|  |  |
| --- | --- |
| **Assignment Case** |  |
| COMP6153001  Operating System |
| **Computer Science** | **O223-COMP6153-WT12-01** |
| ***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 |

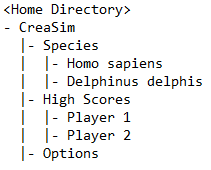
## Soal

*Case*

**CreaSim**

**CreaSim** is a **game** developed by **Bluejack Studios**, where you are the **‘creator’ of life** on a desolate planet. As the new intern ofBluejack Studios, you are tasked to create a **bash command line** to do each of the following tasks:

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



1. Create a file named "**Score.sav**" inside "**Creasim/High Scores/Player 1**", and **change its permission** just like so:

**Owner** : Read, Write**,** and Execute the file

**Group** : Read and Write the file

**Others** : Read the file

1. From **home directory**, **search** for **file(s)** with the following criteria:
   * The filename **is** "**Score.sav**"
   * Inside the **Repository** directory
2. **Show all processes owned by root**
3. Java Programming

**Bluejack Studios** wants you to create a **simple prototype of CreaSim**. You are tasked to create this program using **Java programming** language with the 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 concrete classes had. Concrete classes consist of **specific** attributes and behaviors that are not common between both 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 can only be accessed using an **accessor** and **mutator** that may perform validation before accessing the encapsulated attribute.

1. Inheritance

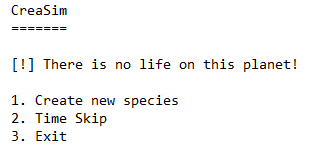
All of the concrete classes **must inherit all** attributes and behaviors from the abstract class.

1. Polymorphism

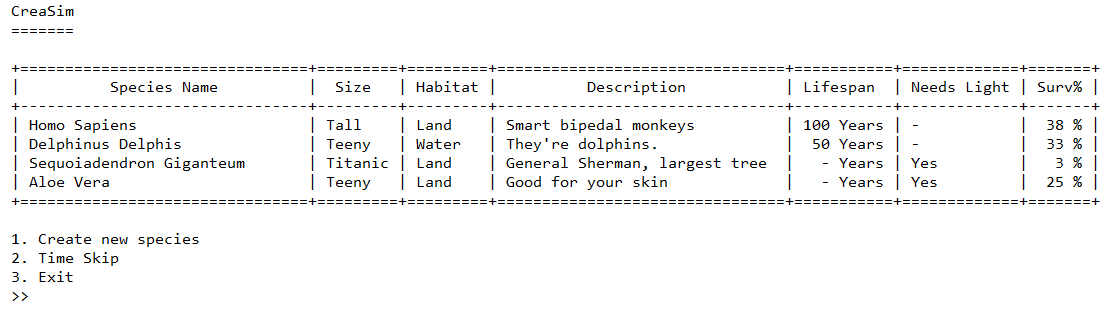
If the concrete class has **a specific implementation** of the inherited behavior (method) that **differs** from the abstract class, the concrete class can **override** or **overload** the behavior from the abstract class.

The following are the specifications for the program:

* First, the program will **display main menu** consists of:
* List of species
  + If there are **no species on the planet**, print an error **“[!] There is no life on this planet!”**.
  + else, **show every species** and **its details**.
* Create New Species
* Time Skip
* Exit



*Figure 1. Menu (no species)*

**

*Figure 2. Menu (with species)*

* If the user chooses **menu 1** (“**Create new species**”), the program will prompt user to fill the details, such as:
  + **Input species type**. Validate the input must be between “**Animal**” or “**Plant**”. **(case-sensitive)**
  + **Input species name**. Validate the input must be **between 5 to 30 characters long. (inclusive)**
  + **Input species size**. Validate the input must be between “**Teeny**”, “**Tall**”, or “**Titanic**”. **(case-sensitive)**
  + **Input species habitat**. Validate the input must be between “**Land**”, “**Water**”, or “**Air**”. **(case-sensitive)**
  + **Input species description**. Validate the input must be **between 1 to 30 characters long. (inclusive)**
  + **Input species lifespan (Animal)**. Validate the input must be between **1 to 150. (inclusive)**
  + **Input species light dependency (Plant)**. Validate the input must be between “**Yes**”or “**No**”**. (case-sensitive)**
  + **Species survivability percentage.** This detail is not stored within the class, but is calculated by the **calculateSurvivability()** method.
    - **Animal** survivability formula is as follows:

|  |
| --- |
| **Survivability Percentage**  = (100 / 1050) \* **[SizeMod]** \* **Lifespan** |

* + - **Plant** survivability formula is as follows:

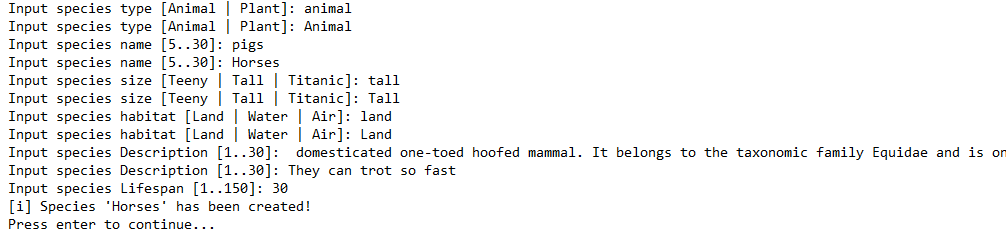
|  |
| --- |
| **Survivability Percentage**  = (100 / 28) \* **[SizeMod]** \* **[LightMod]** |

Where:

|  |  |
| --- | --- |
| **Size** | **SizeMod** |
| “Teeny” | 7 |
| “Tall” | 4 |
| “Titanic” | 1 |

|  |  |
| --- | --- |
| **Light Dependency** | **LightMod** |
| “Yes” | 1 |
| “No” | 4 |

* + After that, the program will **insert** **the details of the species** and **show** a **success message**, and **return to the main menu**



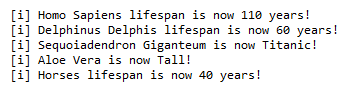
*Figure 3. Add Species*

* If the user chooses **menu 2** (“**Time Skip**”):
  + If there are **no species on the planet**, the program will show an **“[i] Nothing changed...” message** and **return to the main menu**.

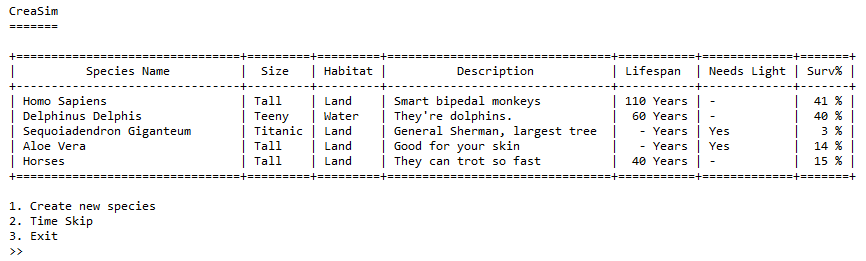


*Figure 4. Time skip (no species)*

* + **Otherwise**, the program will **evolve every animal** and **grow every plant** on the planet, like so:
    - **Animals evolve** by **adding their lifespans by 10**. Print their new lifespan.
    - **Plants grow** by **increasing their size**, from Teeny to Tall, and from Tall to Titanic. Print their new size.



*Figure 5. Time Skip*



*Figure 6. Species after time skip*

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