbi feltételek a Liskov-elvet?

- Base.Calculate post: $0 \leq x \leq 100$
- Derived.Calculate post: $0 \leq x \leq 1000$

Igen

Nem

The correct answer is:

Nem

Question 3

Complete

Mark 1.00 out of 1.00

Flag question

Ha két osztály kölcsönösen függ egymástól, mivel lehetne feloldani a körkörös függőséget?

- Liskov Substitution Principle
- Open/Closed Principle
- Dependency Inversion Principle
- Don't Repeat Yourself
- Law of Demeter
- Single Responsibility Principle

The correct answer is:

Dependency Inversion Principle

Question **4**

Complete

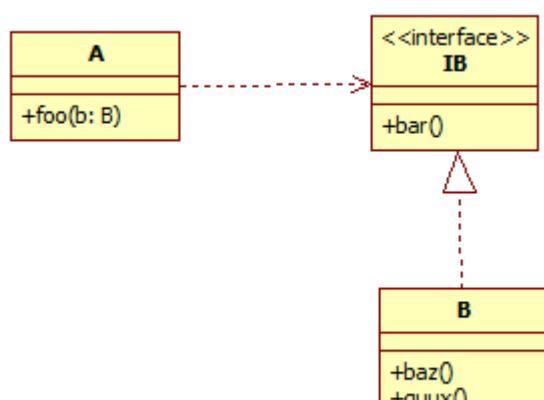
Mark 1.00 out of 1.00

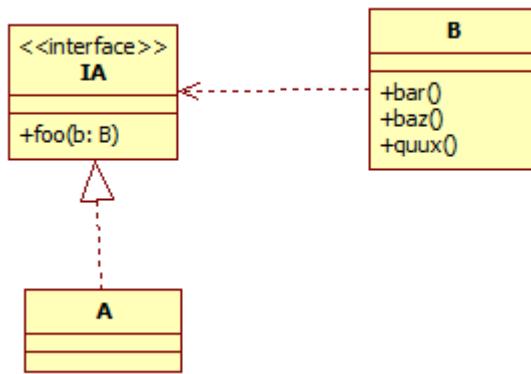
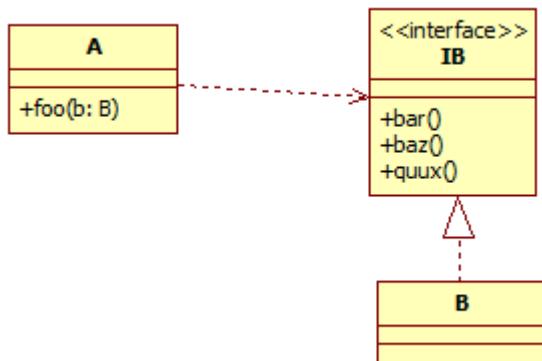
 Flag question

Tudjuk, hogy az A osztály Java implementációja a következő:

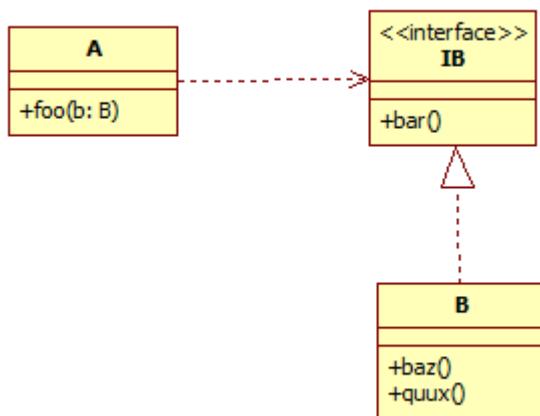
public class A { public void foo(B b) { b.bar(); }}

Rajzoljuk át az alábbi osztálydiagramot a ISP elvnek megfelelően!



A

The correct answer is:



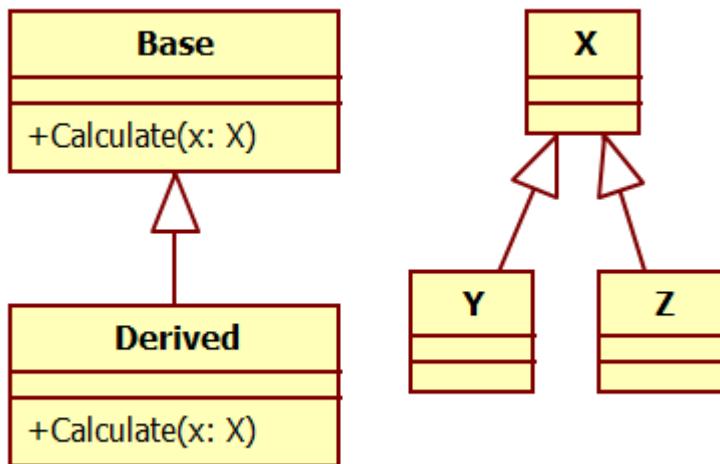
Question 5

Complete

Mark 1.00 out of 1.00

Flag question

Tekintsük az alábbi osztálydiagramot:



Teljesítik-e az alábbi feltételek a Liskov-elvet?

- Base.Calculate pre: x lehet null
- Derived.Calculate pre: x nem lehet null

 Igen Nem

The correct answer is:

Nem

Question 6

Complete

Mark 1.00 out of 1.00

Flag question

Igaz vagy hamis az alábbi állítás?

Az objektumorientált tervezési elveket mindig be kell tartani.

Igaz

Hamis

The correct answer is:

Hamis

Question **7**

Complete

Mark 1.00 out of 1.00

 Flag question

Sérti-e az OCP elvet a Visitor tervezési minta?

Igen

Nem

The correct answer is:

Igen

Question **8**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis az alábbi állítás?

Ha egy osztály sérti az SRP elvet, és bonyolultsága miatt nem lehet több osztályra szétdarabolni, akkor az ISP elv segíthet a függőségek minimalizálásában.

Igaz

Hamis

The correct answer is:

Igaz

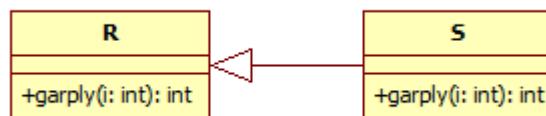
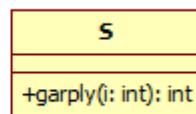
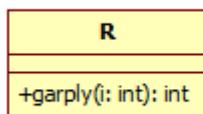
Question **9**

Complete

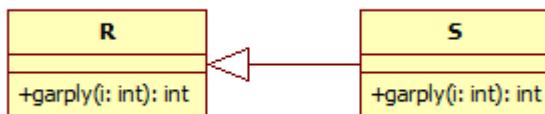
Mark 1.00 out of 1.00

Flag question

Tudjuk, hogy az **R** és **S** osztályok között öröklődés van, és nem sértik a Liskov-elvet. Rajzolja be a közöttük lévő öröklődést, ha tudjuk, hogy **S** **garply()** függvényének utófeltételei erősebbek, mint **R** **garply()** függvényének utófeltételei.



The correct answer is:



Question **10**

Complete

Mark 1.00 out of 1.00

Flag question

Igaz vagy hamis az alábbi állítás?

A szoftver karbantarthatósága szempontjából a magas kohézió és a laza csatolás az előnyös.

 Igaz Hamis

The correct answer is:

Igaz

Question 11

Complete

Mark 1.00 out of 1.00

Flag question

Igaz vagy hamis az alábbi állítás?

Az öröklődés célja az ősosztályban definiált mezők újrahasznosítása.

 Igaz Hamis

The correct answer is:

Hamis

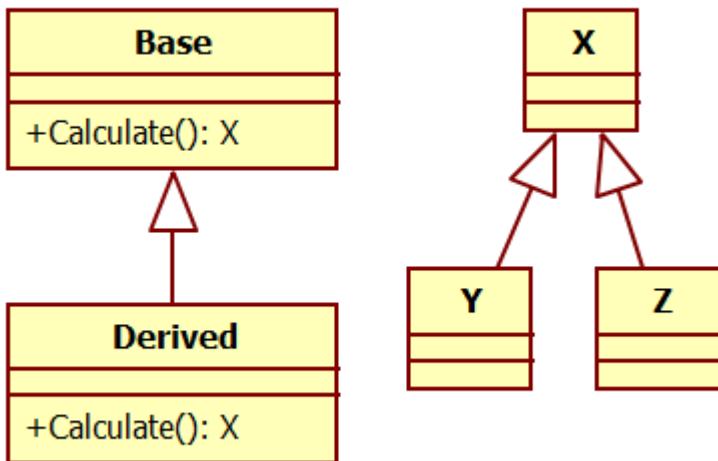
Question 12

Complete

Mark 1.00 out of 1.00

Flag question

Tekintsük az alábbi osztálydiagramot:



Teljesítik-e az alábbi feltételek a Liskov-elvet?

- Base.Calculate post: eredmény lehet null
- Derived.Calculate post: eredmény nem lehet null

Igen

Nem

The correct answer is:

Igen

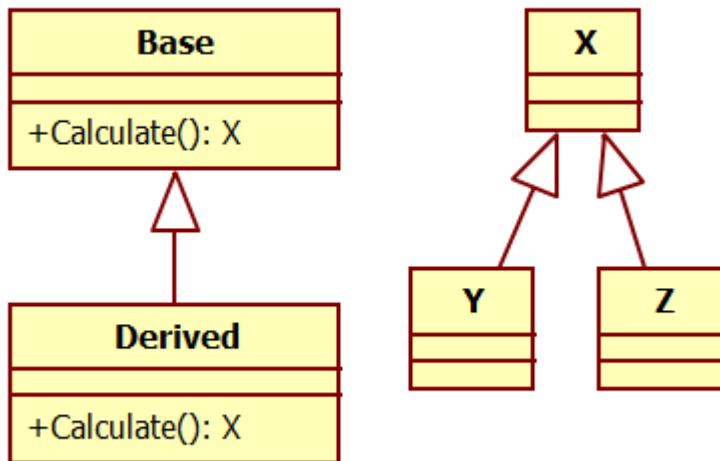
Question **13**

Complete

Mark 1.00 out of 1.00

Flag question

Tekintsük az alábbi osztálydiagramot:



Teljesítik-e az alábbi feltételek a Liskov-elvet?

- Base.Calculate post: eredmény nem lehet null
- Derived.Calculate post: eredmény lehet null

Igen

Nem

The correct answer is:

Nem

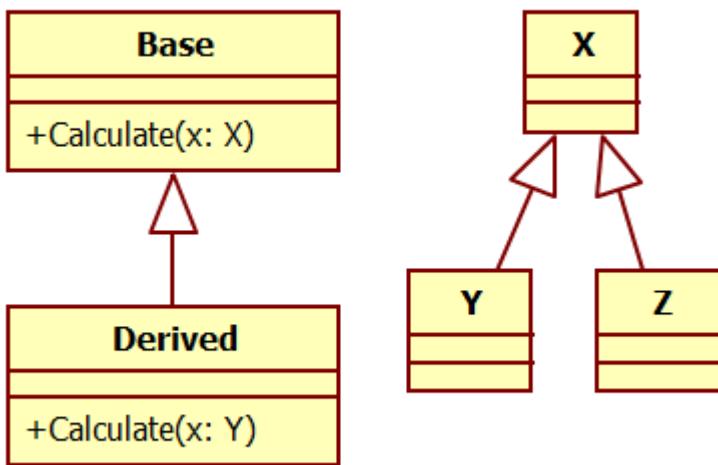
Question **14**

Complete

Mark 0.00 out of 1.00

Flag question

Tekintsük az alábbi osztálydiagramot:



Teljesül-e Liskov-elv?

- Igen
- Nem

The correct answer is:

Nem

Question **15**

Complete

Mark 1.00 out of 1.00

Flag question

Igaz vagy hamis az alábbi állítás?

A többnyelvűséget támogató erőforrásfájlok sértik a Single Choice Principle (SCP) elvet.

- Igaz
- Hamis

The correct answer is:

Hamis**Question 16**

Complete

Mark 1.00 out of 1.00

Flag question

Igaz vagy hamis az alábbi állítás?

Ha sérül a TDA elv, az azt jelezheti, hogy a felelősségek rosszul vannak kiosztva.

 Igaz Hamis

The correct answer is:

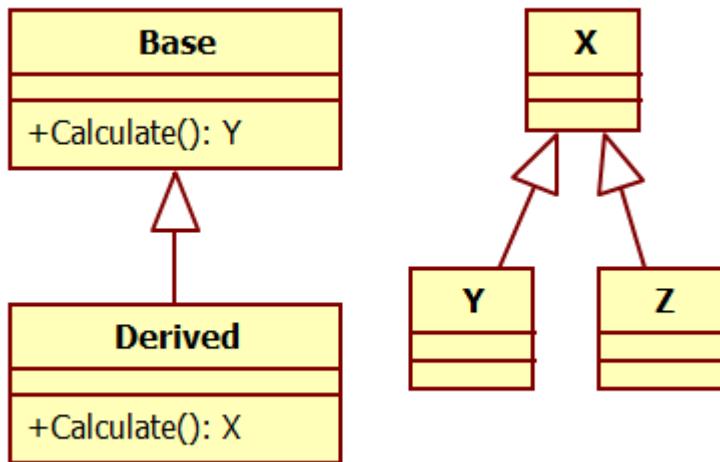
Igaz**Question 17**

Complete

Mark 1.00 out of 1.00

Flag question

Tekintsük az alábbi osztálydiagramot:



Teljesül-e Liskov-elv?

- Igen
 Nem

The correct answer is:

Nem

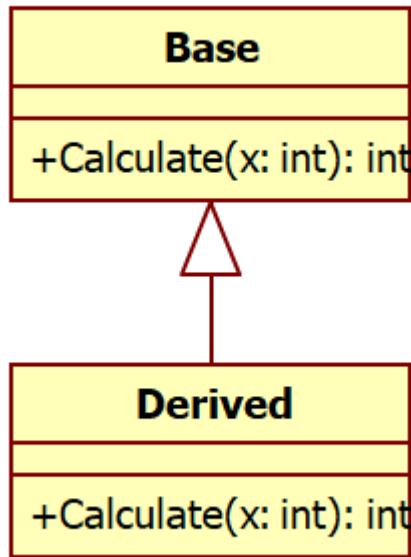
Question **18**

Complete

Mark 1.00 out of 1.00

 Flag question

Tekintsük az alábbi osztálydiagramot:



Teljesítik-e az alábbi feltételek a Liskov-elvet?

- Base.Calculate pre: $0 \leq x \leq 100$
- Derived.Calculate pre: $20 \leq x \leq 50$

- Igen
 Nem

The correct answer is:

Nem

Question **19**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis az alábbi állítás?

A YAGNI elv szerint nincs szükségünk objektumorientált tervezési elvekre.

- Igaz
- Hamis

The correct answer is:

Hamis

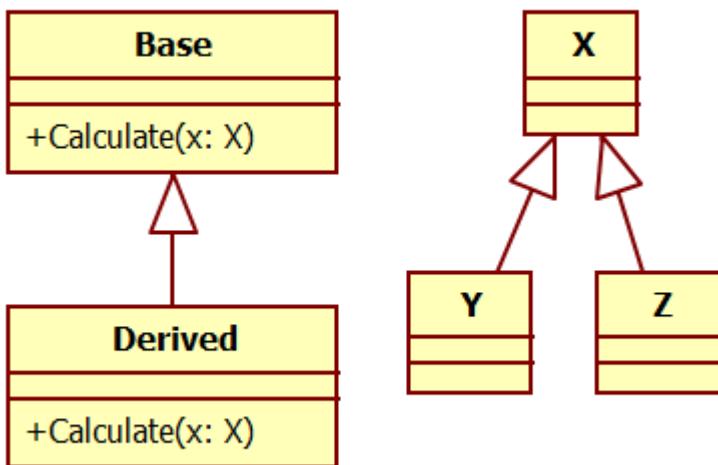
Question **20**

Complete

Mark 0.00 out of 1.00

 Flag question

Tekintsük az alábbi osztálydiagramot:



Teljesítik-e az alábbi feltételek a Liskov-elvet?

- `Base.Calculate pre: x` nem lehet null
- `Derived.Calculate pre: x` lehet null

Igen

Nem

The correct answer is:

Igen

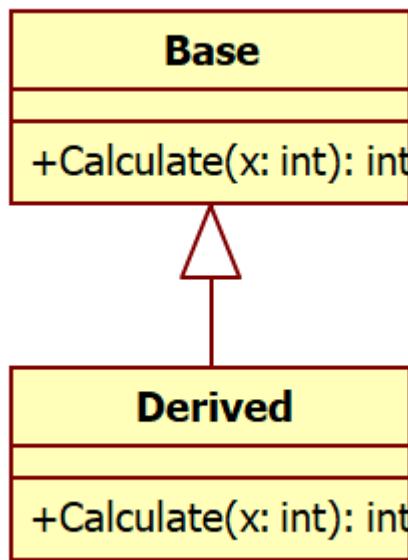
Question **21**

Complete

Mark 1.00 out of 1.00

Flag question

Tekintsük az alábbi osztálydiagramot:



Teljesítik-e az alábbi feltételek a Liskov-elvet?

- Base.Calculate post: $0 \leq x \leq 100$
- Derived.Calculate post: $20 \leq x \leq 50$

Igen

Nem

The correct answer is:

Igen

Question **22**

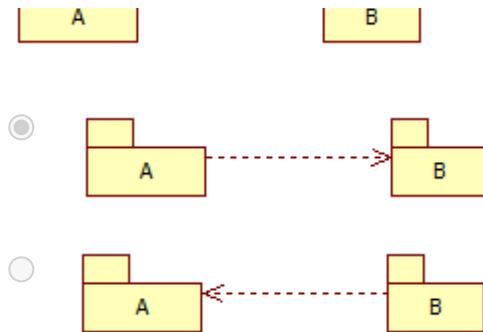
Complete

Mark 1.00 out of 1.00

Flag question

Az **A** csomag gyakran változik, a **B** csomag ritkán. Rajzolja be a függőség irányát az SDP elvnek megfelelően!





The correct answer is:



Question **23**

Complete

Mark 1.00 out of 1.00

Flag question

Sérti-e az ISP elvet a Visitor tervezési minta?

Igen

Nem

The correct answer is:

Igen

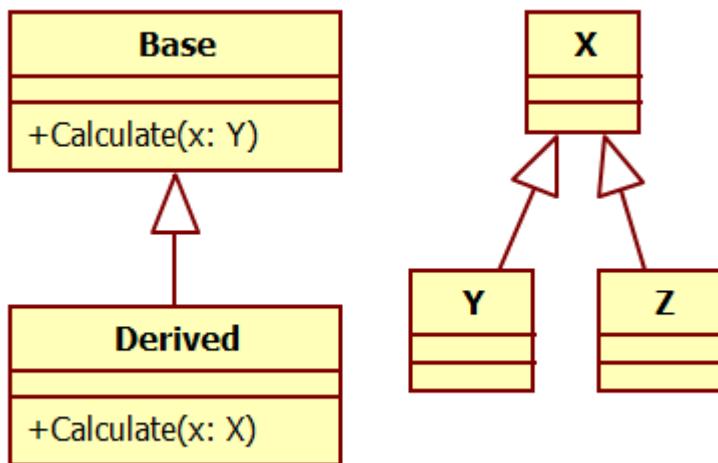
Question **24**

Complete

Mark 0.00 out of 1.00

Flag question

Tekintsük az alábbi osztálydiagramot:



Teljesül-e Liskov-elv?

- Igen
- Nem

The correct answer is:

Igen

Question **25**

Complete

Mark 1.00 out of 1.00

Flag question

Mely tervezési elvek részei a SOLID elveknek?

- Stable Abstractions Principle
- Don't Repeat Yourself
- Stable Dependencies Principle
- Dependency Inversion Principle
- Interface Segregation Principle

- Liskov Substitution Principle
- Tell, don't ask
- Single Responsibility Principle
- Acyclic Dependencies Principle
- Single Choice Principle
- Law of Demeter
- Open/Closed Principle

The correct answers are:

Single Responsibility Principle,
Open/Closed Principle,
Liskov Substitution Principle,
Interface Segregation Principle,
Dependency Inversion Principle

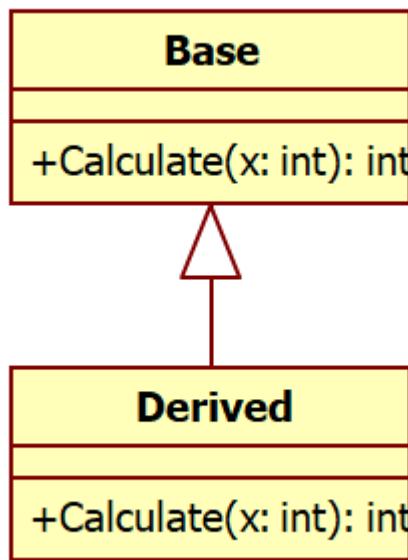
Question **26**

Complete

Mark 1.00 out of 1.00

 Flag question

Tekintsük az alábbi osztálydiagramot:



Teljesítik-e az alábbi feltételek a Liskov-elvet?

- Base.Calculate pre: $0 \leq x \leq 100$
- Derived.Calculate pre: $0 \leq x \leq 1000$

Igen

Nem

The correct answer is:

Igen

Question **27**

Complete

Mark 1.00 out of 1.00

Flag question

Igaz vagy hamis az alábbi állítás?

Egy szoftver akkor van jól tervezve, ha a stabil csomagok függnek az instabil csomagoktól.

Igaz Hamis

The correct answer is:

Hamis

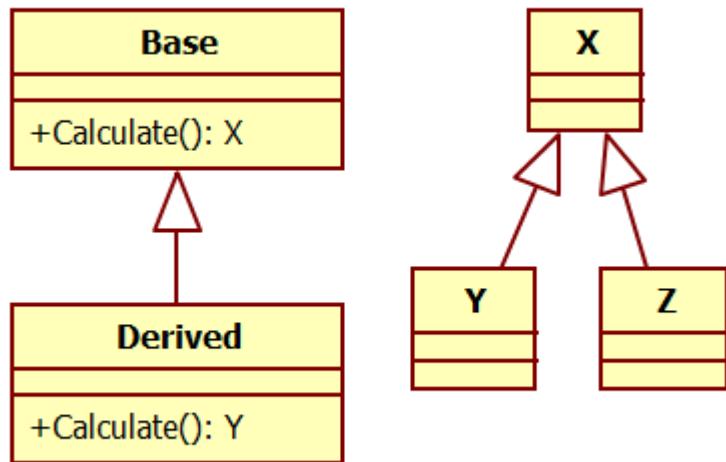
Question **28**

Complete

Mark 1.00 out of 1.00

 Flag question

Tekintsük az alábbi osztálydiagramot:



Teljesül-e Liskov-elv?

 Igen Nem

The correct answer is:

Igen

Question **29**

Complete

Mark 1.00 out of 1.00

Flag question

Sérti-e az DIP elvet a Visitor tervezési minta?

- Igen
- Nem

The correct answer is:

Nem

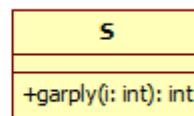
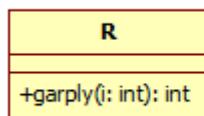
Question 30

 Complete

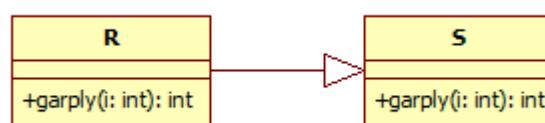
Mark 1.00 out of 1.00

Flag question

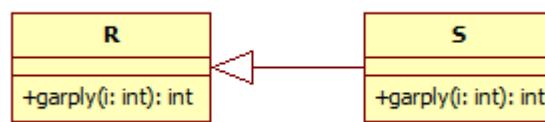
Tudjuk, hogy az **R** és **S** osztályok között öröklődés van, és nem sértik a Liskov-elvet. Rajzolja be a közöttük lévő öröklődést, ha tudjuk, hogy **S garply()** függvényének prekondíciói erősebbek, mint **R garply()** függvényének prekondíciói.



-



-



The correct answer is:



Objektumorientált szoftvertervezés - BMEVIIIAC00

2020/21/1

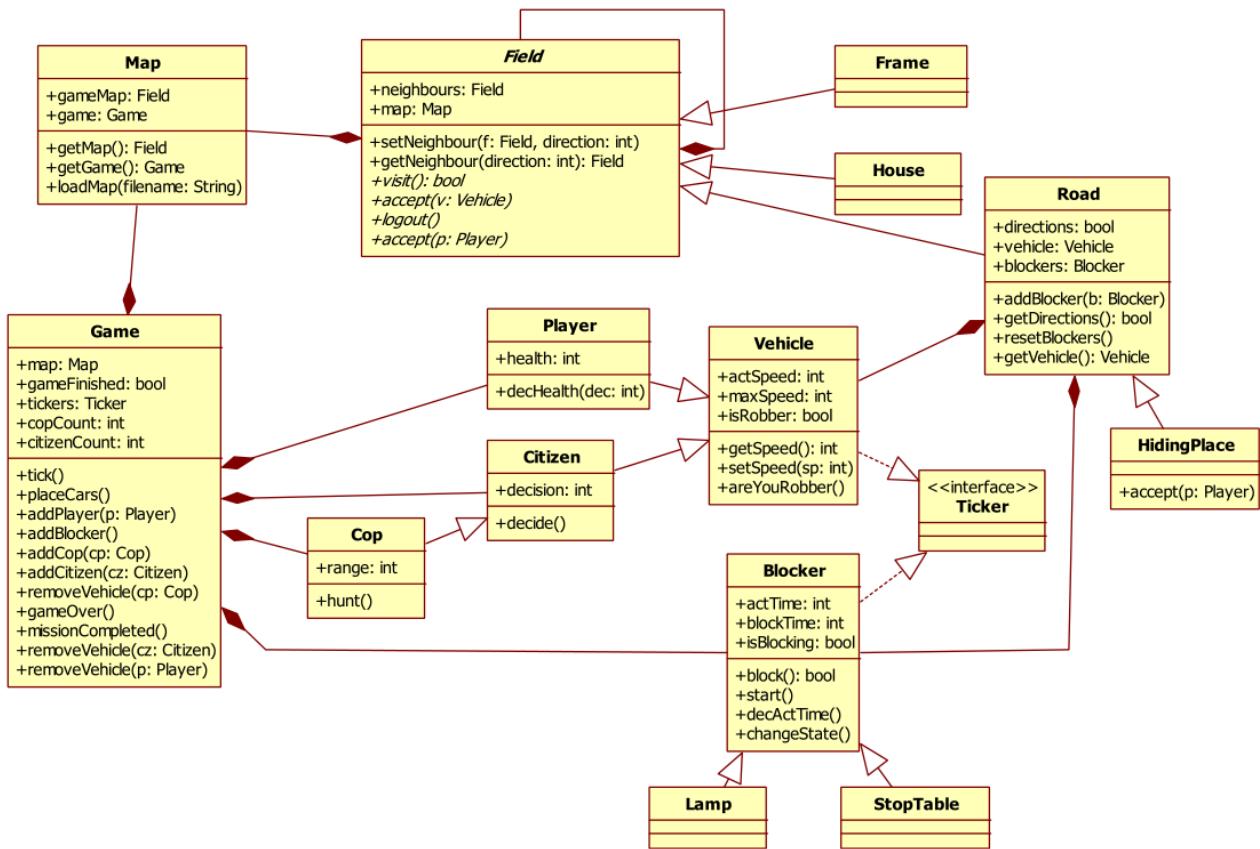
Question 1

Complete

Mark 1.00 out of 1.00

Remove flag

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely osztályok sértik meg a következő OO tervezési heurisztikát?

"Kerüljük az isten-osztályokat!"

- Blocker
- Ticker
- Lamp

- Field
- HidingPlace
- Vehicle
- Game
- Player
- House
- Citizen
- Cop
- Map
- Frame
- StopTable
- Road

The correct answer is:

Game

Question **2**

Complete

Mark 2.00 out of 5.00

 Flag question

Párosítsa az alábbi OO tervezési heurisztikákat a nekik leginkább megfelelő tervezési mintákkal!

Az öröklési hierarchia gyökerében interfészek vagy absztrakt osztályok legyenek!

Abstract factory

A tartalmazott objektumok ne beszélgeszenek egymással közvetlenül!

Visitor

A tartalmazó objektum használja a tartalmazott objektumokat!

Decorator

A statikus szemantikát és kényszereket a modell struktúrájába építük be!

Mediator

Soha ne vizsgáljuk egy objektum típusát, használunk helyette polimorfizmust!

Interpreter

The correct answer is: Az öröklési hierarchia gyökerében interfések vagy absztrakt osztályok legyenek!

- Abstract factory, A tartalmazott objektumok ne beszélgettessék egymással közvetlenül!
- Mediator, A tartalmazó objektum használja a tartalmazott objektumokat!
- Decorator, A statikus szemantikát és kényszereket a modell struktúrájába építük be!
- Interpreter, Soha ne vizsgáljuk egy objektum típusát, használunk helyette polimorfizmust!
- Visitor

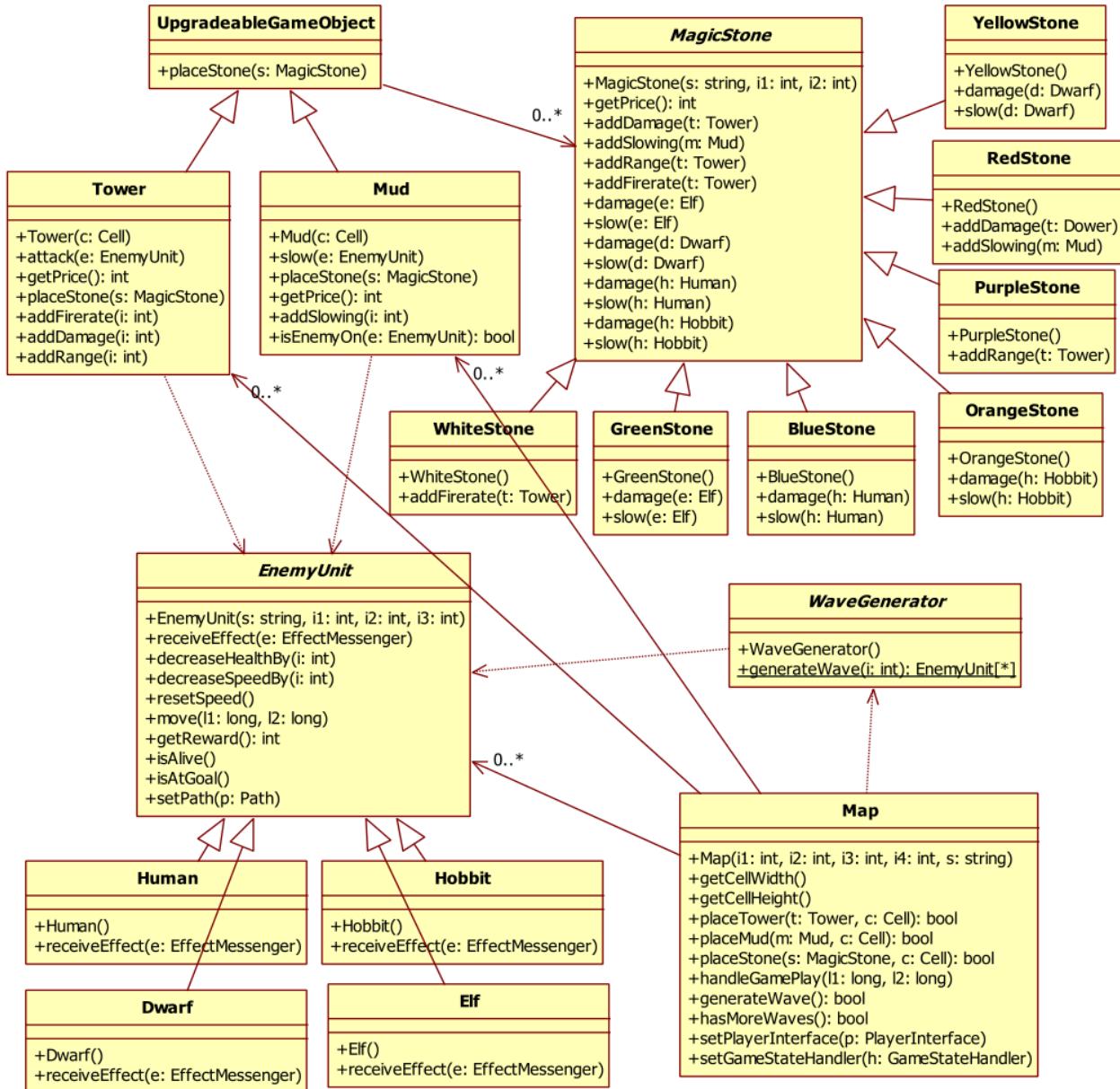
Question **3**

Complete

Mark 1.00 out of 1.00

 Remove flag

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely OO tervezési heurisztikák teljesülnek az alábbiak közül?

- Egy osztály ne függön a saját leszármazottaitól!
- Implementáljuk a sztenderd metódusokat!
- A gyakran változó dinamikus szemantikát és kényszereket külső viselkedésként implementáljuk!
- Soha ne kódoljuk a típust enum vagy int értékekbe, használjunk helyette polimorfizmust!
- Asszociáció helyett preferáljuk a tartalmazást!

The correct answers are:

A gyakran változó dinamikus szemantikát és kényszereket külső viselkedésként implementáljuk!,

Soha ne kódoljuk a típust enum vagy int értékekbe, használunk helyette polimorfizmust!,

Egy osztály ne függjön a saját leszármazottaitól!

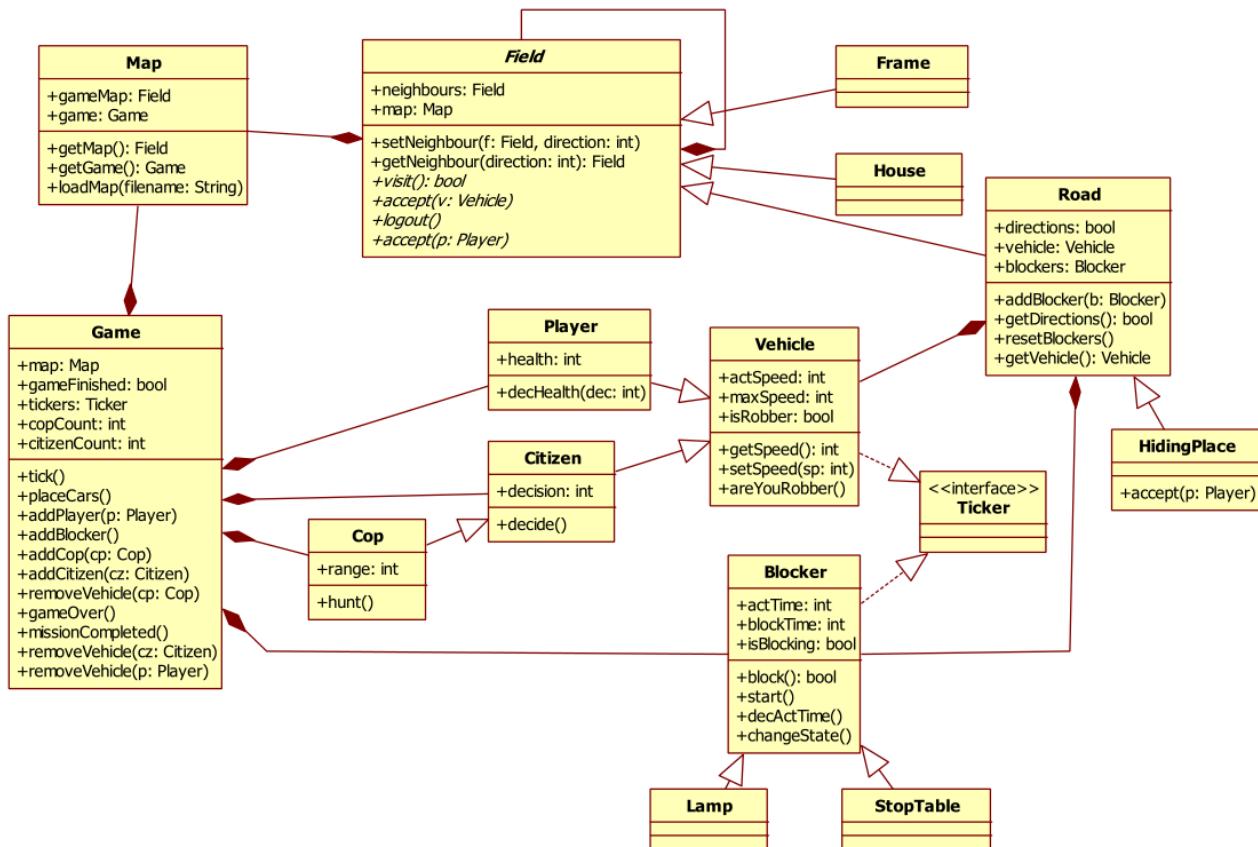
Question 4

Complete

Mark 1.00 out of 1.00

Remove flag

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely osztályok sértik meg a következő OO tervezési heurisztikát?

"Ne készítünk függvényeket a típusok illetve a képességek megkülönböztetésére, használunk helyettük polimorfizmust!"

Map

Road

Cop

Cop

Field

 Lamp Game StopTable Vehicle Blocker Player Ticker Citizen HidingPlace Frame House

The correct answer is:

Vehicle

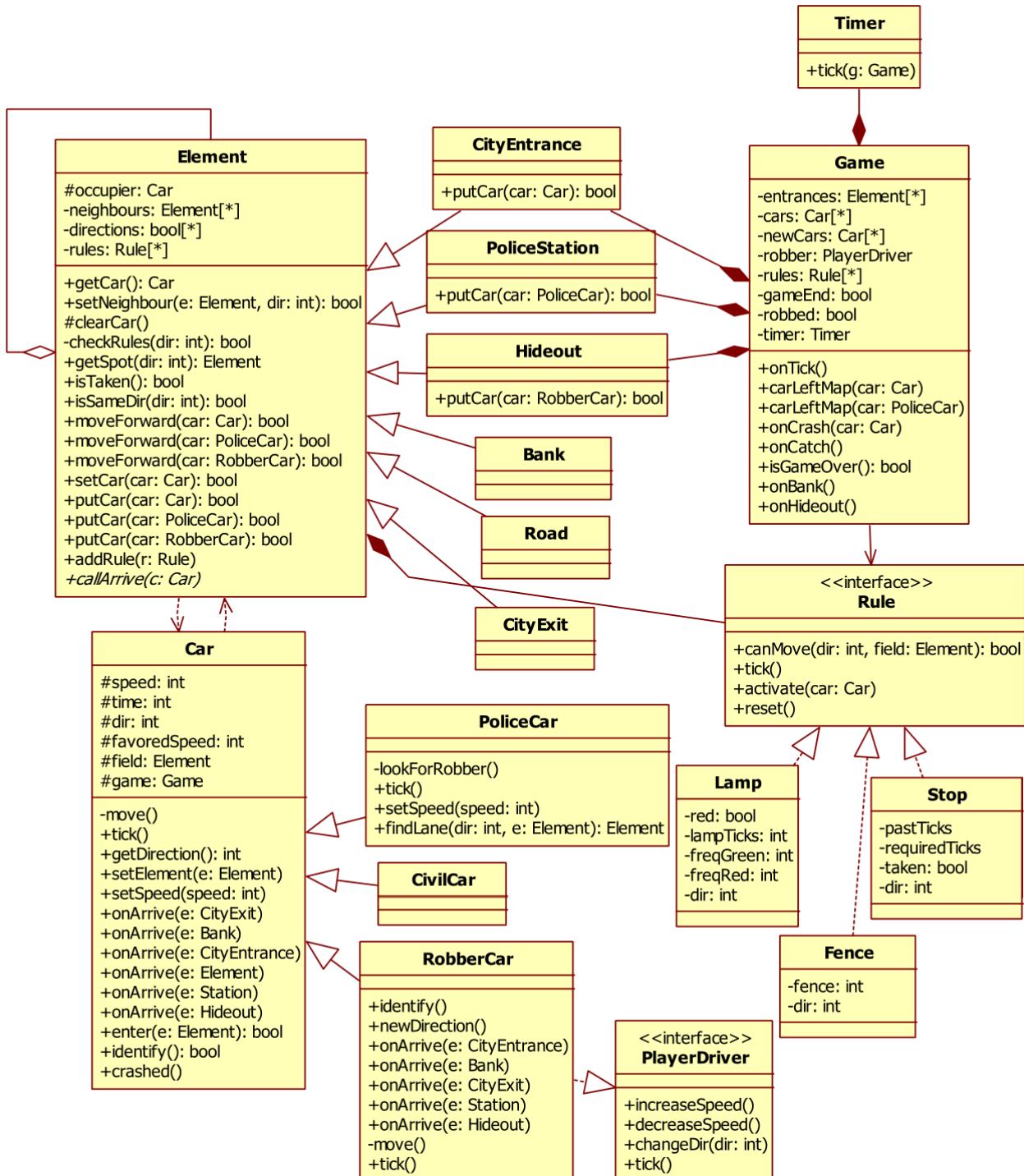
Question **5**

Complete

Mark 1.00 out of 1.00

 Remove flag

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely osztályok sértik meg a következő OO tervezési heurisztikát?

"A viselkedést modellezük, ne a szerepeket!"

- PoliceCar
- CityExit
- Lamp

Element Game Stop RobberCar PlayerDriver Bank Fence Road Timer Rule Car CivilCar Hideout PoliceStation CityEntrance

The correct answer is:

CivilCar

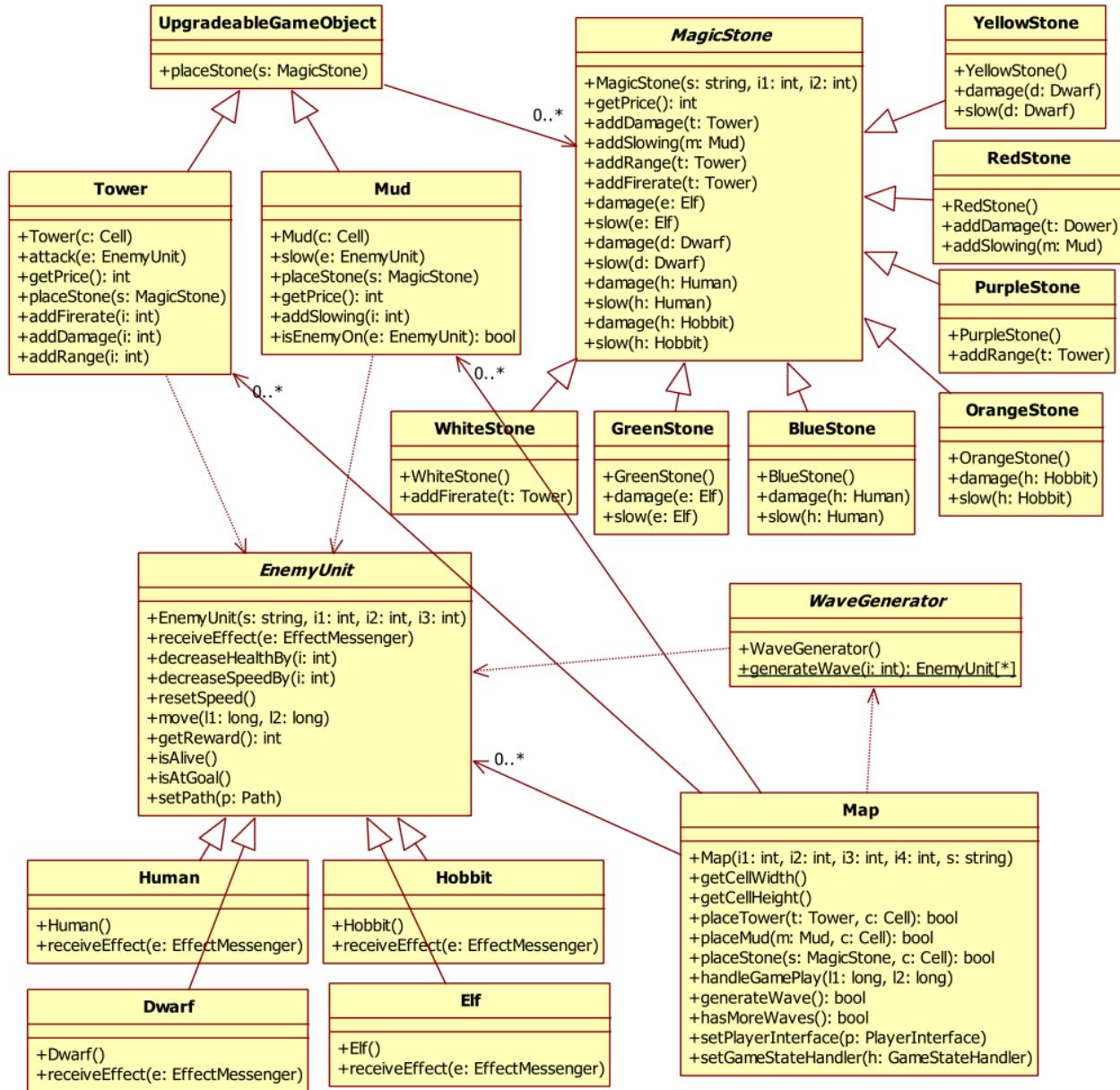
Question **6**

Complete

Mark 1.00 out of 1.00

 Remove flag

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely OO tervezési heurisztikák teljesülnek az alábbiak közül?

- A viselkedést modellezük, ne a szerepeket!
- Ne keverjük össze a leszármazottakat az objektumokkal! Vigyázzunk azokkal a leszármazottakkal, amelyekből csak egyetlen példányt hozunk létre!
- Az öröklődés célja mindenkor a viselkedés újrahasznosítása!
- Az öröklési hierarchia gyökerében interfészök vagy absztrakt osztályok legyenek!
- Kerüljük a csak adattárolásra használt osztályokat!

The correct answers are:

Ne keverjük össze a leszármazottakat az objektumokkal! Vigyázzunk azokkal a leszármazottakkal, amelyekből csak egyetlen példányt hozunk létre,

Kerüljük a csak adattárolásra használt osztályokat!,

A viselkedést modellezük, ne a szerepeket!,

Az öröklődés célja mindenkor a viselkedés újrahasznosítása!

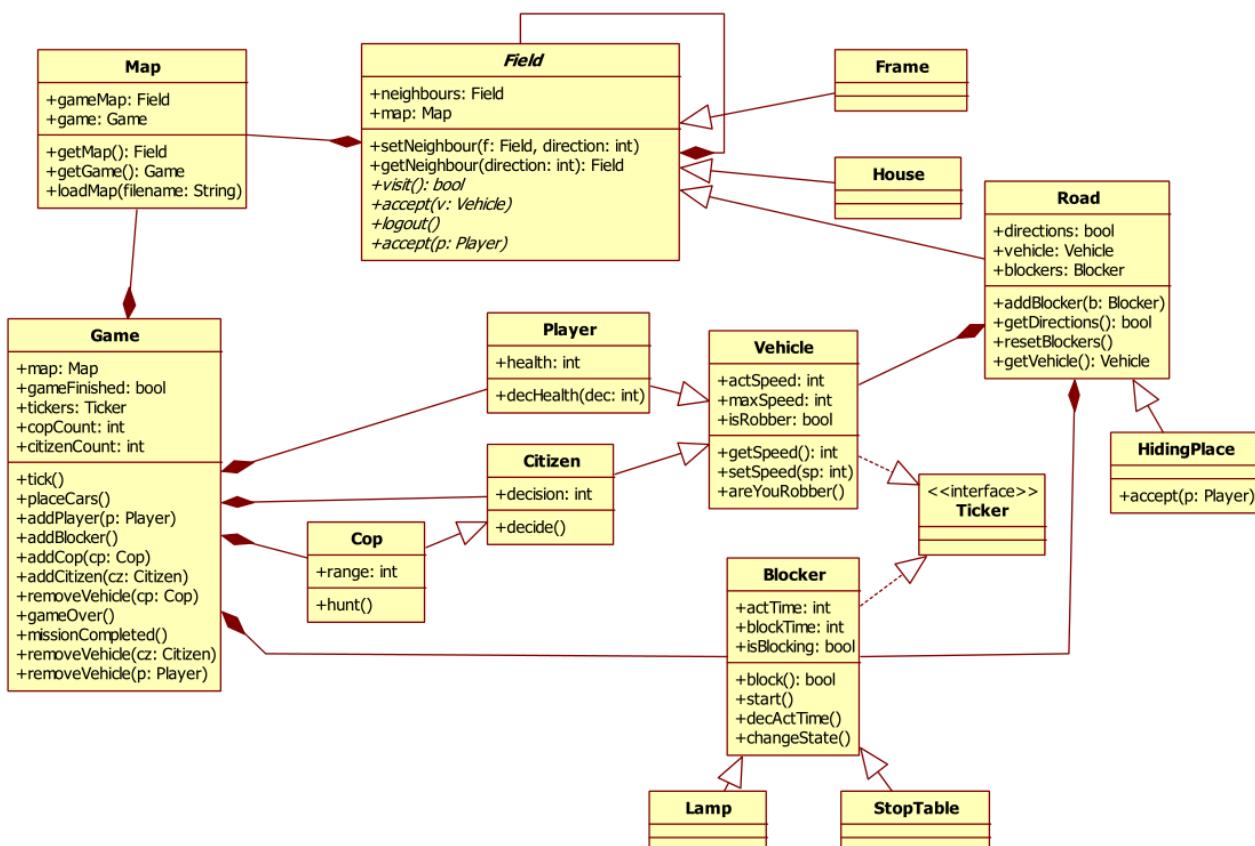
Question 7

Complete

Mark 1.00 out of 1.00

Remove flag

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Az alábbi kijelentések közül melyek igazak?

- A Game és a Map közötti tartalmazás iránya rossz.
- A Map és a Field közötti tartalmazás iránya rossz.

A Lamp és a StopTable működésre vonatkozóan nem különbségei.

A Lamp és a StopTable működése semmiben sem különbözik.



A Frame és a House működése semmiben sem különbözik.

The correct answers are:

A Map és a Field közötti tartalmazás iránya rossz.,

A Lamp és a StopTable működése semmiben sem különbözik.

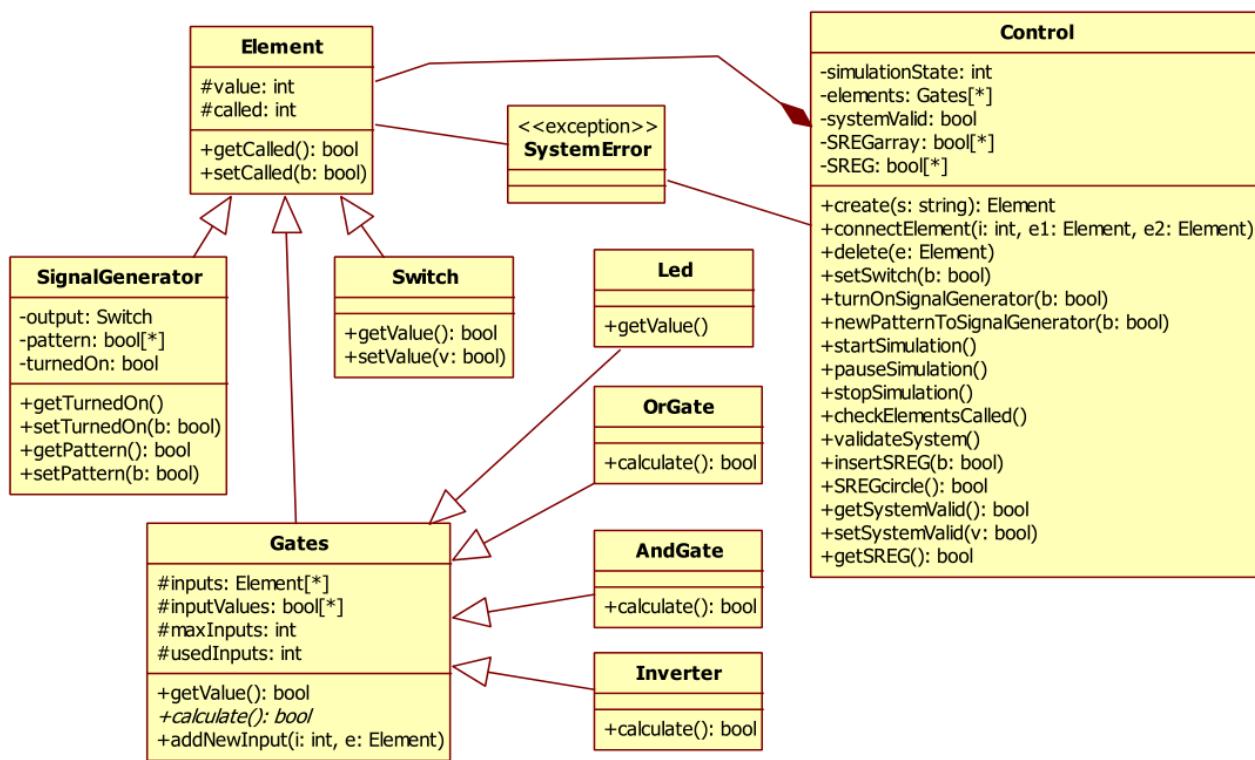
Question 8

Complete

Mark 0.80 out of 1.00

Flag question

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely OO tervezési heurisztikák sérülnek az alábbiak közül?

- Egy osztály csak egy absztrakcióval rendelkezzen!
- A dinamikus szemantikát és kényszereket viselkedésként implementáljuk!
- Ha többszörös öröklésre van szükségünk, gondoljuk át még egyszer a terveket!

- Az összetartozó adatot és viselkedést tartsuk egy helyen!
- Ha reflection-re van szükségünk, modellezzen osztályokat, ne objektumokat!
- Az attribútumok mindenkor legyenek privátok!
- Ne készítsünk függvényeket a típusok illetve a képességek megkülönböztetésére, használunk helyettük polimorfizmust!
- Az öröklési hierarchia gyökerében interfések vagy absztrakt osztályok legyenek!
- Az öröklési hierarchia legyen mély, de legfeljebb hét szintű!

The correct answers are:

Az attribútumok mindenkor legyenek privátok!,

Egy osztály csak egy absztraktcióval rendelkezzen!,

Az összetartozó adatot és viselkedést tartsuk egy helyen!,

Az öröklési hierarchia gyökerében interfések vagy absztrakt osztályok legyenek!

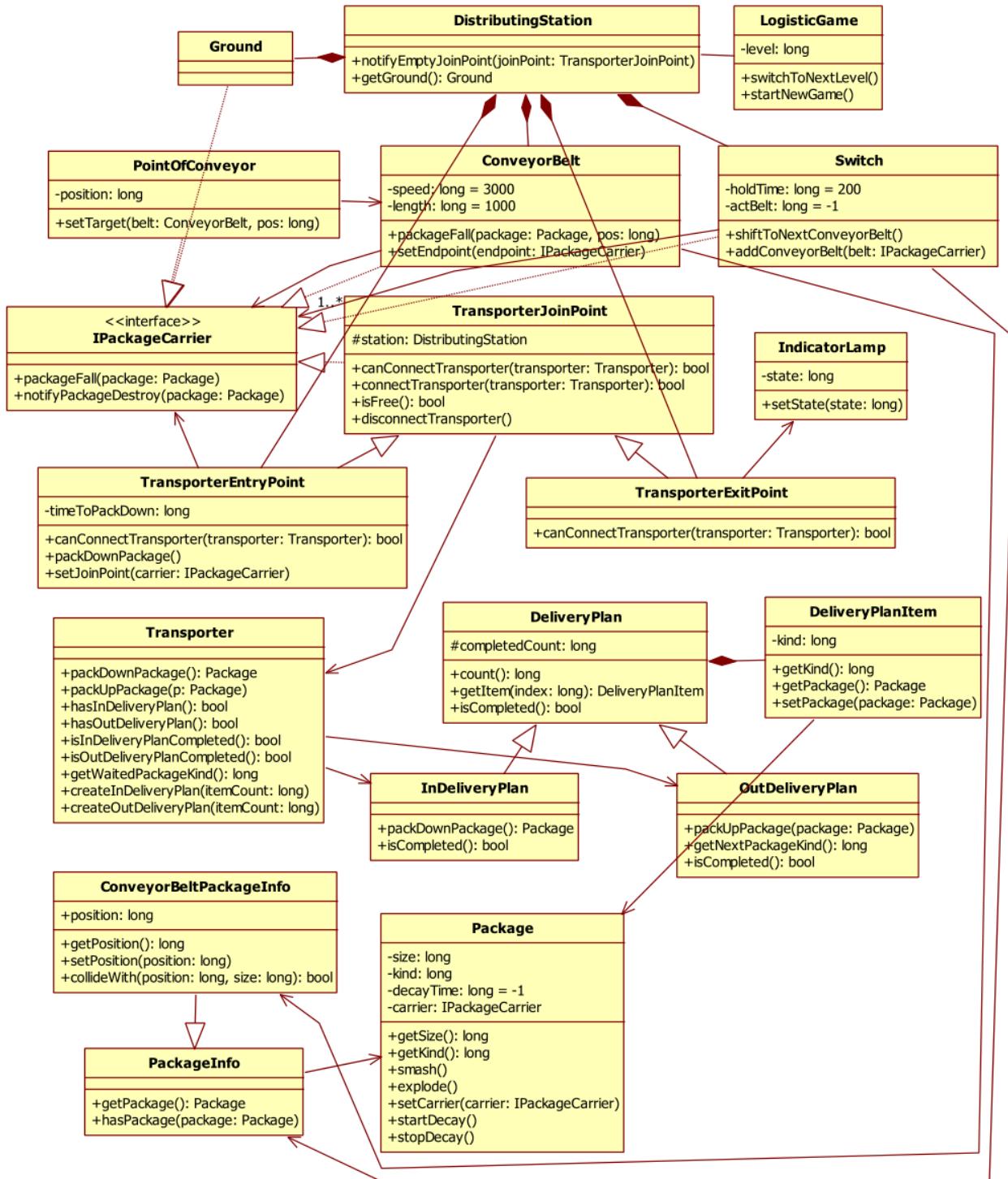
Question **9**

Complete

Mark 0.56 out of 1.00

 Flag question

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely OO tervezési heurisztikák teljesülnek az alábbiak közül?

- Implementáljuk a sztenderd metodusokat!
- Modellezésnél maradjunk a rendszer határain belül!
- Soha ne kódoljuk a típust enum vagy int értékekbe, használjunk helyette polimorfizmust!

Egy osztály ne függjön a saját leszármazottaitól!



A felelősségeket egyenletesen osszuk szét!

- Ne készítsünk függvényeket a típusok illetve a képességek megkülönböztetésére, használunk helyettük polimorfizmust!
- A viselkedést modellezük, ne a szerepeket!
- Az öröklési hierarchia gyökerében interfések vagy absztrakt osztályok legyenek!
- Kerüljük az isten-osztályokat!
- Az öröklődés célja mindig a viselkedés újrahasznosítása!
- Közös interfészt csak akkor valósítsunk meg, ha a viselkedés is közös!

The correct answers are:

Kerüljük az isten-osztályokat!,

A viselkedést modellezük, ne a szerepeket!,

Az öröklődés célja mindig a viselkedés újrahasznosítása!,

A felelősségeket egyenletesen osszuk szét!,

Egy osztály ne függjön a saját leszármazottaitól!,

Ne készítsünk függvényeket a típusok illetve a képességek megkülönböztetésére, használunk helyettük polimorfizmust!,

Soha ne kódoljuk a típust enum vagy int értékekbe, használunk helyette polimorfizmust!,

Közös interfészt csak akkor valósítsunk meg, ha a viselkedés is közös!,

Modellezésnél maradjunk a rendszer határain belül!

Question **10**

Complete

Mark 5.00 out of 5.00

 Remove flag

Párosítsa az alábbi OO tervezési heurisztikákat a nekik leginkább megfelelő OO tervezési elvekkel!

DRY

A közös viselkedés és közös adat minél magasabban legyen az öröklési hierarchiában!

Az öröklődés célja mindenkor a viselkedés újrahasznosítása!

LSP

Kerüljük az isten-osztályokat!

SRP

Ne készítsünk függvényeket a típusok illetve a képességek megkülönböztetésére, használunk helyettük polimorfizmust!

TDA

Egy osztály ne függjen a saját leszármazottaitól!

ADP

The correct answer is: A közös viselkedés és közös adat mindenkor magasabban legyen az öröklési hierarchiában!

- DRY, Az öröklődés célja mindenkor a viselkedés újrahasznosítása!
- LSP, Kerüljük az isten-osztályokat!
- SRP, Ne készítsünk függvényeket a típusok illetve a képességek megkülönböztetésére, használunk helyettük polimorfizmust! → TDA, Egy osztály ne függjen a saját leszármazottaitól!
- ADP

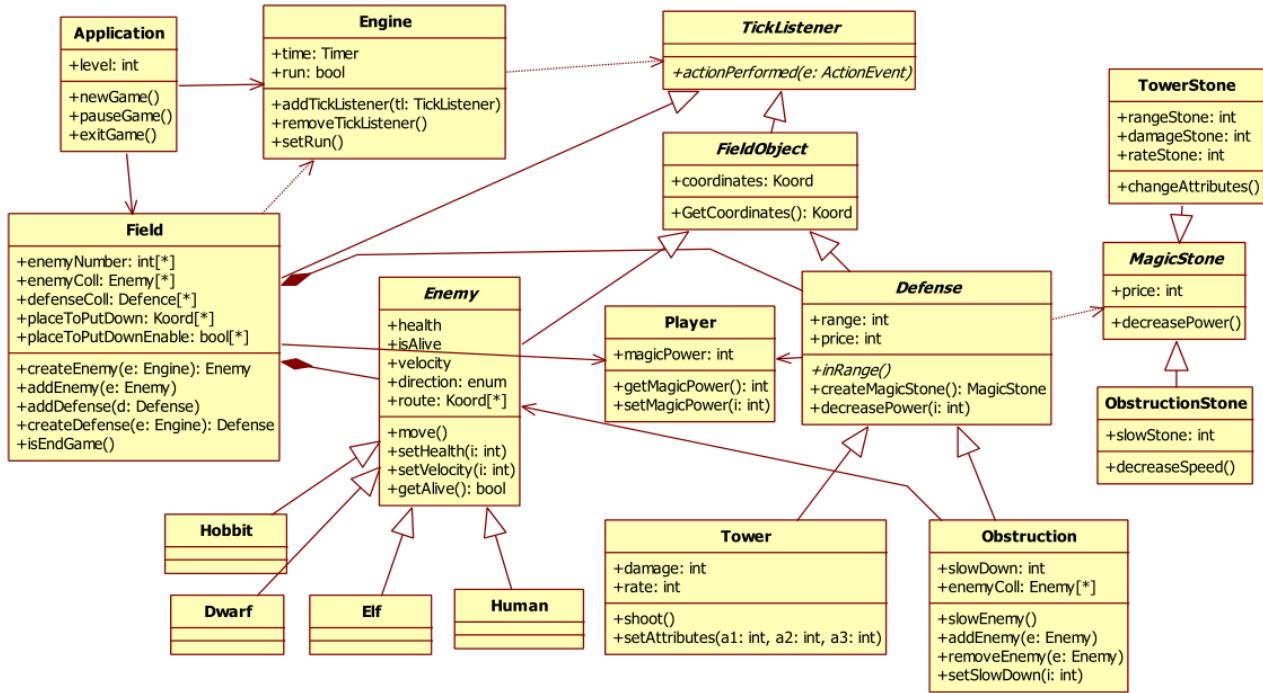
Question 11

Complete

Mark 0.00 out of 1.00

Flag question

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely osztályok sértik meg a következő OO tervezési heurisztikát?

"Mindig a nézet függjön a modelltől, sosem a modell a nézettől!"

- Obstruction
- Elf
- Human
- Tower
- Hobbit
- Field
- MagicStone
- Engine
- Dwarf
- ObstructionStone
- Player
- TowerStone

Application

The correct answers are:

Field,

Hobbit,

Elf,

Dwarf,

Human,

Tower,

Obstruction

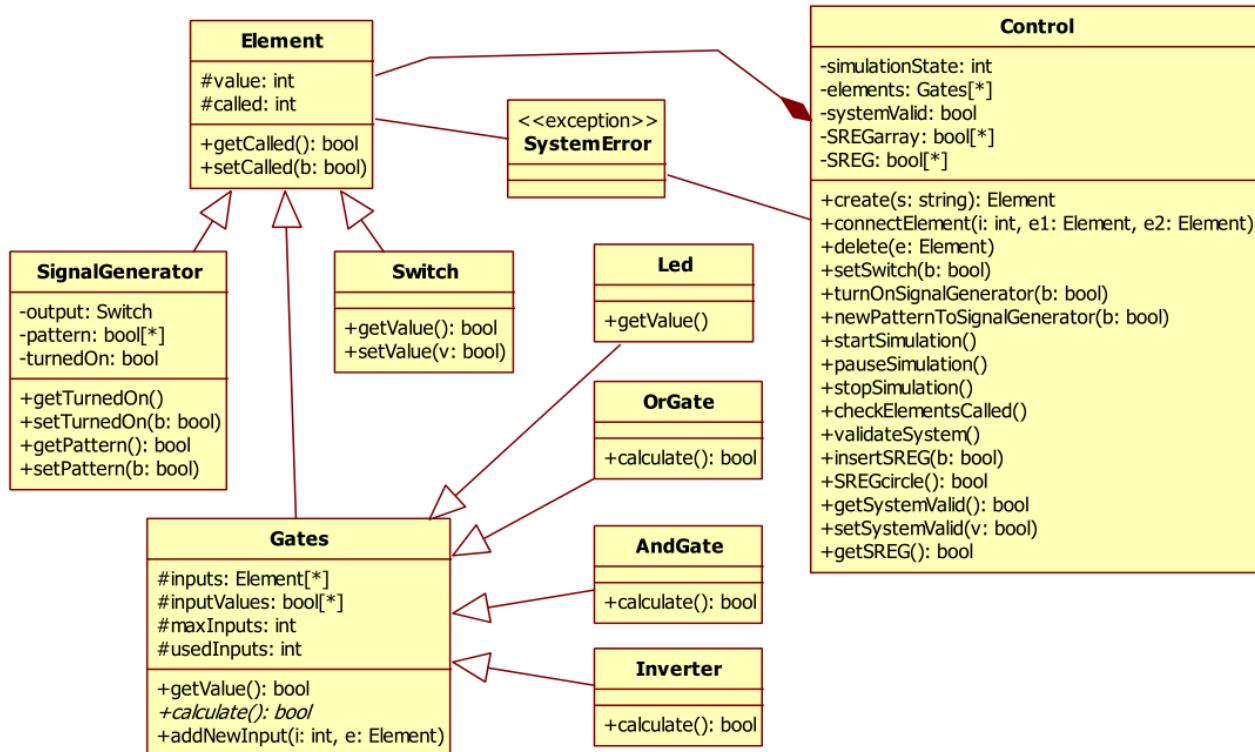
Question 12

Complete

Mark 1.00 out of 1.00

Remove flag

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely osztályok sértik meg a következő OO tervezési heurisztikát?

"A felelősségeket egyenletesen osszuk szét!"

AndGate

Element

 Inverter Gates OrGate Switch Led Control SignalGenerator

The correct answers are:

Element,

SignalGenerator,

Switch,

Gates,

Led,

OrGate,

AndGate,

Inverter,

Control

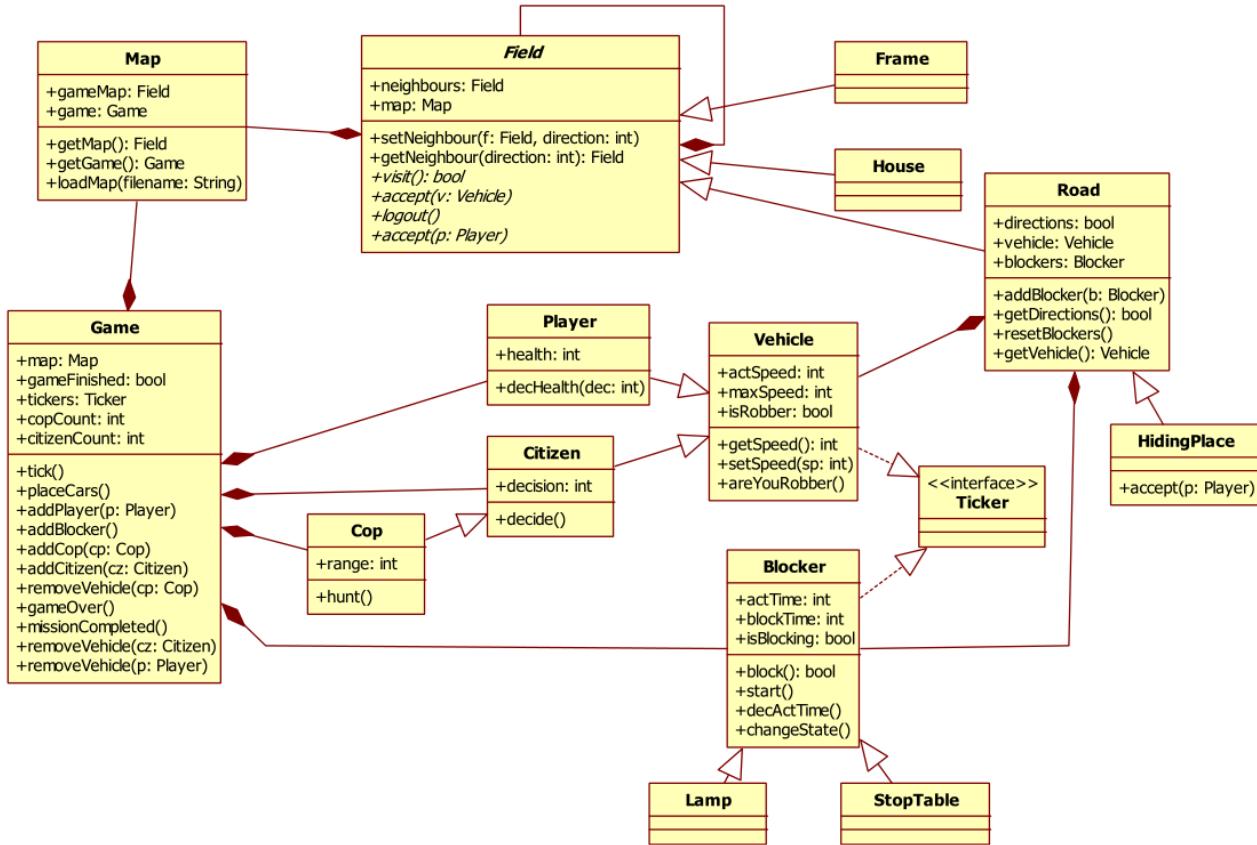
Question 13

Complete

Mark 1.00 out of 1.00

 Remove flag

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely osztályok sértik meg a következő OO tervezési heurisztikát?

"Az attribútumok minden legyenek privátok!"

- Field
- Vehicle
- Player
- House
- StopTable
- Game
- Ticker
- Map
- Blocker
- Road

Lamp Citizen Frame Cop HidingPlace

The correct answers are:

Game,

Map,

Field,

Player,

Cop,

Citizen,

Vehicle,

Blocker,

Road

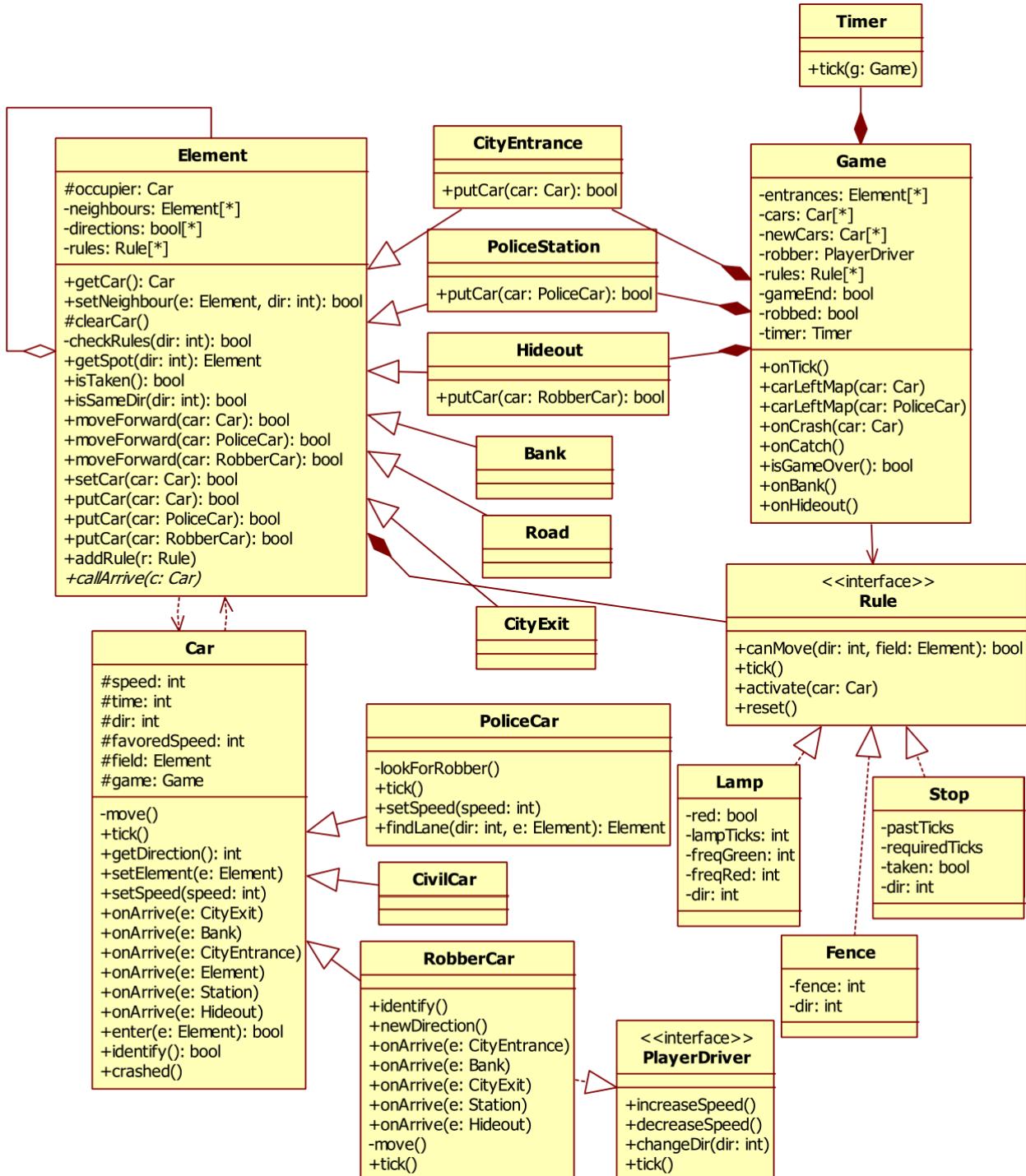
Question 14

Complete

Mark 1.00 out of 1.00

 Remove flag

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Az alábbi kijelentések közül melyek rossz tervezői döntések objektumorientált szempontból?

- A Game nem tárolja az Element-eket.
- Az Element csak függ a Car-tól, nincs rá asszociációja.
- Az Element leszármazottja a Road.
- A Lamp implementálja a Rule interfészt.

The correct answers are:

Az Element csak függ a Car-tól, nincs rá asszociációja.,

A Game nem tárolja az Element-eket.

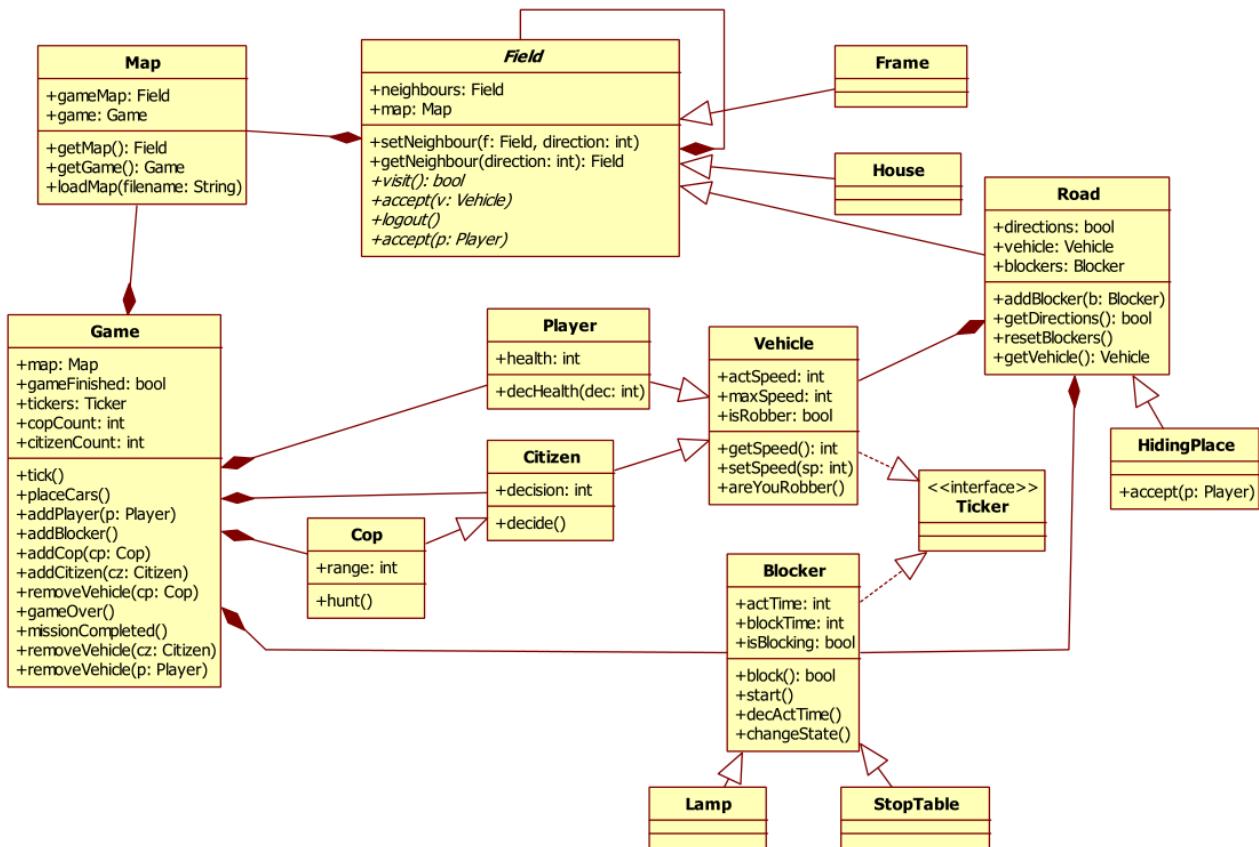
Question 15

Complete

Mark 1.00 out of 1.00

Remove flag

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely osztályok sértik meg a következő OO tervezési heurisztikát?

"A viselkedést modellezük, ne a szerepeket!"

- Player
- HidingPlace
- StopTable

- Ticker
- Frame
- Blocker
- Field
- House
- Vehicle
- Road
- Citizen
- Game
- Map
- Cop
- Lamp

The correct answers are:

Lamp,

StopTable

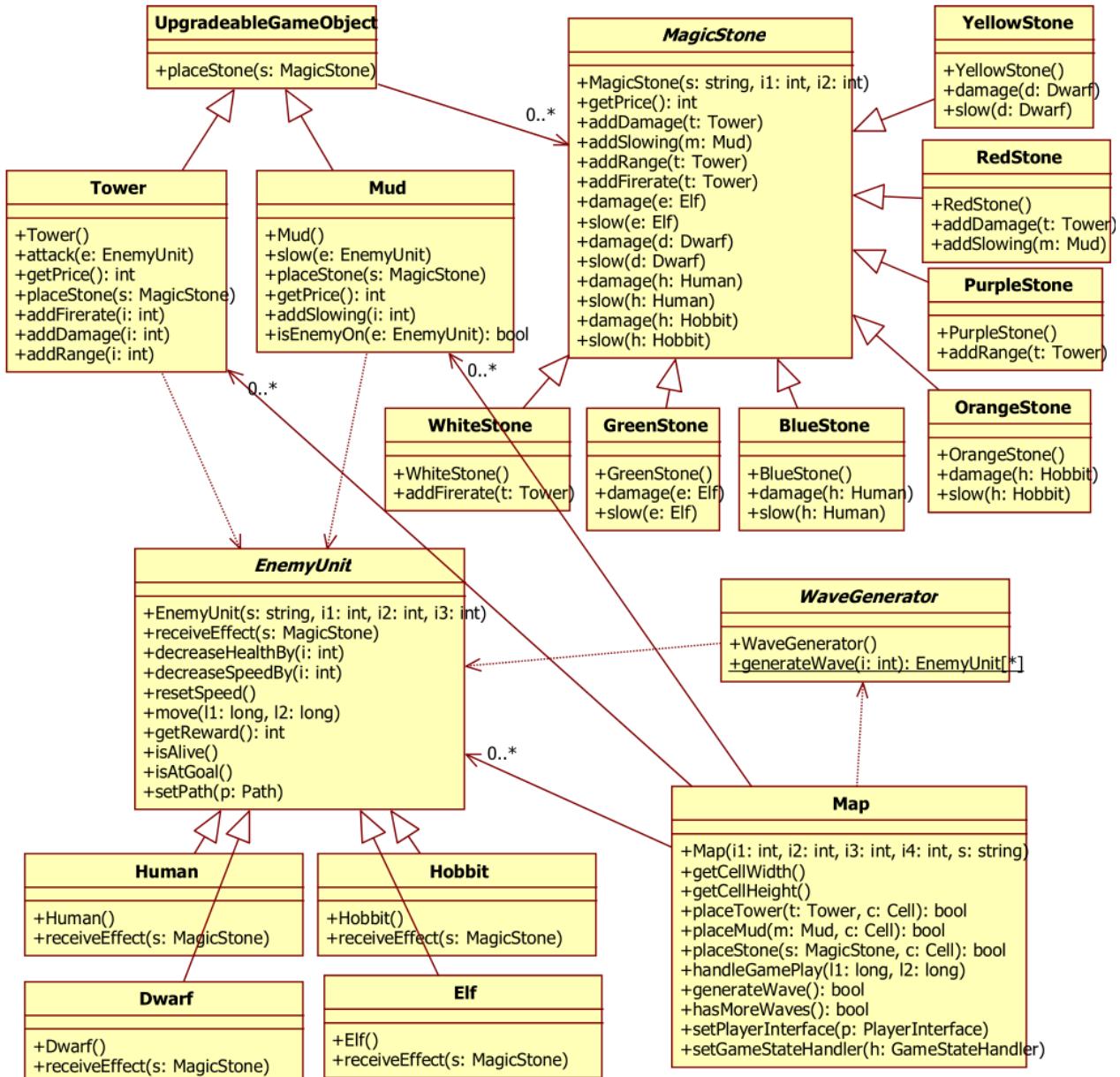
Question **16**

Complete

Mark 0.00 out of 1.00

 Flag question

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely osztályok sértik meg a következő OO tervezési heurisztikát?

"Egy osztály ne függjön az őt használó osztályuktól!"

- UpgradeableGameObject
- Hobbit
- EnemyUnit
- Map
- Elf
- Tower

- Human
- Mud
- WaveGenerator
- Dwarf
- MagicStone

The correct answers are:

Tower,

Mud,

MagicStone,

Human,

Dwarf,

Hobbit,

Elf

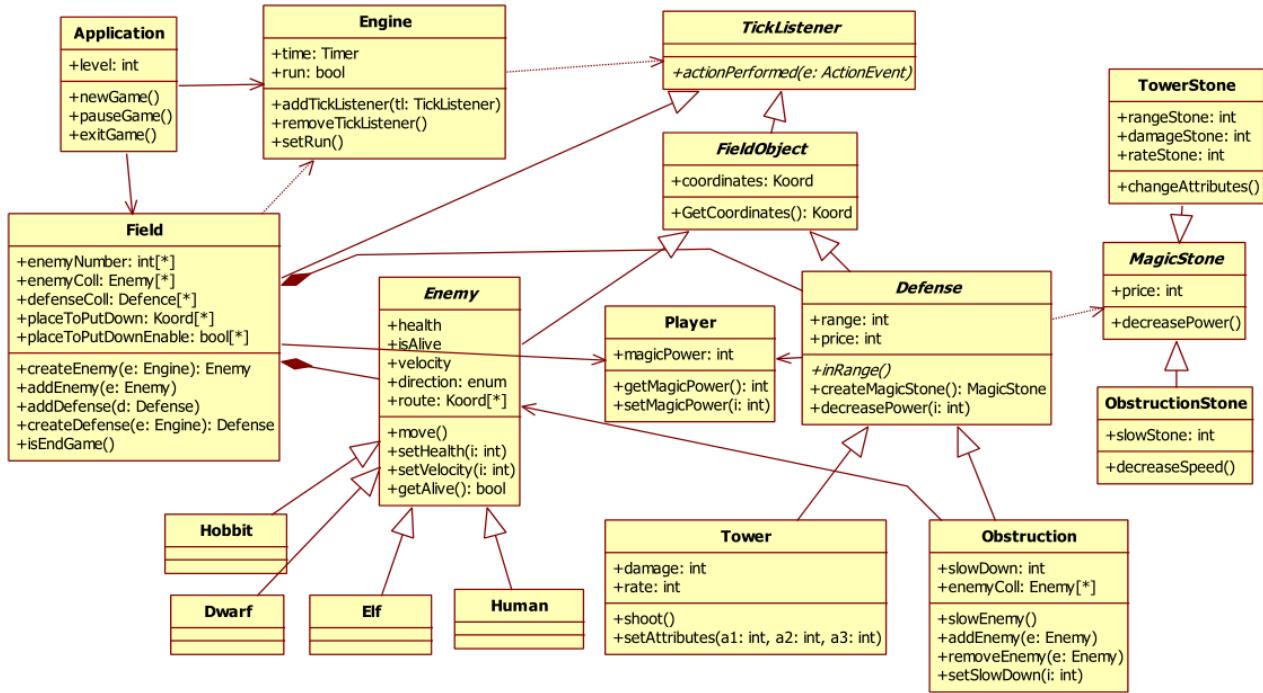
Question **17**

Complete

Mark 0.60 out of 1.00

 Flag question

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely osztályok sértik meg a következő OO tervezési heurisztikát?

"Kerüljük a csak adattárolásra használt osztályokat!"

- Tower
- FieldObject
- TowerStone
- Engine
- Player
- Obstruction
- Defense
- Hobbit
- Human
- ObstructionStone
- Field
- Elf

- Dwarf
- Enemy
- MagicStone
- Application

The correct answer is:

Player

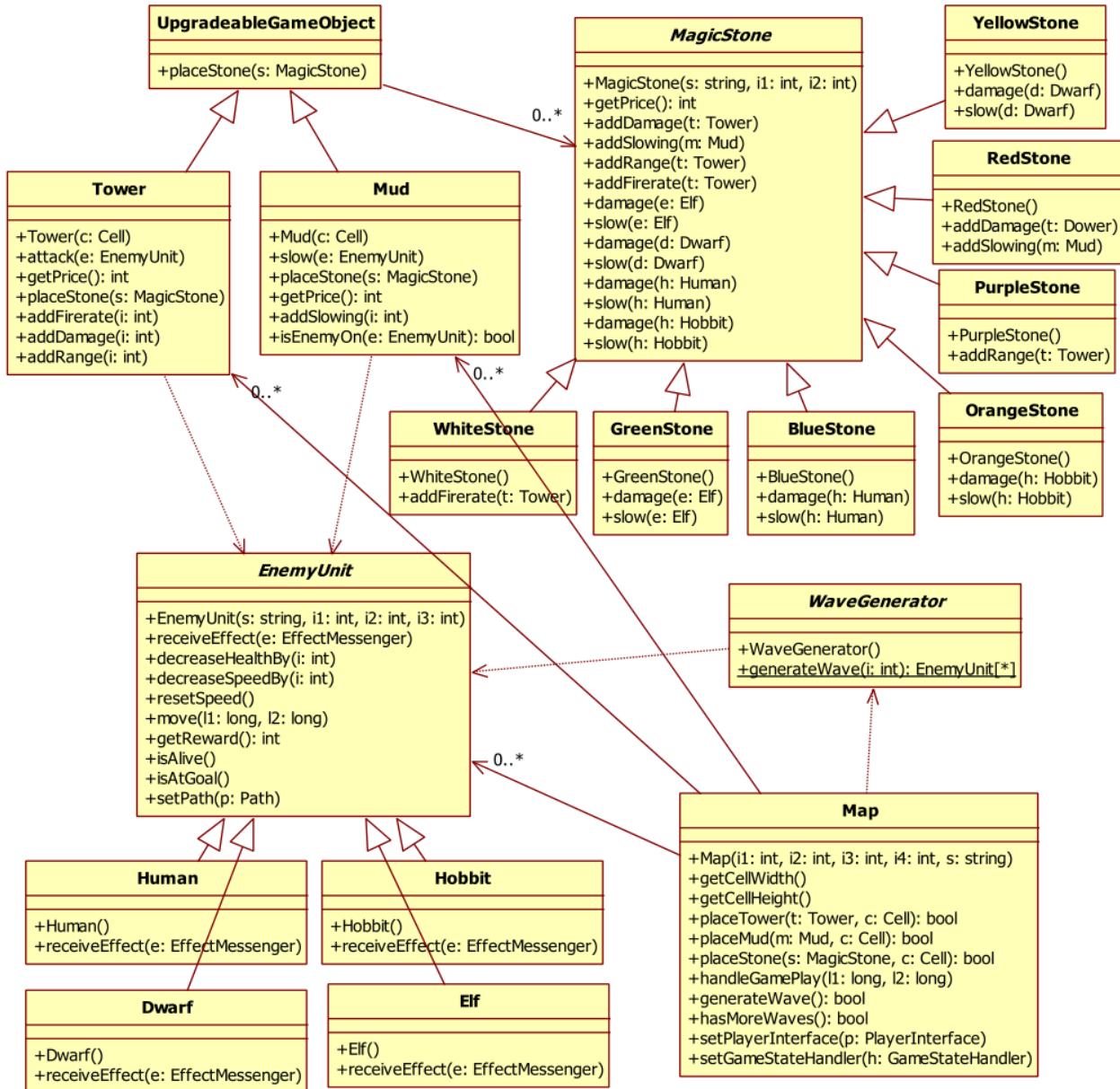
Question **18**

Complete

Mark 1.00 out of 1.00

 Remove flag

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely osztályok teljesítik a következő OO tervezési heurisztikát?

"Az opcionális elemeket tartalmazásként implementáljuk, ne öröklődéssel!"

EnemyUnit

WaveGenerator

Elf

Tower

Human

Mud

- Map
- Hobbit
- Dwarf

The correct answers are:

Tower,

Mud

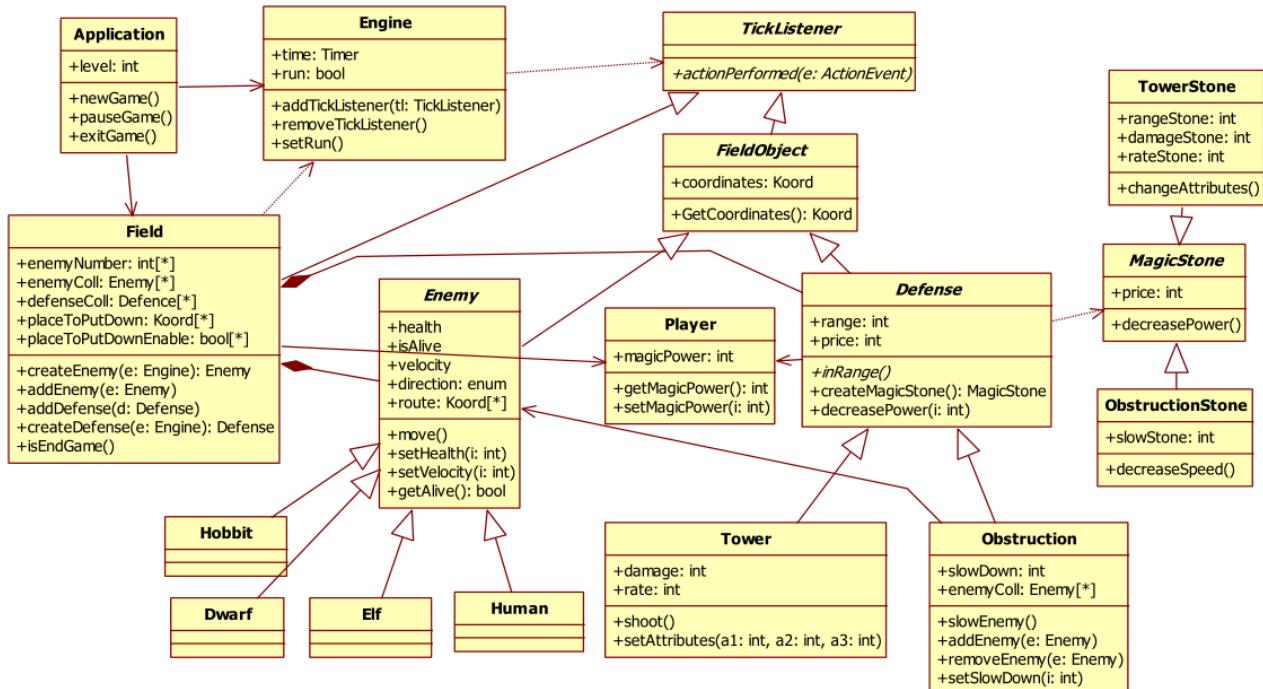
Question 19

Complete

Mark 0.86 out of 1.00

 Flag question

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely OO tervezési heurisztikák teljesülnek az alábbiak közül?

- Az öröklési hierarchia gyökerében interfészok vagy absztrakt osztályok legyenek!
- Ne készítünk függvényeket a típusok illetve a képességek megkülönböztetésére, használjunk helyettük polimorfizmust!

- Implementáljuk a sztenderd metodusokat!
- Az attribútumok mindig legyenek privátok!
- Kerüljük az isten-osztályokat!
- Az öröklési hierarchia legyen mély, de legfeljebb hét szintű!
- Az opcionális elemeket tartalmazásként implementáljuk, ne öröklődéssel!
- Ne keverjük össze a leszármazottakat az objektumokkal! Vigyázzunk azokkal a leszármazottakkal, amelyekből csak egyetlen példányt hozunk létre!
- Egy osztály ne függön a saját leszármazottaitól!

The correct answers are:

Kerüljük az isten-osztályokat!,

Egy osztály ne függön a saját leszármazottaitól!,

Az öröklési hierarchia legyen mély, de legfeljebb hét szintű!,

Az öröklési hierarchia gyökerében interfések vagy absztrakt osztályok legyenek!,

Ne készítsünk függvényeket a típusok illetve a képességek megkülönböztetésére, használunk helyettük polimorfizmust!,

Ne keverjük össze a leszármazottakat az objektumokkal! Vigyázzunk azokkal a leszármazottakkal, amelyekből csak egyetlen példányt hozunk létre!,

Az opcionális elemeket tartalmazásként implementáljuk, ne öröklődéssel!

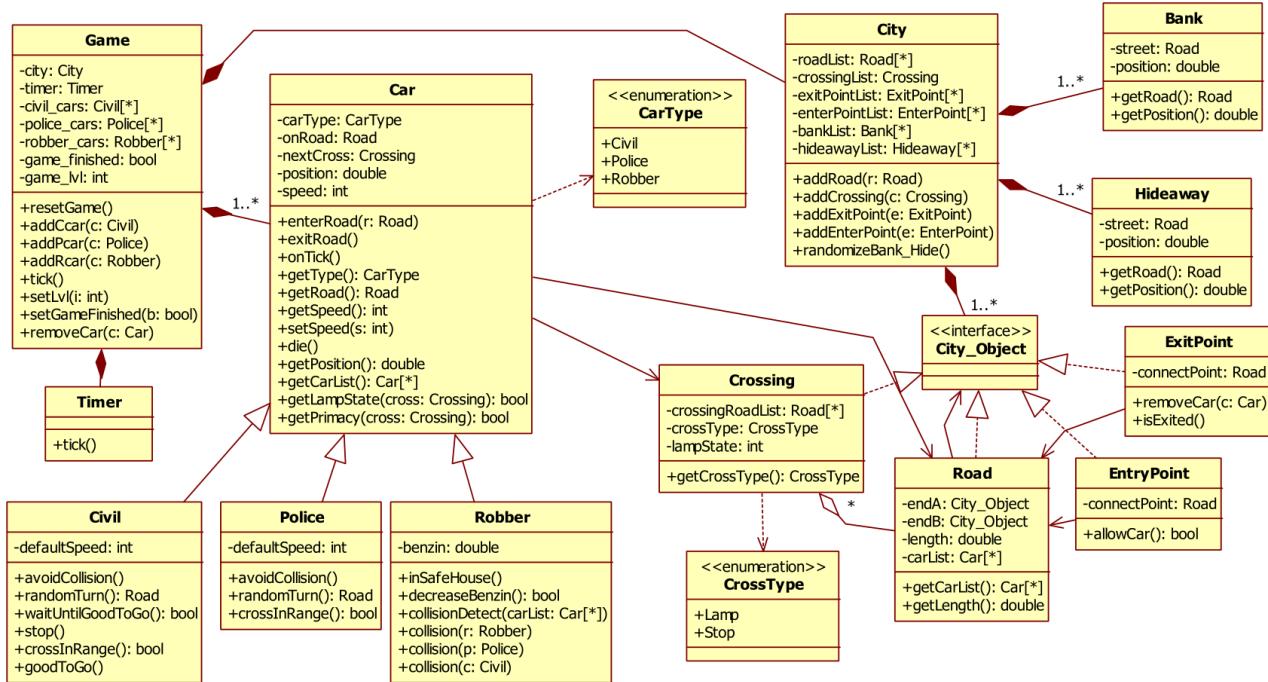
Question **20**

Complete

Mark 1.00 out of 1.00

 Remove flag

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely osztályok sértik meg a következő OO tervezési heurisztikát?

"Soha ne kódoljuk a típust enum vagy int értékekbe, használjunk helyette polimorfizmust!"

- Bank
- Civil
- ExitPoint
- Robber
- Police
- Car
- City
- EntryPoint
- Timer
- Crossing
- Hideaway
- Game

CityObject Road

The correct answers are:

Car,

Crossing

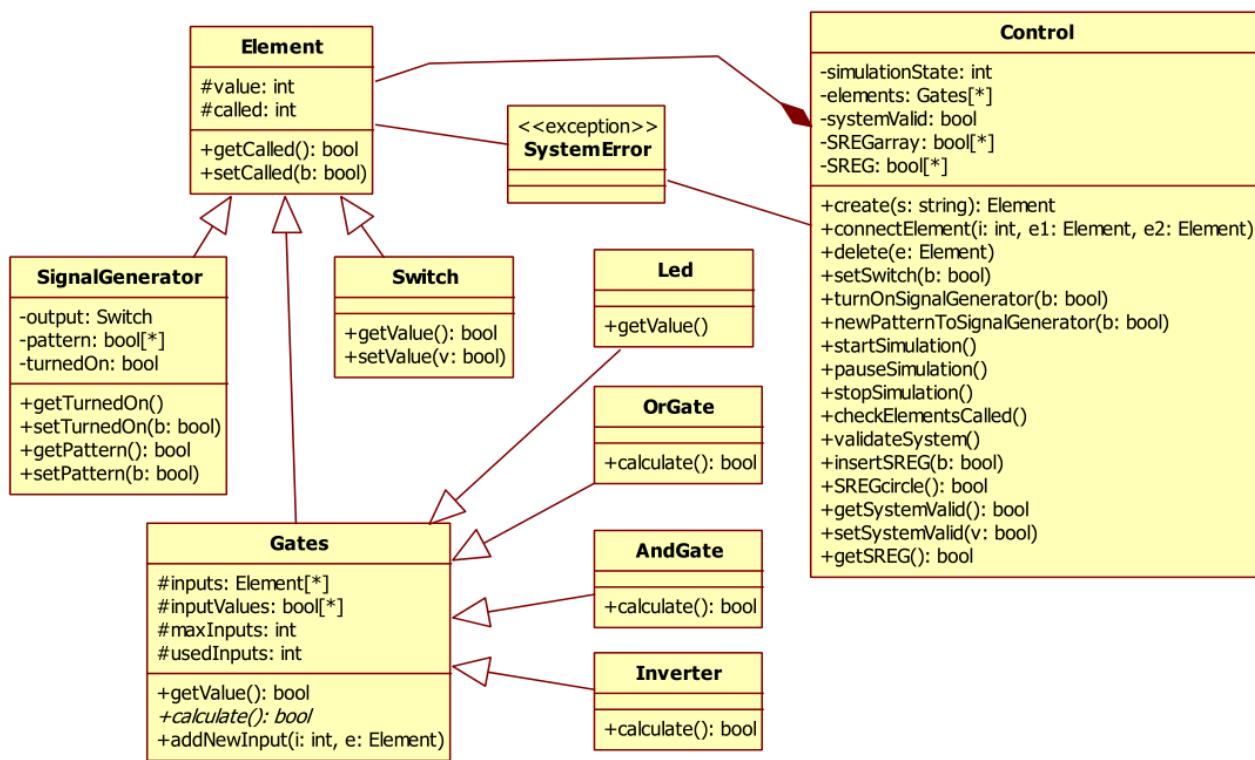
Question 21

Complete

Mark 0.55 out of 1.00

Flag question

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Mely OO tervezési heurisztikák sérülnek az alábbiak közül?

- Kerüljük az isten-osztályokat!
- Közös interfész csak akkor valósítsunk meg, ha a viselkedés is közös!
- A statikus szemantikát és kényszereket a modell struktúrájába építsük be!

- Ne keverjük össze a leszármazottakat az objektumokkal! Vigyázzunk azokkal a leszármazottakkal, amelyekből csak egyetlen példányt hozunk létre!
- Soha ne kódoljuk a típust enum vagy int értékekbe, használjunk helyette polimorfizmust!
- Kerüljük a csak adattárolásra használt osztályokat!
- Modellezsnél maradjunk a rendszer határain belül!
- Modellezzük a valódi világ működését!
- Absztrakt osztályok az öröklési hierarchia gyökerében legyenek!

The correct answers are:

Kerüljük az isten-osztályokat!,

Kerüljük a csak adattárolásra használt osztályokat!,

Modellezzük a valódi világ működését!,

Absztrakt osztályok az öröklési hierarchia gyökerében legyenek!

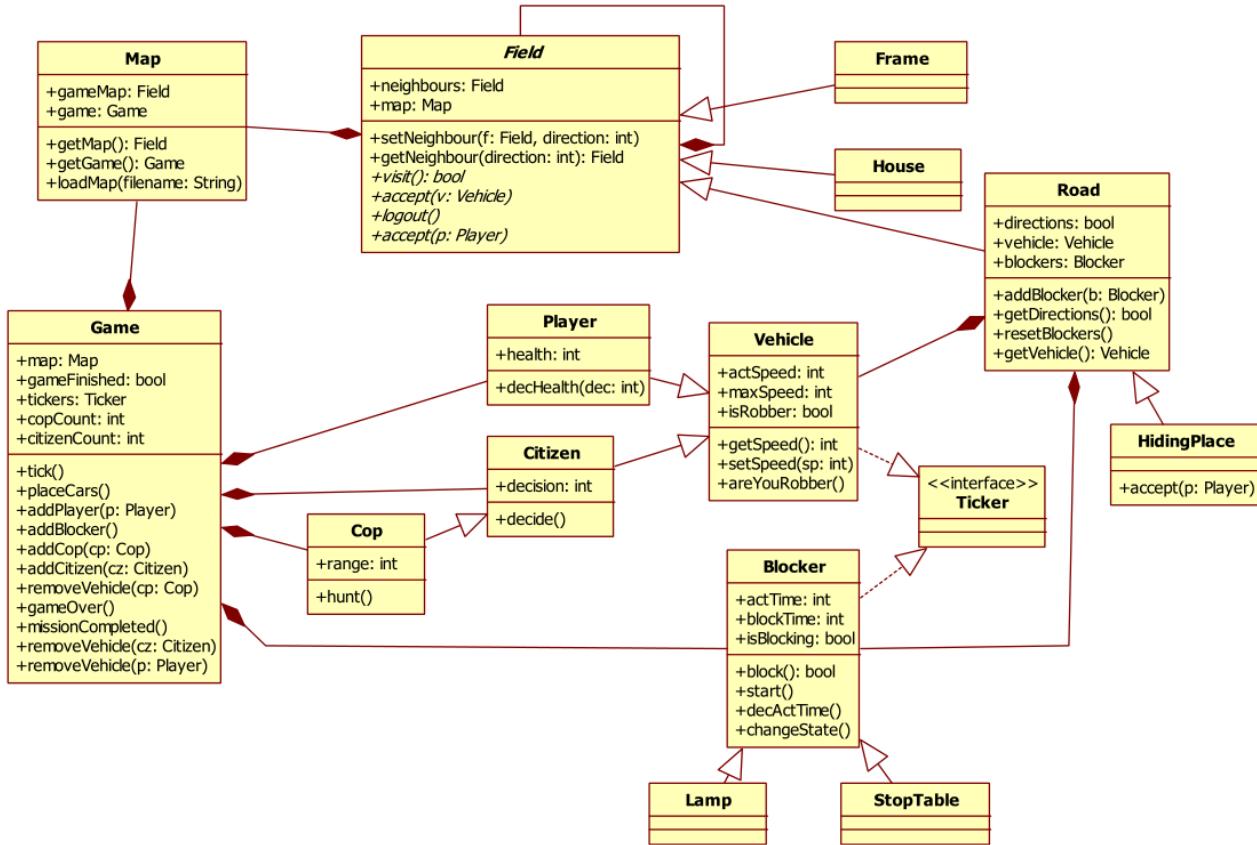
Question **22**

Complete

Mark 0.30 out of 1.00

 Flag question

Tekintsük az alábbi osztálydiagramot (követelmények mutatása/elrejtése):



Az alábbi kijelentések közül melyek rossz tervezői döntések objektumorientált szempontból?

- A Ticker interfész üres.
- A Cop-ot tartalmazza a Game és a Road is.
- A Field leszármazottja a Road és a House is.
- A Game személyesen ismeri a Player, Cop és Citizen osztályokat, ahelyett, hogy egy Vehicle heterogén kollekciót tárolna.
- A Game tartalmazza a Map-et.
- A Field tartalmazza a szomszédait.
- A Blocker-t tartalmazza a Game és a Road is.

The correct answers are:

A Game személyesen ismeri a Player, Cop és Citizen osztályokat, ahelyett, hogy egy Vehicle heterogén kollekciót tárolna.,

A Cop-ot tartalmazza a Game és a Road is.,

A Blocker-t tartalmazza a Game és a Road is.,

A Field tartalmazza a szomszédait.,

A Ticker interfész üres.

Objektumorientált szoftvertervezés - BMEVIIIAC00 2020/21/1

Question 1

Complete

Mark 4.00 out of 7.00

 Flag question

Párosítsa az alábbi refaktorálási mintákat és code smell-eket a nekik leginkább megfelelő OO tervezési elvekkel!

Refactor: Extract interface

SRP

Smell: Speculative generality

YAGNI

Refactor: Replace type code with class

DRY

Smell: Duplicated code

DRY

Smell: Message chains

LoD

Smell: Large class

SRP

Refactor: Separate domain from presentation

ADP

The correct answer is: Refactor: Extract interface → DIP,

Smell: Speculative generality → YAGNI, Refactor: Replace type code with class → OCP, Smell:

Duplicated code → DRY, Smell: Message chains → LoD, Smell: Large class → SRP, Refactor: Separate domain from presentation → SDP

Question 2

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

A refaktoráláshoz szükség van a kódot megfelelően lefedő tesztekre.

Select one:

True

False

The correct answer is 'True'.

Question **3**

Complete

Mark 6.00 out of 10.00

 Flag question

Párosítsa az alábbi OO tervezési heurisztikákat a nekik leginkább megfelelő refaktorálási mintákkal és code smell-ekkel!

A közös viselkedés és közös adat minél magasabban legyen az öröklési hierarchiában!

Refactor: Pull up method

Kerüljük a csak adattárolásra használt osztályokat!

Smell: Data class

Minimalizáljuk az együttműködő osztályok között használt metódusok számát!

Smell: Feature envy

Az attribútumok minden legyenek privátok!

Refactor: Encapsulate field

Minimalizáljuk a publikus metódusok számát!

Refactor: Hide method

Ne használjuk másik osztály nempublikus tagjait!

Refactor: Encapsulate field

Soha ne kódoljuk a típust enum vagy int értékekbe, használunk helyette polimorfizmust!

Refactor: Replace type code with class

Mindig a nézet függjön a modelltől, **sosem a modell a nézettől!**

Smell: Refused bequest

[Sosem a modell a nézettől.](#)

Ha az ōs működését üres

implementációval írjuk felül, akkor hibás
az öröklési hierarchia!

Smell: Inappropriate intimacy

Az opcionális elemeket tartalmazásként
implementáljuk, ne öröklődéssel!

Refactor: Remove control flag

The correct answer is: A közös viselkedés és közös adat minél magasabban legyen az öröklési hierarchiában! → Refactor: Pull up method,

Kerüljük a csak adattárolásra használt osztályokat! → Smell: Data class, Minimalizáljuk az együttműködő osztályok között használt metódusok számát! → Smell: Feature envy,

Az attribútumok mindenlegyenek privátok! → Refactor: Encapsulate field, Minimalizáljuk a publikus metódusok számát! → Refactor: Hide method, Ne használjuk másik osztály nempublikus tagjait! → Smell: Inappropriate intimacy, Soha ne kódoljuk a típust enum vagy int értékekbe, használunk helyette polimorfizmust! → Refactor: Replace type code with class,

Mindig a nézet függjön a modelltől, sosem a modell a nézettől! → Refactor: Separate domain from presentation, Ha az ōs működését üres implementációval írjuk felül, akkor hibás az öröklési hierarchia! → Smell: Refused bequest,

Az opcionális elemeket tartalmazásként implementáljuk, ne öröklődéssel! → Refactor: Introduce null object

Question 4

Complete

Mark 1.00 out of 1.00

Flag question

Igaz vagy hamis a következő állítás?

A refaktorálás során érdemes egyszerre több refaktorálási műveletet is elvégezni, hogy időt spóroljunk.

Select one:

- True
 False

The correct answer is 'False'.

Question 5

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

A refaktorálás javítja a szoftver struktúráját.

Select one:

 True False

The correct answer is 'True'.

Question 6

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Ha büdös kódot (code smell) találunk a szoftverben, akkor a szoftverben hiba (bug) van.

Select one:

 True False

The correct answer is 'False'.

Question 7

Complete

Mark 6.00 out of 7.00

 Flag question

Párosítza az alábbi refaktorálási mintákat a nekik leginkább megfelelő code smell-ekkel!

Hide delegate

Message chains

Preserve whole object

Long parameter lists

Extract method

Duplicated code

Inline class

Lazy class

Decompose conditional

Speculative generality

Replace magic numbers with symbolic constant

Shotgun surgery

Replace array with object

Primitive obsession

The correct answer is: Hide delegate → Message chains,

Preserve whole object → Long parameter lists, Extract method → Duplicated code, Inline class → Lazy class, Decompose conditional → Switch statement, Replace magic numbers with symbolic constant → Shotgun surgery, Replace array with object → Primitive obsession

Question **8**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

A refaktorálás feladata a kódban lévő hibák (bugok) javítása.

Select one:

True

False

The correct answer is 'False'.

Question **9**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

A refaktorálási műveletek között vannak olyanok, amelyek egymás inverzei.

Select one:

 True False

The correct answer is 'True'.

Objektumorientált szoftvertervezés - BMEVIIIAC00 2020/21/1

Question 1

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Egy hosszú magyarázó név jobb, mint egy hosszú magyarázó komment.

Select one:

True

False

The correct answer is 'True'.

Question 2

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kódrészlet megfelel a clean code elveknek:

```
public static bool Precedes(Account first, Account second)
{
    return first.Id < second.Id;
}
```

Select one:

True False

The correct answer is 'True'.

Question 3

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

A publikus API-t mindig el kell látni dokumentációs kommentekkel, de a belső implementációnál elég, ha követjük a clean code elveket.

Select one:

 True False

The correct answer is 'True'.

Question 4

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

A kivételek típusait a hívott objektum szemszögéből kell definiálni.

Select one:

 True False

The correct answer is 'False'.

Question **5**

Complete

Mark 0.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kódrészlet megfelel a clean code elveknek a hibakezelés szempontjából:

```
public int Max(List<int> values)
{
    if (values == null) throw new ArgumentNullException(nameof(values));
    if (values.Count == 0) throw new ArgumentException("At least one value must be specified");
    int result = values[0];
    for (int i = 1; i < values.Count; i++)
    {
        if (values[i] > result) result = values[i];
    }
    return result;
}
```

Select one:

- True
- False

The correct answer is 'True'.

Question **6**

Complete

Mark 1.00 out of 1.00

 Flag question

Hány paramétere legyen egy függvénynek a clean code elvek szerint?

- Maximum 5

- Maximum 3
- Tetszőlegesen sok
- Minél kevesebb

The correct answers are: Minél kevesebb, Maximum 3

Question **7**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Egy rövid név jobb, mint egy hosszú név, feltéve, hogy egyértelmű a jelentése.

Select one:

- True
- False

The correct answer is 'True'.

Question **8**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kódrészlet megfelel a clean code elveknek a visszatérési értékek szempontjából:

```
public List<Book> FindBooksByName(List<Book> books, string name)
{
    if (name == null) return null;
```

```
List<Book> result = new List<Book>();  
for (var b: books)  
{  
    if (b.Name.Contains(name)) result.Add(b);  
}  
return result;  
}
```

Select one:

- True
 False

The correct answer is 'False'.

Question **9**

Complete

Mark 1.00 out of 1.00

 Flag question

Egy metódus paramétereinek számát úgy csökkentettük, hogy néhány paramétert tagváltozóvá tettünk az osztályban. Milyen eseteket rontottunk el az alábbiak közül?

- Ha a metódust több szál is hívhatja párhuzamosan.
 Ha a metódus ciklusban dolgozta fel a paramétert.
 Ha a metódus rekurzív függvény volt.
 Ha a metódus továbbadta a paramétert egy privát metódusnak.

The correct answers are: Ha a metódus rekurzív függvény volt., Ha a metódust több szál is hívhatja párhuzamosan.

Question **10**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kódrészlet .NET környezetben megfelel a clean code elveknek:

```
public interface IPerson
{
    int Age { get; }
    string Name { get; }
}
```

Select one:

True

False

The correct answer is 'True'.

Question **11**

Complete

Mark 1.00 out of 1.00

 Flag question

Egy függvény visszatérési típusa lista. Mi ilyenkor a NullObject mintát követő visszatérési érték?

- "null" érték
- Egyelemű lista.
- Üres lista.

The correct answer is: Üres lista.

Question **12**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kódrészlet megfelel a clean code elveknek:

```
// TODO(https://github.com/myproject/issues/36176):  
// Remove when it is provided by the Logger API  
internal string FormatSymbol(Symbol symbol)  
{  
    return this.Format(symbol, true);  
}
```

Select one:

- True
 False

The correct answer is 'True'.

Question 13

Complete

Mark 1.00 out of 1.00

 Flag question

Az alábbiak közül melyek jó osztálynevek a clean code elvek szerint?

- Pick
 CP
 Manager
 ColorPicker
 Copy
 Color

The correct answers are: Color, ColorPicker

Question 14

Complete

Mark 1.00 out of 1.00

Flag question

Igaz vagy hamis a következő állítás?

Egy objektumorientált programnyelven írt alkalmazásban nincs helye procedurális módon fejlesztett adatszerkezeteknek.

Select one:

 True False

The correct answer is 'False'.

Question 15

Complete

Mark 1.00 out of 1.00

Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kódrészlet megfelel a clean code elveknek:

```
List<Book> bookSet;
```

Select one:

 True False

The correct answer is 'False'.

Question 16

Complete

Mark 1.00 out of 1.00

 Flag question

Az alábbiak közül melyek a jó kommentek a clean code elvek szerint?

- Publikus API dokumentálása
- TODO kommentek
- Blokkzáró kommentek
- Kikommentezett kód

The correct answers are: TODO kommentek, Publikus API dokumentálása

Question 17

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kódrészlet megfelel a clean code elveknek:

```
int t; // elapsed time in seconds
```

Select one:

- True
- False

The correct answer is 'False'.

Question 18

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kódrészlet megfelel a clean code elveknek a hibakezelés szempontjából:

```
public int Max(List<int> values)
{
    if (values == null) return -1;
    if (values.Count == 0) return -2;
    int result = values[0];
    for (int i = 1; i < values.Count; i++)
    {
        if (values[i] > result) result = values[i];
    }
    return result;
}
```

Select one:

- True
 False

The correct answer is 'False'.

Question **19**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

A csúnya kódért nem csak a fejlesztő felelős.

Select one:

- True
 False

The correct answer is 'False'.

Question **20**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kódrészlet megfelel a clean code elveknek a kivételkezelés szempontjából:

```
public void Save(string message)
{
    try
    {
        file.WriteLine(message);
    }
    catch(Exception ex)
    {
        // Oh, come on!
    }
}
```

Select one:

 True False

The correct answer is 'False'.

Question 21

 Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az adatbázistáblákat reprezentáló osztályokon megsérthatő a Demeter-törvény.

Select one:

 True False

The correct answer is 'True'.

Question **22**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Egy hosszú magyarázó név jobb, mint egy rövid, kódolt név.

Select one:

True

False

The correct answer is 'True'.

Question **23**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kódrészlet Java környezetben megfelel a clean code elveknek:

```
public interface IPerson {  
    int getAge();  
    String getName();  
}
```

Select one:

True

False

The correct answer is 'False'.

Question 24

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kódrészlet megfelel a clean code elveknek a visszatérési értékek szempontjából:

```
public List<Book> FindBooksByName(List<Book> books, string name)
{
    List<Book> result = new List<Book>();
    if (name == null) return result;
    for (var b: books)
    {
        if (b.Name.Contains(name)) result.Add(b);
    }
    return result;
}
```

Select one:

 True False

The correct answer is 'True'.

Question 25

Complete

Mark 1.00 out of 1.00

 Flag question

Az alábbiak közül melyek a rossz kommentek a clean code elvek szerint?

- Napló jellegű kommentek
- Feltűnő banner jellegű kommentek

- Tisztázó kommentek
- Figyelmeztetés a következményekre

The correct answers are: Napló jellegű kommentek, Feltűnő banner jellegű kommentek

Question **26**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kódrészlet megfelel a clean code elveknek:

```
// 2010.04.10. Added by Bob
// 2010.05.03. Modified by Alice
public class Date
{
    public int Year { get; set; }
    public int Month { get; set; }
    public int Day { get; set; }
}
```

Select one:

- True
 False

The correct answer is 'False'.

Question **27**

Complete

Mark 0.50 out of 1.00

 Flag question

Az alábbiak közül melyek jó nevek ciklusváltozónak a clean code elvek szerint?

i bookIdentifier a accountIterator

The correct answers are: i, a

Question 28

Complete

Mark 0.33 out of 1.00

 Flag question

Az alábbiak közül melyek jó függvénynevek a clean code elvek szerint?

 ColorPicker F Pick Copy CP PickColor Manager

The correct answers are: Copy, PickColor, Pick

Question 29

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kódrészlet megfelel a clean code elveknek:

```
public class Book
{
    /// <summary>
    /// Default constructor
    /// </summary>
    public Book()
    {
    }

    /// <summary>
    /// Gets or sets the title.
    /// </summary>
    public string Title { get; set; }

    /// <summary>
    /// Gets or sets the author.
    /// </summary>
    public string Author { get; set; }
}
```

Select one:

- True
 False

The correct answer is 'False'.

Question **30**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Egy reguláris kifejezés magyarázataként elfogadható a komment, de vigyázni kell, hogy ne legyen félrevezető, és hogy minden tartsuk szinkronban a reguláris kifejezés valódi jelentésével.

Select one:

True False

The correct answer is 'True'.

Objektumorientált szoftvertervezés - BMEVIIIAC00 2020/21/1

Question 1

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Elosztott kommunikáció esetén a klienst és a szervert mindig ugyanolyan programnyelven kell megírni.

Select one:

- True
 False

The correct answer is 'False'.

Question 2

Complete

Mark 0.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Egy API tervezése során az API implementációja megelőzi az API használatát bemutató példakódok írását.

Select one:

- True
 False

The correct answer is 'False'.

Question **3**

Complete

Mark 1.00 out of 1.00

 Flag question

Az alábbiak közül mely elemek részei a WSDL-nek?

- types
- rest
- service
- message
- body
- portType
- api
- json
- binding
- header
- envelope

The correct answers are:

types,

message,

portType,

binding,

service

Question 4

Complete

Mark 1.00 out of 1.00

Flag question

Igaz vagy hamis a következő állítás?

Az alábbi C függvények alapján azt mondhatjuk, hogy C fájlkezelő API-ja legalább egy API tervezési elvet megsért.

```
int fscanf(FILE* stream, const char* format, ...);  
size_t fread(void* ptr, size_t size, size_t count, FILE* stream);
```

Select one:

 True False

The correct answer is 'True'.

Question 5

Complete

Mark 1.00 out of 1.00

Flag question

Igaz vagy hamis a következő állítás?

Elvileg egy jól megtervezett API teljes, így minden funkciót biztosít, amire a felhasználónak szüksége van. Ilyen tulajdonságú API-t a felhasználói igények felmérése után a gyakorlatban is könnyű készíteni.

Select one:

 True False

The correct answer is 'False'.

Question 6

Complete

Mark 1.00 out of 1.00

Flag question

Igaz vagy hamis a következő állítás?

Elosztott kommunikáció esetén a paraméterek és visszatérési értékek átküldése shallow-copy másolással történik.

Select one:

True

False

The correct answer is 'False'.

Question **7**

Complete

Mark 2.00 out of 3.00

Flag question

Az alábbiak közül mely szabályok azok, amelyeket az API tervezési elvekben és a Clean code által megfogalmazott elvekben is megtalálunk?

- Kerüljük a mellékhatásokat!
- Kerüljük a rövidítéseket!
- Használjuk a helyi terminológiát!
- Kerüljük a hosszú paraméterlistákat!
- Dokumentáljuk az API-t!
- Használunk unchecked (runtime) kivételeket!
- Vagy öröklésre tervezünk, és dokumentáljuk is, vagy tiltsuk meg az öröklődést!
- Használunk kényelmi függvényeket!

The correct answers are:

Használjuk a helyi terminológiát!,

Kerüljük a rövidítéseket!, Kerüljük a hosszú paraméterlistákat!, Kerüljük a mellékhatásokat!,

Dokumentáljuk az API-t!,

Használunk unchecked (runtime) kivételeket!

Question **8**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi kód egy Word dokumentum mentését végzi a Word programozható API-ján keresztül. A kód alapján azt mondhatjuk, hogy a Word API legalább egy API tervezési elvet megsért.

```
object fileName = @"C:\Test\NewDocument.docx";
document.SaveAs(ref fileName, ref missing, ref missing, ref missing, ref missing, ref missi
    ref missing, ref missing, ref missing, ref missing, ref missing,
    ref missing, ref missing, ref missing, ref missing);
```

Select one:

True

False

The correct answer is 'True'.

Question **9**

Complete

Mark 4.00 out of 5.00

 Flag question

Elosztott kommunikáció során az alábbiak közül melyek jellemzők a Proxy-ra és melyek az Adapter-re?

Szerver objektum példányosítása	Adapter
Függvényhívás hálózati kéréssé alakítása	Proxy
Kliens kapcsolatok fogadása	Adapter
A szerver megkeresése	Proxy
Paraméterek sorosítása	Proxy
Implementálja a szerver interfészét	Proxy
Visszatérési érték sorosítása	Adapter
Beérkező hálózati kérés függvényhívássá alakítása	Proxy
Csatlakozás a szerverhez	Proxy
Úgy viselkedik, mintha lokális szerver objektum lenne	Adapter

The correct answer is:

Szerver objektum példányosítása → Adapter,

Függvényhívás hálózati kéréssé alakítása → Proxy,

Kliens kapcsolatok fogadása → Adapter,

A szerver megkeresése → Proxy,

Paraméterek sorosítása → Proxy,

Implementálja a szerver interfészét → Proxy,

Visszatérési érték sorosítása → Adapter,

Beérkező hálózati kérés függvényhívássá alakítása → Adapter,

Csatlakozás a szerverhez → Proxy,

Úgy viselkedik, mintha lokális szerver objektum lenne → Proxy

Question **10**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

RMI esetén a kliens és a szerver ugyanolyan keretrendszerben (pl. Java vagy .NET) futnak.

Select one:

- True
 False

The correct answer is 'True'.

Question **11**

Complete

Mark 0.00 out of 1.00

 Flag question

A cookie-k milyen állapottárolást tesznek lehetővé?

- Állapot a szerver adatbázisában
 Állapot a kliens oldalon
 Állapot a szerver memóriájában

The correct answer is:

Állapot a kliens oldalon

Question **12**

Complete

Mark 0.00 out of 1.00

 Flag question

REST esetén melyik HTTP metódus szolgál egy erőforrás létrehozására, ha a szerver adja az azonosítót?

Answer: GET

The correct answer is: POST

Question 13

Complete

Mark 1.00 out of 1.00

 Flag question

Mire szolgál a trading service?

- Szolgáltatások logikai név alapján történő keresésére
- Szolgáltatások használati díjának adminisztrálására
- Szolgáltatások interfész alapján történő keresésére
- Tőzsdei szolgáltatások szabványos interfészleírója

The correct answer is:

Szolgáltatások interfész alapján történő keresésére

Question 14

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

REST esetén csak szöveges formátumot (pl. XML, JSON) használhatunk.

Select one:

- True
- False

The correct answer is 'False'.

Question 15

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Egy API tervezésével azért kell körültekintően eljárni, mert egy publikált API-n, amit már mások is használnak, nehéz változtatni.

Select one:

 True False

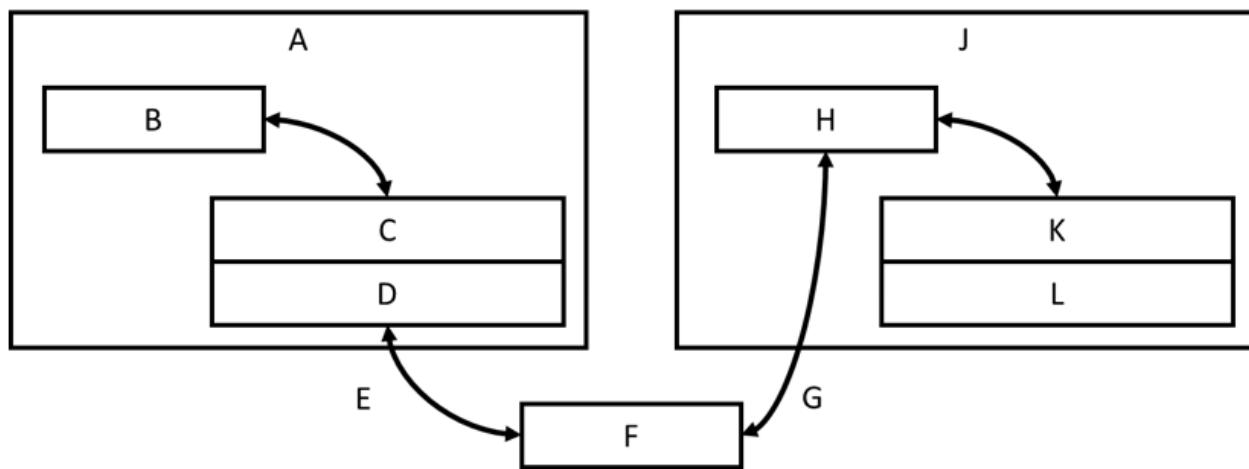
The correct answer is 'True'.

Question 16 Complete

Mark 5.00 out of 5.00

 Flag question

Adja meg az alábbi ábrán látható komponensek neveit az elosztott objektumorientáltság témaábrához kapcsolódóan!



L Szerver implementáció

J Szerver memóriatér

A

- Kliens memóriatér
- C Szerver interfész
- E Sorosítás
- B Kliens objektum
- G Sorosítás
- H Adapter
- D Proxy
- F Hálózat
- K Szerver interfész

The correct answer is: L → Szerver implementáció, J → Szerver memóriatér, A → Kliens memóriatér, C → Szerver interfész, E → Sorosítás, B → Kliens objektum, G → Sorosítás, H → Adapter, D → Proxy, F → Hálózat, K → Szerver interfész

Question 17

Complete

Mark 0.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Amikor a kliens név alapján keresi a szervert, akkor a szerver logikai nevét egy DNS szerver oldja fel.

Select one:

- True
- False

The correct answer is 'False'.

Question 18

Complete

Mark 0.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Az alábbi szabványos CORBA kód egy szervert indít el. A kód alapján azt mondhatjuk, hogy a CORBA API legalább egy API tervezési elvet megsért.

```
public class Server {  
    public static void main(String args[]) {  
        try{  
            // initializing ORB  
            ORB orb = ORB.init(args,new Properties());  
  
            // getting reference to POA  
            org.omg.CORBA.Object obj =  
            orb.resolve_initial_references("RootPOA");  
            POA rootpoa = POAHelper.narrow(obj);  
            // getting reference to POA manager  
            POAManager manager = rootpoa.the_POAManager();  
            // activating manager  
            manager.activate();  
  
            // getting NameService  
            obj = orb.resolve_initial_references("NameService");  
            NamingContextExt ncRef =  
            org.omg.CosNaming.NamingContextExtHelper.narrow(obj);  
  
            // creating servant  
            CountImpl c_impl = new CountImpl();  
            // connecting servant to ORB  
            Count counter = c_impl._this(orb);  
            // binding servant reference to NameService  
            ncRef.rebind(ncRef.to_name("count"), counter);  
  
            System.out.println("Object activated");  
            // starting orb  
            orb.run();  
  
        } catch(Exception e) {  
            e.printStackTrace(System.out);  
        }  
    }  
}
```

Select one:

 True

False

The correct answer is 'True'.

Question 19

Complete

Mark 0.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Egy olyan szerver, amely egyetlen szálon szolgál ki, az nem hatékony, mert a kliensnek meg kell várnia, hogy a szerver az előző kliens kérését kiszolgálja.

Select one:

 True False

The correct answer is 'False'.

Question 20

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

A publish-subscribe modellben egy üzenetet egyszerre csak egy feliratkozó kap meg.

Select one:

 True False

The correct answer is 'False'.

Objektumorientált szoftvertervezés - BMEVIIIAC00 2020/21/1

Question 1

Complete

Mark 0.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

A Strategized locking minta a Strategy tervezési mintát követi.

Select one:

True

False

The correct answer is 'True'.

Question 2

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Readers-writer lock esetén több szál is olvashatja egyszerre az erőforrást.

Select one:

True

False

The correct answer is 'True'.

Question **3**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

A half-sync/half-async minta aszinkron rétege a Reactor és a Proactor minta segítségével is implementálható.

Select one:

- True
 False

The correct answer is 'True'.

Question **4**

Complete

Mark 0.75 out of 1.00

 Flag question

Egy programozási nyelv fejlesztőkörnyezete minden karakterleütésre az aktuális programkód fordítását indítja el egy háttérszálon. Az alábbi minták a hatékony működéshez elengedhetetlenek:

- Starvation
 Cancellation token
 Reactor
 Dead-lock
 GlobalContext

The correct answer is: Cancellation token

Question **5**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

Atomi műveletek kombinációja is atomi művelet.

Select one:

True

False

The correct answer is 'False'.

Question **6**

Complete

Mark 0.00 out of 1.00

 Flag question

Egy többszálú webszerver implementálása során a kéréseket kezelő függvényekben elérhetővé szeretnénk tenni a kliensről szóló információkat, de nem szeretnénk azokat a függvényeknek paraméterként átadni. Melyik minta segíthet a probléma megoldásában az alábbiak közül?

- ThreadLocalContext
- Future
- Asynchronous completion token
- Reactor
- Cancellation token

The correct answer is: ThreadLocalContext

Question **7**

Complete

Mark 1.00 out of 1.00

Flag question

Egy grafikus Windows alkalmazás eseményfeldolgozója az alábbi mintát követi:

- Proactor
- Double-checked locking
- Semaphor
- Reactor
- Monitor object

The correct answer is:

Reactor

Question **8**

Complete

Mark 0.00 out of 1.00

Flag question

Igaz vagy hamis a következő állítás?

A Guarded suspension minta Monitor object használatának segítségével implementálható.

Select one:

- True
- False

The correct answer is 'True'.

Question **9**

Complete

Mark 2.00 out of 9.00

Flag question

Párosítsa az adott feladat leírását az őt megoldó konkurens és elosztott minta nevével!

Egyetlen szál futásának engedélyezése egy művelet eredményének elkészülte után

Proactor

Aszinkron és szinkron feldolgozás szétcsatolása

Future/Task/Deferred

Több szál futásának engedélyezése egy művelet eredményének elkészülte után

Reactor

Több egyidejű olvasás, de csak egyetlen egyidejű írás engedélyezése

Readers-writer lock

Az alkalmazás által kezdeményezett műveletek elkészültét jelző események egyszálú szinkron feldolgozása

Double-checked locking

Egy háttérben futó művelet eredményéről információ szerzése, illetve az eredmény bevárása

Double-checked locking

Szálak egymást váltva dolgozzák fel a beérkező eseményeket

Half-sync/half-async

Szálanként specifikus értékek

Strategized locking

Aszinkron kérésekre adott válaszok hatékony demultiplexálása

Asynchronous completion token

Beérkező események szálakhoz rendelése központilag irányított módon

Monitor object

Több művelet csoportosítása, hogy azok végrehajtása atominak tűnjön

Thread-local storage

Várakozás egy lock megszerzésére és egy előfeltétel teljesülésére

Scheduler

Kölcsönös kizáras és más szálak értesítése

Active object

Függvény meghívásának és végrehajtásának szétcsatolása

ManualResetEvent

Csak akkor lock-olunk, ha szükséges

Monitor object

Kívülről érkező események egyszálú szinkron

feldolgozása

Active object

Háttérben futó aszinkron művelet megszakítása

Cancellation token

Hatékony lock-olás egy- és többszálú esetben is

Strategized locking

The correct answer is:

Egyetlen szál futásának engedélyezése egy művelet eredményének elkészülte után → AutoResetEvent,

Aszinkron és szinkron feldolgozás szétcsatolása → Half-sync/half-async,

Több szál futásának engedélyezése egy művelet eredményének elkészülte után → ManualResetEvent,

Több egyidejű olvasás, de csak egyetlen egyidejű írás engedélyezése → Readers-writer lock,

Az alkalmazás által kezdeményezett műveletek elkészültét jelző események egyszálú szinkron feldolgozása → Proactor,

Egy háttérben futó művelet eredményéről információ szerzése, illetve az eredmény bevárása → Future/Task/Deferred,

Szálak egymást váltva dolgozzák fel a beérkező eseményeket → Leader-followers,

Szálanként specifikus értékek → Thread-local storage,

Aszinkron kérésekre adott válaszok hatékony demultiplexálása → Asynchronous completion token,

Beérkező események szálakhoz rendelése központilag irányított módon → Scheduler,

Több művelet csoportosítása, hogy azok végrehajtása atomikus tűnjön → Scoped locking,

Várakozás egy lock megszerzésére és egy előfeltétel teljesülésére → Guarded suspension,

Kölcsönös kizárás és más szálak értesítése → Monitor object,

Függvény meghívásának és végrehajtásának szétcsatolása → Active object,

Csak akkor lock-olunk, ha szükséges → Double-checked locking,

Kívülről érkező események egyszálú szinkron feldolgozása → Reactor,

Háttérben futó aszinkron művelet megszakítása → Cancellation token,

Hatékony lock-olás egy- és többszálú esetben is → Strategized locking

Question 10

Complete

Mark 0.80 out of 2.00

Flag question

A Reactor minta elemei a kövezkezők:

- Reactor
- EventQueue
- Handle
- Application
- Asynchronous operation processor
- Completion handler
- Reactive event handler

The correct answers are: Handle, EventQueue, Reactive event handler, Reactor, Application

Question **11**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

A Mutex egy olyan Szemaformak felel meg, amelynek számlálója 1 értékű.

Select one:

- True
- False

The correct answer is 'True'.

Question **12**

Complete

Mark 3.00 out of 3.00

 Flag question

Az Active object minta elemei a kövezkezők:

- Monitor object
- Interface
- Scheduler
- Handle
- Proxy
- Task
- Semaphore
- Active object
- Future
- Task queue

The correct answers are: Interface, Proxy, Task, Scheduler,
Future,
Active object,
Task queue

Question **13**

Complete

Mark 1.00 out of 1.00

 Flag question

A Scoped locking mintát melyik kulcsszóval lehet megvalósítani C# nyelv esetén?

Answer:

The correct answer is: lock

Question 14

Complete

Mark 1.00 out of 1.00

Flag question

Egy alkalmazásban párhuzamosan több szálról is jöhetnek nyomtatandó feladatok, de egyidőben egyszerre csak egyetlen dokumentum nyomtatása lehetséges. A nyomtatási munkák szinkronizálásához az alábbi minták adnak segítséget:

- Leader-followers
- Thread-local context
- ManualResetEvent
- AutoResetEvent

The correct answer is:

AutoResetEvent

Question 15

Complete

Mark 0.00 out of 1.00

Flag question

A Double-checked locking mintát melyik tervezési minta esetén célszerű használni az alábbiak közül?

- Command
- Strategy
- Visitor
- Singleton
- Observer

The correct answer is:

Singleton

Question **16**

Complete

Mark 1.00 out of 1.00

 Flag question

A Scoped locking mintát melyik kulcsszóval lehet megvalósítani Java nyelv esetén?

Answer: synchronized

The correct answer is: synchronized

Question **17**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

A Reactor és a Proactor minta is implementálható a half-sync/half-async minta segítségével.

Select one:

True

False

The correct answer is 'True'.

Question **18**

Complete

Mark 1.00 out of 1.00

 Flag question

Igaz vagy hamis a következő állítás?

A Double-checked locking mintát nem mindig lehet korrektül implementálni.

Select one:

 True False

The correct answer is 'True'.

Question **19**

Complete

Mark 1.00 out of 1.00

 Flag question

Egy node.js szerveren futó alkalmazás az alábbi konkurens és elosztott mintákat használja:

- Proactor
- Reactor
- Monitor object
- Double-checked locking
- Semaphor

The correct answers are:

Reactor,

Proactor

