|  |  |
| --- | --- |
| **Assignment Case** |  |
| COMP6153001  Operating System |
| **Computer Science** | **O223-COMP6153-IX11-01** |
| ***Valid on*** *Odd Semester Year 2021/2022* | **Revision 00** |

1. Seluruh mahasiswa tidak diperkenankan untuk:

*All students are not allowed to:*

* + - Berdiskusi dan/atau bekerja sama dengan mahasiswa lainnya

*Discuss and/or work together with other student participants*

* + - Melihat sebagian atau seluruh jawaban mahasiswa lain

*Seeing a part or the whole answer from another student*

* + - Membuka dan menyalin dari **BUKU** atau **CATATAN**, **VIDEO** dari pengajar (recording kelas, VBL, Youtube, dsb) dan **REFERENSI** lainnya

*Open and copy from any resources such as notes, videos (class recording, VBL, Youtube, etc) and other references*

* + - Membuka dan menyalin jawaban dari internet (google, stackoverflow, dsb)

*Open and copy answer from the internet (google, stackoverflow, etc)*

* + - Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal,

*Working with another theme which is not in accordance with the existing theme in the matter of the case,*

* + - Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + - Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika mahasiswa terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai mahasiswa** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the student is proved to the actions described in point 1 above, the score of the student which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan jawaban, segala jenis pengumpulan jawaban di luar jadwal tidak dilayani.

*Pay attention to the submission schedule, all kinds of submission outside the schedule will not be accepted*

1. Bila Anda tidak membaca peraturan ini, maka Anda dianggap telah membaca dan menyetujuinya

*If you have missed to read these regulations, so you are considered to have read and agreed on it*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **UAP**  *Final Exam* |
| 40% | 60% |

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

|  |
| --- |
| **Software**  *Software* |
| VMware Workstation 15  VM Ubuntu Client 20.04  Java 8  Eclipse 2020.6  NachOS 5.0j |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri, dan uap untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment collection for this subject are described as follows:*

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **UAP**  *Final Exam* |
| DOCX, JAVA, CLASS | JAVA, CLASS |

## Soal

*Case*

**Bluejack Pet Shop**

**Bluejack Pet Shop** is pet shop that is currently building their website to provide better service and management. You are hired as their programmer and to create a **bash command line** to do each of the following tasks:

1. Create a **Repository** directory in **home directory** using the following structure. (single execution)

<Home directory>

`-- Pet Shop

|-- Reptiles

| |-- Iguana

| |-- Turtle

|-- Birds

| |-- Canary

| |-- Parrot

1. **Create a file**  named **Transaction.txt** and **Sales.txt** on **~/Pet Shop/Birds/Canary/** and change both file’s access time and modification time to **1 December current year,** on **18:30:50**
2. From **home directory**, **search** for all **empty files**.
3. **Display all** running process in system regardless from where they were executed and the users, and **show the parent and child process** with indentation.
4. Java Programming

You are also tasked to create this program using **Java programming** language with following concepts:

1. Abstract Class

You need to design at least **three** classes, **one abstract** class, and **two concrete** classes. Abstract class consists of all **common** attributes and behavior that both of concrete class had. Concrete class consist of **specific** attribute and behavior that not common between the concrete classes

1. Encapsulation

To **hide** the data of a class from an **illegal** direct access, all of the attributes of the class must be **encapsulated** and will be accessed using an **accessor** and **mutator** that may perform validation before accessing the encapsulated attribute

1. Inheritance

All of the concrete class **must inherit all** attribute and behavior from the abstract class

1. Polymorphism

If the concrete class has **a specific implementation** of the inherited behavior (method) that **differ** from the abstract class, the concrete class can **override** or **overload** the behavior from the abstract class

The following are the specifications for the program:

* First, the program will **display main menu** consists of:
* Add New Pet
* View all Pets
* Exit



Figure 1. Menu

* If the user chooses **menu 1** (“**Add New Pet**”), the program will prompt the user to input:
  + **Pet Type.**  Validate the input must be either 1 (dog) or 2 (snake).
  + **Pet’s Breed**. Validate the input must be **between 5 and 20 characters**.
  + **Age**. Validate the input must be **more than 0**.
  + **Base Price**. Validate the input must be **between 10000 and 10000000**.
  + **Gender**. Validate the input must be either **Female**or **Male** with **case sensitive.**
    - * + If **Pet Type is dog**, then prompt the user to input:

**Dog Size**. This input must be either **Small**, **Medium**, or **Big** with **insensitive case**.

**Dog Fur Length**. This input must be either **Short**, **Medium**, or **Long** with **case sensitive.**

* + - * + If **Pet Type is snake**, then prompt the user to input:
        + **Snake Length**. This input must be **between 0.1** and **10**
  + After that, the program will **generate** the **Pet ID**  depending on the **pet type**:
* **Dog Pet ID Format:**

**DYXXX**

**Y = The first letter of the dog’s size**

**X = random number from 0 until 9**

**Example:**

**DB098, DM139**

**Snake Pet ID Format**:

**SXXXX**

**X = random number from 0 until 9**

**Example:**

**S1356**

* + Then the program will **insert** the **pet data**, **show** a **success message**, and **return to main menu**.

Text

Description automatically generated

Figure 2. Add Pet Dog

Graphical user interface, application

Description automatically generated

Figure 3. Add Pet Snake

* If the user chooses **menu 2** (“**View all pets**”):
  + If there is **no pet data**, the program will show **message** and **return to main menu**.



* + **Otherwise**, the program will **display all animal data** that are stored on the list.
  + Then, the program will **prompt the user to enter the Pet ID** to view the pet’s detailed information. Validate that the **Pet ID** must be **exists**.
  + After that, the program will, **show the specified pet’s data** and **return to main menu**.
  + The price of **Dog** is **calculated** based on the following formula:

**Dog Price = Base Price + Fur Length Price**

|  |  |
| --- | --- |
| **Dog’s Fur Length** | **Price** |
| Short | Rp. 0 |
| Medium | Rp. 20000 |
| Long | Rp. 50000 |

* + The selling price of **Snake** is **calculated** based on the following formula:

**Snake Price = Base Price + (Rounded Down Length \* 10000)**

Graphical user interface, text, application, email

Description automatically generated

Figure 4. View All Pet

* If the user chooses **menu 3** (“**Exit**”), the program will **exit**.