

Atsiskaitančiojo kodas (vidko) C | 1 | 5 | 4 | 3 |
(pildo studentas)

Studijų modulio pavadinimas, kodas

T120B029 Programų sistemų analizės ir projektavimo įrankiai

Atsakingo dėstytojo vardas, pavardė Eduardas Bareiša

Egzamino data
(pildo atsakingas dėstytojas)

2017-06-02

Egzamino pradžia
(pildo atsakingas dėstytojas)

10:30

Egzamino trukmė
(pildo atsakingas dėstytojas)

1,5 val.

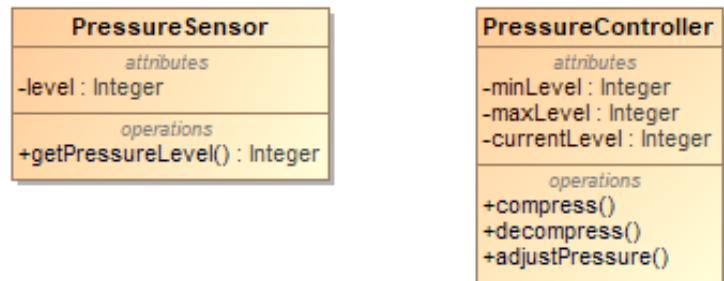
INSTRUKCIJA ATSISKAITANČIAJAM

1. Diagramų atvaizdavimui galima naudoti pieštuką. Pseudokodas turi atspindėti klasių diagramos esmę.

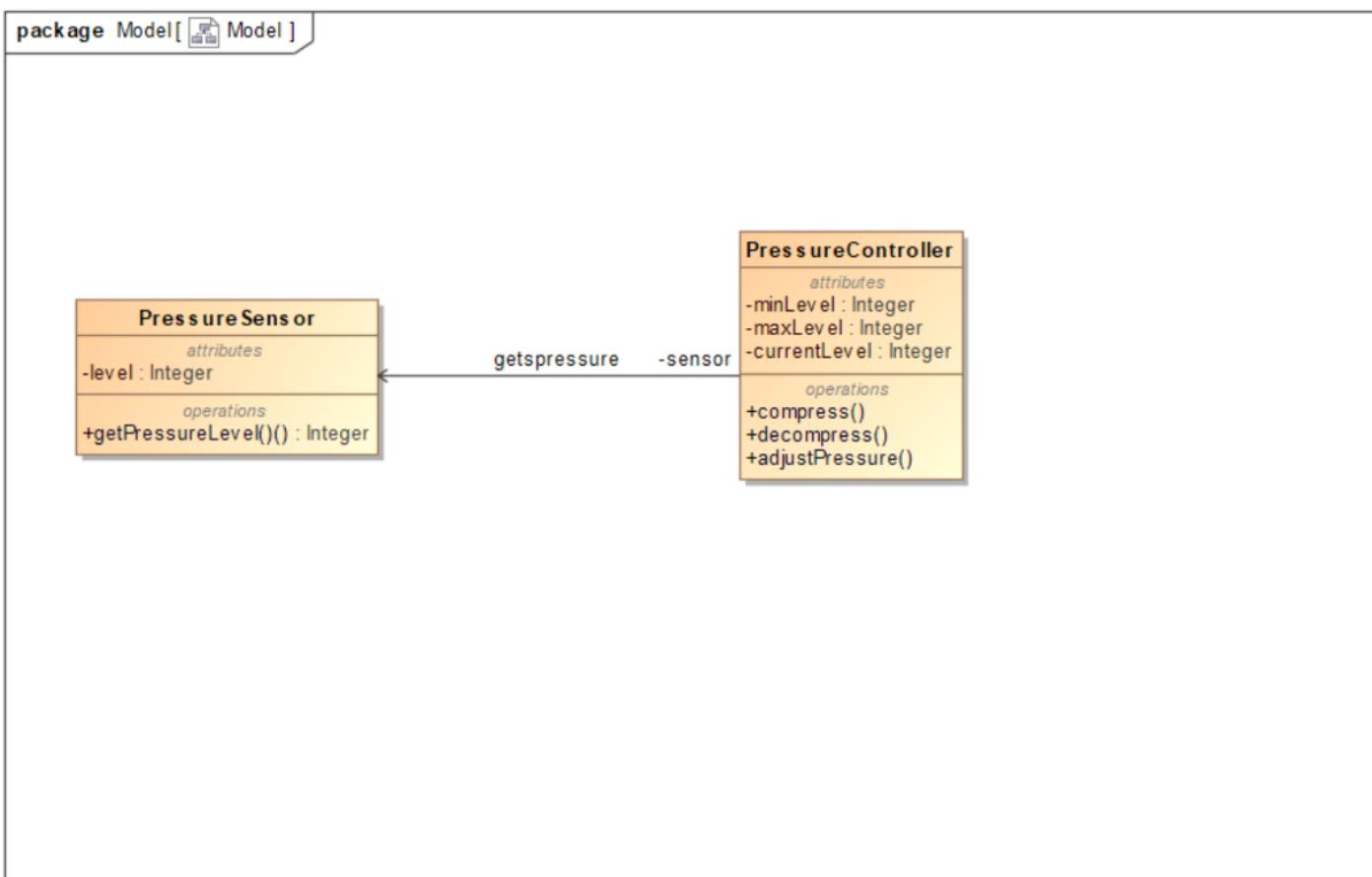
UŽDUOTYS:1

1. Klausimas: (0,5 balo) - association

Sujunkite klasses asociacijos ryšiu (Pav. 1) ir parašykite pseudokodą, demonstruojanti metodo `adjustPressure()` veikimą: slėgis didinamas, kai daviklio parodymai žemesnji už minimalią reikšmę ir slėgis mažinamas, kaip daviklio parodymai didesni už maksimalią reikšmę.



Pav. 1 Klasių diagrama „Kontroleris ir sensorius“



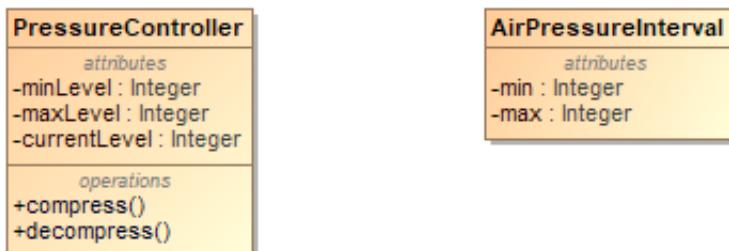
Listingas

```
public class PressureController {  
    private int minLevel;  
    private int maxLevel;  
    private int currentLevel;  
    private PressureSensor sensor  
  
    public compress() {  
        //....  
    }  
}
```

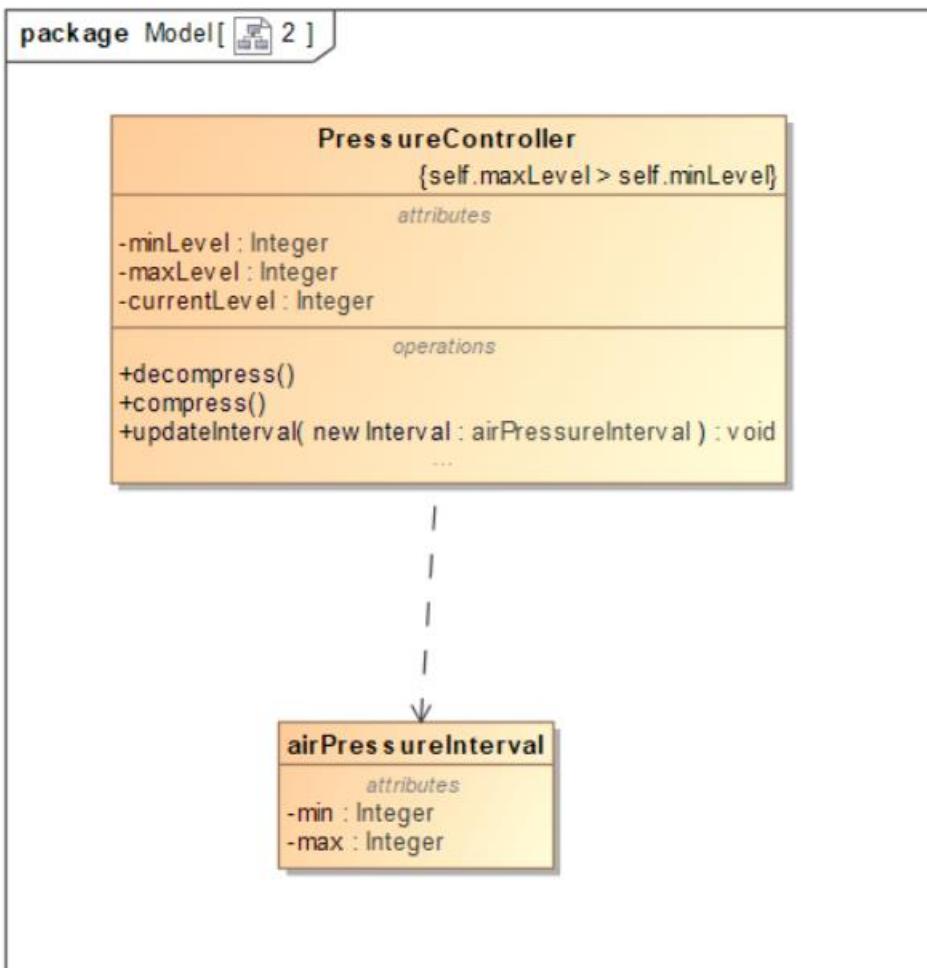
```
public decompress() {  
    //....  
  
}  
  
public adjustPressure() {  
    currentLevel = sensor.getPressureLevel();  
  
    if(currentLevel < minLevel) compress();  
    else if (currentLevel > maxLevel) decompress();  
  
}  
}
```

2. Klausimas: (0,5 balo) – dependency ---

Sujunkite klasses priklausomybės ryšiu (Pav. 2) ir sukurkite valdikliui naują metodą `updateInterval()`, atitinkantį priklausomybės ryšį ir atnaujinanti maksimalią ir minimalią reikšmes. Parašykite pseudokodą, demonstruojanti jūsų papildytą diagramą.



Pav. 2. Klasių diagrama „Kontroleris ir reikšmių intervalas“



Listingas

```

public class PressureController {
    private int minLevel;
    private int maxLevel;
    private int currentLevel;

    public void updateInterval(AirPressureInterval intervals) {
        minLevel = intervals.getMin();
        maxLevel = intervals.getMax();
    }
}

```

```
}
```

```
public void compress() {  
    //....  
}
```

```
public void decompress() {  
    //....  
}
```

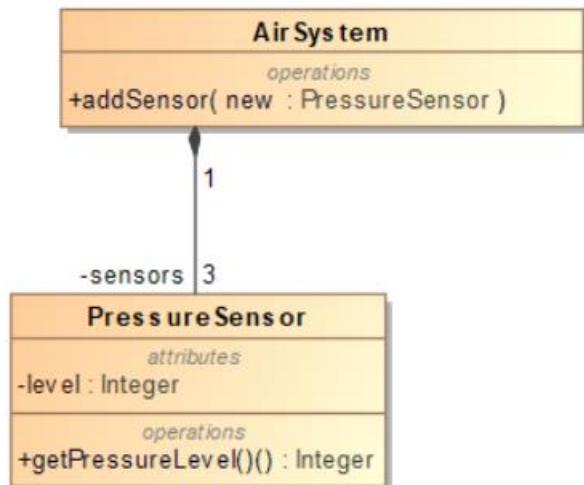
```
}
```

3. Klausimas: (1 balas) - composition

Sujunkite klasses kompozicijos ryšiu, nustatančiu trijų daviklių rinkinį (Pav. 3). Sukurkite sistemai naują metodą `addSensor()`, skirtą daviklio pridėjimui prie rinkinio. Parašykite pseudokodą, demonstruojanti jūsų papildytą diagramą ir metodo veikimą.



Pav. 3 Klasų diagrama "Daviklių rinkinys"



Listingas

```

public class AirSystem {
    private PressureSensor[] sensors;

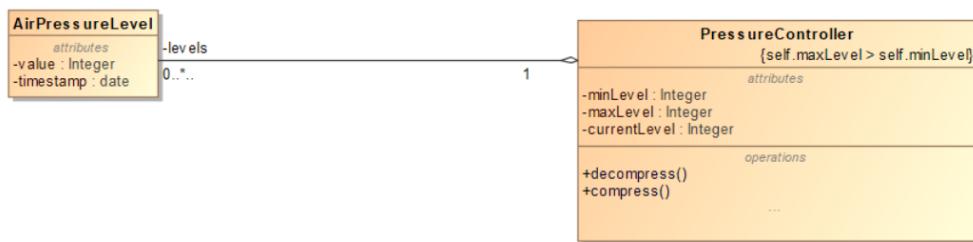
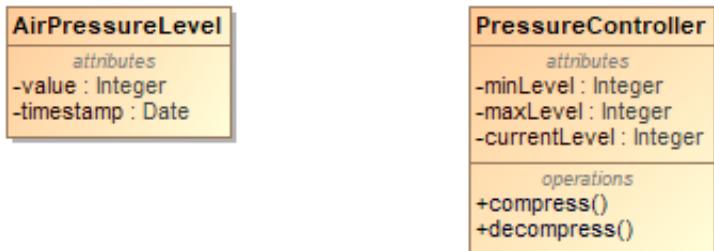
    public AirSystem() {
        sensors = new PressureSensor[3];
    }

    public void addSensor(PressureSensor new) {
        sensors[sensors.length] = new PressureSensor(..)
    }
}

```


4. Klausimas (1 balas) - aggregation

Sujunkite klasses aggregacijos ryšiu (**Error! Reference source not found.**), leidžiančiu saugoti ankstesnių matavimų istoriją, susidedančią iš reikšmės ir laiko atžymos. Pridėkite metodą `saveMeasurement()` ir parašykite pseudokodą, demonstruojanti jūsų papildytą diagramą ir metodo veikimą.



Pav. 4 klasų diagrama „Matavimų istorija“

Listingas

```

//4

public class PressureController {
    ArrayList<AirPressureLevel> levels = new ArrayList<AirPressureLevel>();

    public PressureController() {

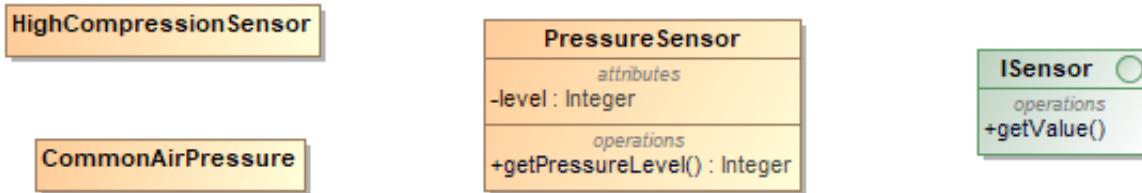
    }

    public void saveMeasurement(AirPressureLevel new) {
        levels.add(AirPressureLevel);
    }
}

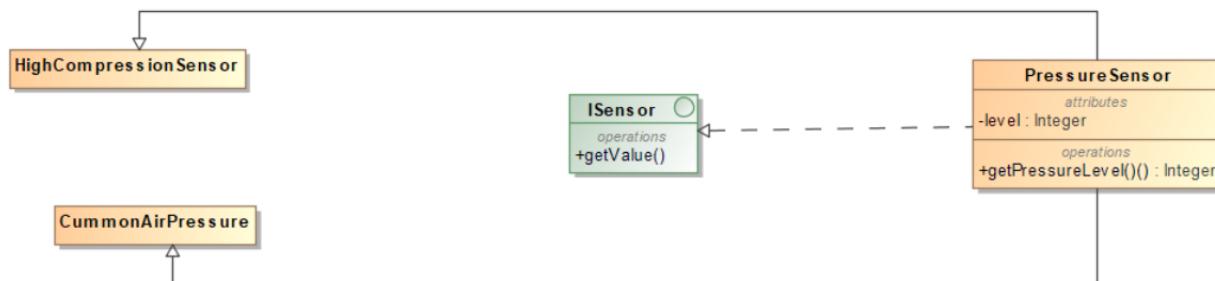
```


5. Klausimas (1 balo) generalization, Relization

Sujunkite klasses ir sąsajas paveldėjimo ir realizacijos ryšiais (Pav. 5). Parašykite pseudokodą, demonstruojanti jūsų papildytą diagramą ir metodo `getValue()` realizaciją, kuri grąžina daviklio matavimus.



Pav. 5 Klasių diagrama „Daviklių hierarchija“



Listingas

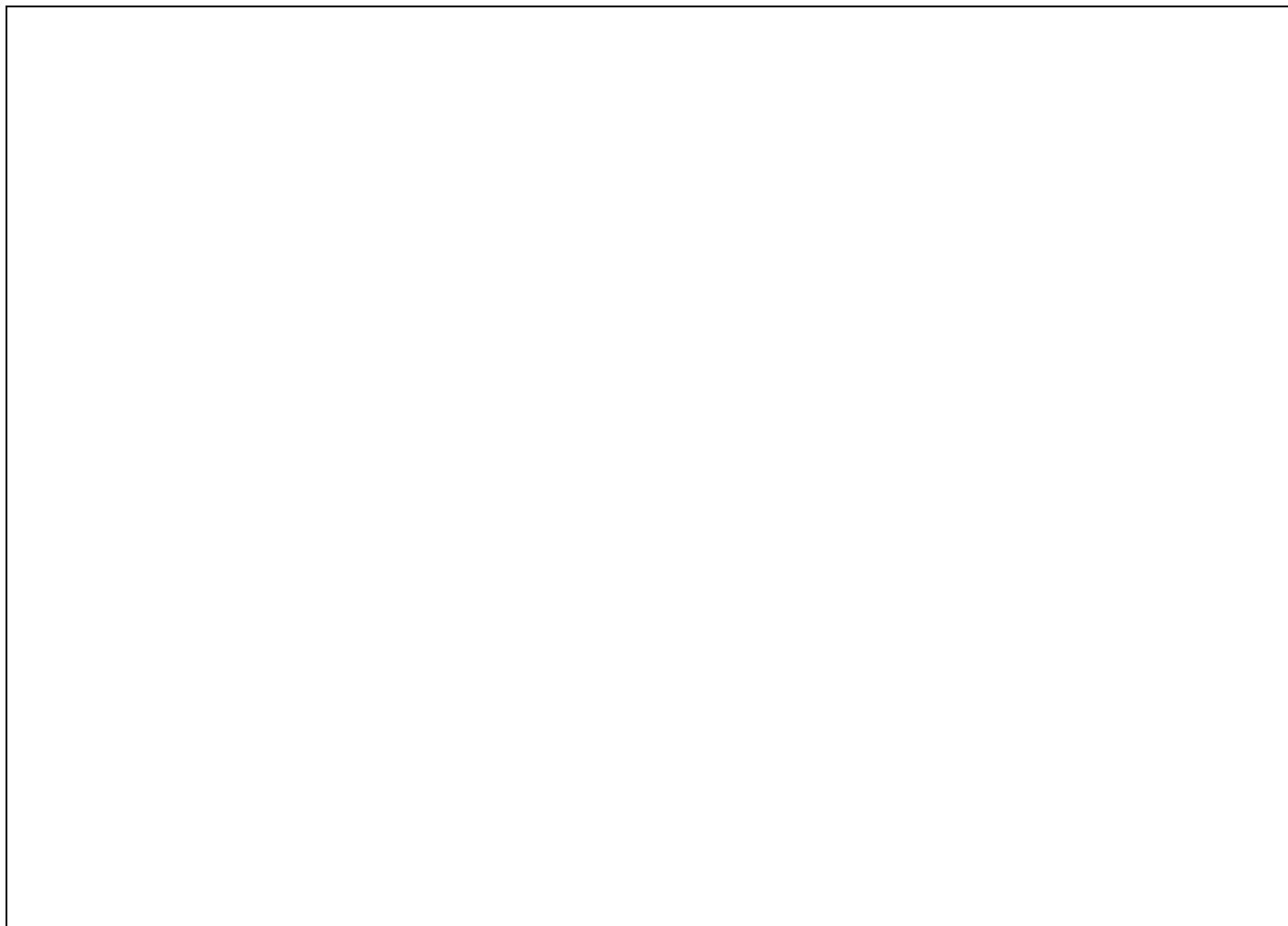
```

public class PressureSensor implements ISensor {
    private int level;

    public PressureController() {
    }

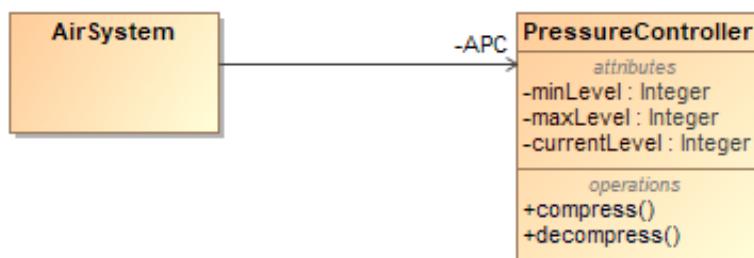
    public void getPressureLevel() {
        //..
    }

    public void getValue {
        //is Isensor paimame HighCompression duomenis ir CommonAirPressure duomenis
    }
}
  
```



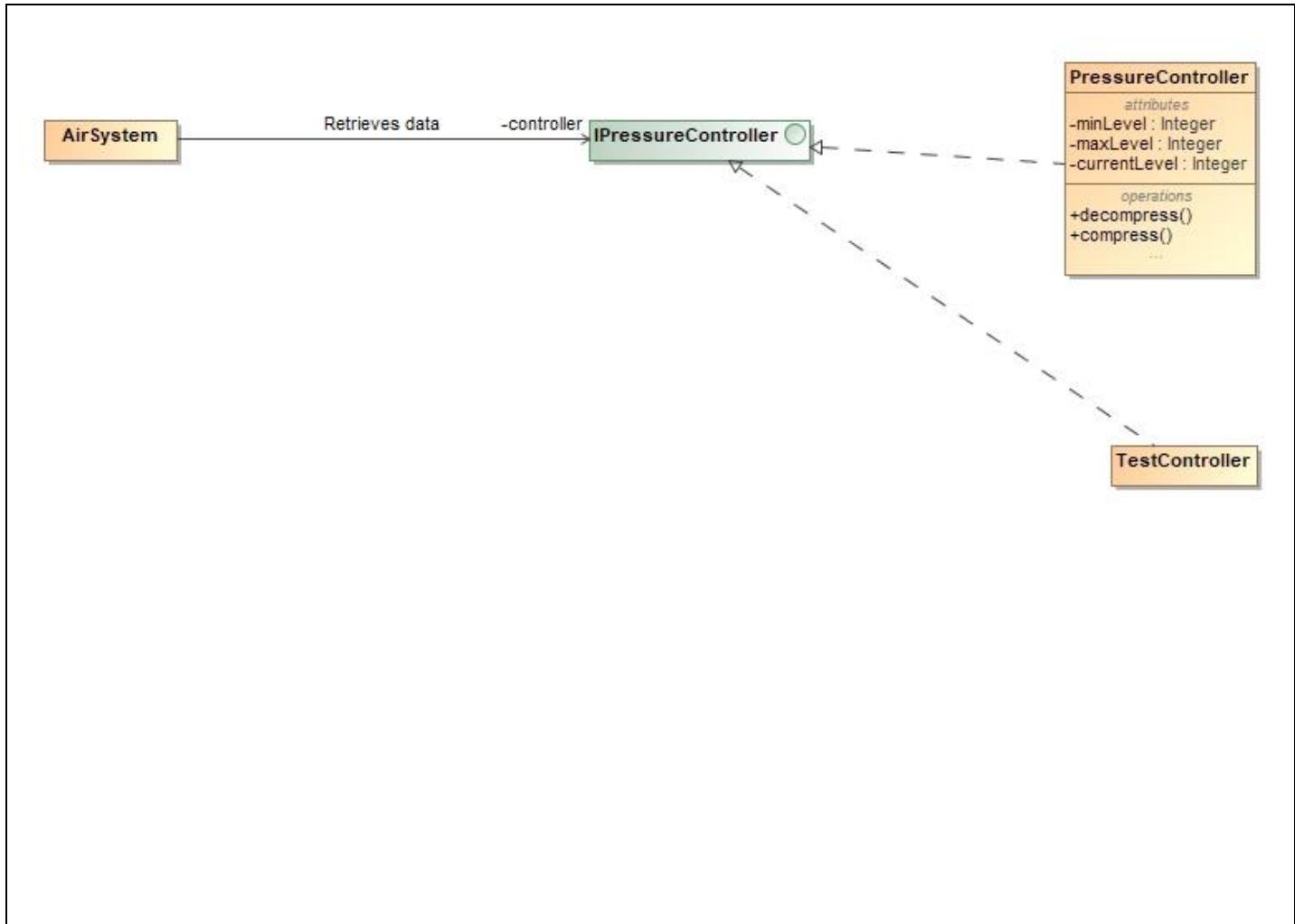
6. Klausimas: (1,5 balai) – Dependency Inversion

Klasių diagramoje (Pav. 6) panaudokite „dependency inversion“ principą, kad klasėje AirSystem galima būtų taikyti naują arba testavimo kontrolerį. Parašykite pseudokodą, demonstruojantį „dependency inversion“ realizaciją per konstruktorių.



Pav. 6 Klasių diagrama

Klasių diagrama

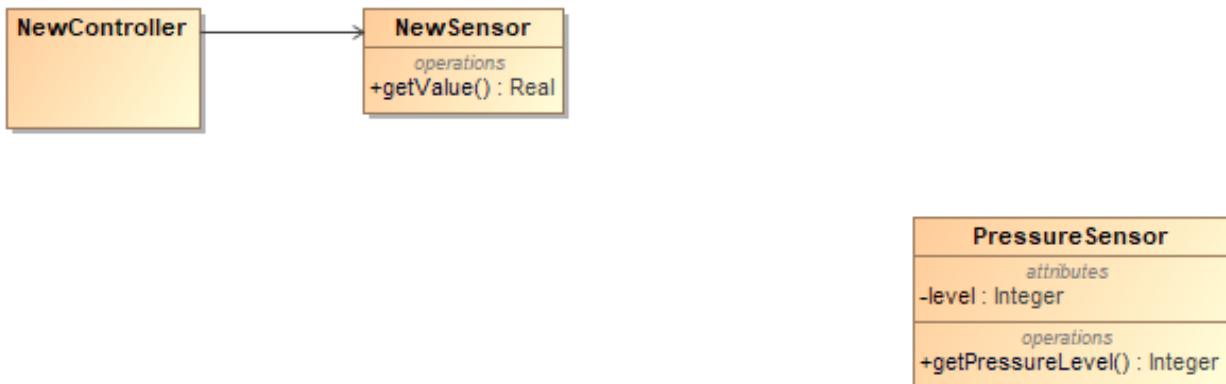


Listingas

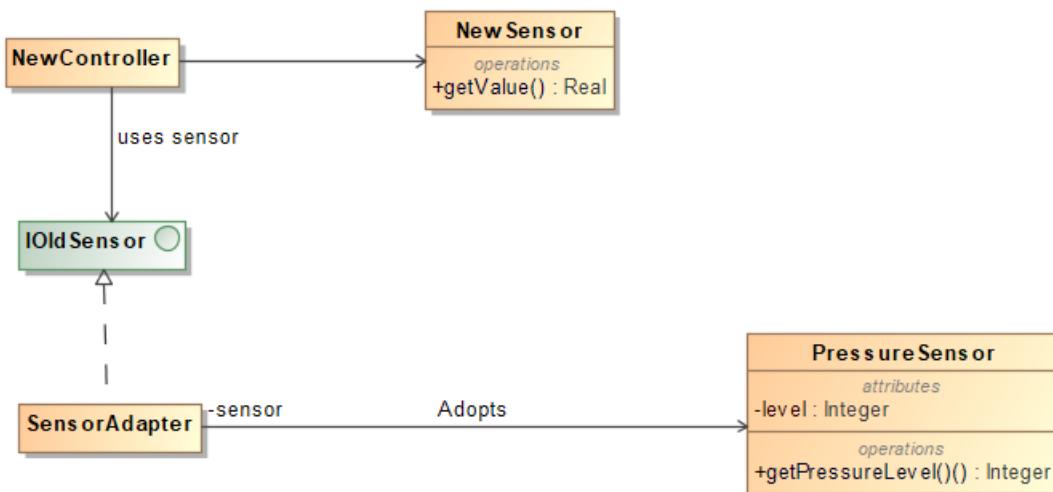
```
public class AirSystem {  
    private IPressureController controller;  
  
    public AirSystem(IPressureController contr) {  
        controller = contr;  
    }  
}
```

7. Klausimas: (1,5 balo) - Adaper

Klasių diagramoje (Pav. 7) pritaikykite Adapter projektavimo šabloną taip, kad galima būtų prijungti esamą daviklį prie naujo valdiklio, naudojančio naują sensorių. Parašykite pseudokodą, demonstruojantį jūsų papildytą diagramą ir adapterio klasės veikimą.



Pav. 7 Klasių diagrama „Adapter“ šablono taikymas



Listingas

```
Public class SensorAdapter : IOldSensor {  
    Private PressureSensor sensor;  
    //getData grazina integer, seno sensoriaus duomenis  
    Public getData() {  
        Sensor.getPressureLevel();  
    }  
}
```

8. Klausimas (1,5 balas) - OCL

Atskirti užduotį į 3 klausimus

Parašykite OCL apribojimus (Pav. 8): **invariantq**, kai min reikšmė negali būti didesnė už max reikšmę, **priešsąlyga** kai sumažinti slėgį galima tik kai dabartinė reikšmė didesnė už max, **posqlyga** kai reikšmių istorija pailgėja viena reikšme.

Parašykite pseudokodą, demonstruojanti OCL apribojimų taikymą diagramos klasėse.



Pav. 8. Klasų diagrama

OCL apribojimai

Inv	<div style="border: 1px solid black; padding: 10px;"> PressureController $\{self.maxLevel > self.minLevel\}$ attributes <code>-minLevel : Integer</code> <code>-maxLevel : Integer</code> <code>-currentLevel : Integer</code> operations <code>+decompress()</code> <code>+compress()</code> </div>
Pre	<div style="border: 1px solid black; padding: 10px;"> decompress <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> Owned Parameter Owned Parameter Set Owned Rule Owned Template Signature Owner Owning Template Parameter Package Import Postcondition Precondition Qualified Name Raised Exception Realizing Element Redefined Element Redefined Operation </div> <div style="margin-bottom: 10px;"> Properties: All </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> $\{\} \text{ mustBeHigher} = self.maxLevel > self.currentLevel$ [PressureController::decompress] </div> <div style="border: 1px solid #ccc; padding: 5px;"> PressureController </div> <div style="border: 1px solid #ccc; padding: 5px; background-color: yellow;"> $\{\} \text{ mustBeHigher} = self.maxLevel > self.currentLevel$ [PressureController::decompress] </div> <div style="border: 1px solid #ccc; padding: 5px;"> PressureController::decompress </div> </div>
Pos t	<p>Čia rezultatų validavimas – reikia pridėti metodą, kuris prided prie istorijos ir gražina rezultatą. <code>History.length + 1 = history.add(newValue)</code></p>

Listingas

```
public class PressureController {
```

```
private int minLevel;
private int maxLevel;
private int currentLevel;

public PressureController(int minlevel, int maxlevel, int currentlevel) {
    if(checkInvariant(minlevel, maxlevel)) {
        //... vykdomas konstruktorius
    }
}

public void compress() {
    if(checkInvariant(minLevel, maxLevel))
    {
        //.. vykdomas kodas
        if(checkPostCondition) {return ats}
        else return null;
    }
}

public void decompress() {

    if(checkInvariant(minLevel, maxLevel) && checkPrecondition(maxLevel, currentLevel))
    {
        //...
    }
}

checkInvariant(int min, int max) {
    return min < max;
}

checkPrecondition(int max, int current) {
    return current > max;
}

checkPostCondition(int[] history){

}

}

Public class PressureController() {
```

9. Klausimas (1,5 balai) – Sequence Diagram

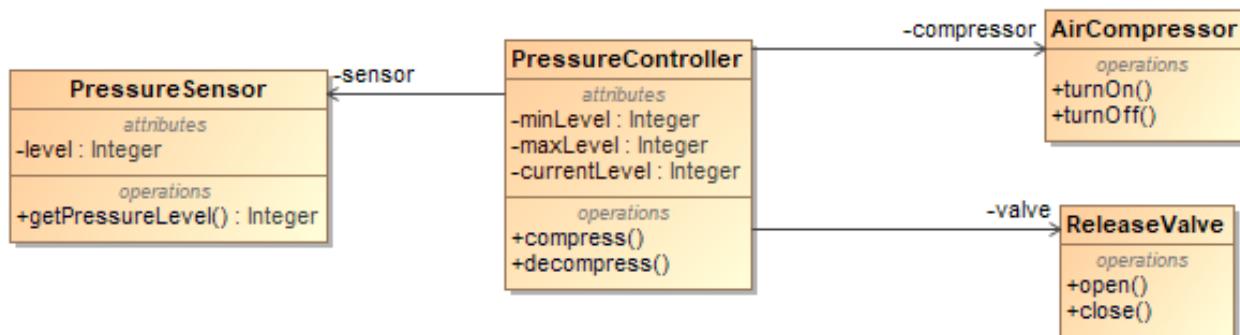
Klasių diagramai (Pav. 9) sudarykite sekų diagramą, kurioje būtų kas 5 sekundes reguliojamas **šviesos lygis** kai matavimų reikšmė

žemiau už minimumą, **įjungiamą lempą**,

viršija maksimumą, **uždaromos užuolaidos**,

tarp minimalios ir maksimalios reikšmės, turi būti **atidaromos užuolaidos** ir **išjungiamą šviesą**.

Parašykite pseudokodą, demonstruojantį klasės Controller veikimą.



Pav. 9. Klasių diagrama „Irenginių valdymas per kontrolerį“

Sekų diagrama

Listingas

