Учреждение образования Республики Беларусь

«Гомельский государственный технический университет им. П. О. Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Отчет по лабораторной работе №1

по дисциплине «Технология разработки программного обеспечения»

на тему «Развитие представлений о разработке программ. Объектно-ориентированное программирование в *Java*»

Выполнил:

студент группы ИТП-21

Иванов. Д. А.

Проверил:

преподаватель

Малиновский И. Л.

Гомель, 2023

**Цель:** изучить основы языка программирования Java. Научится создавать классы и изучить их основные возможности. Изучить такие понятия, как поля класса, конструктор класса и методы класса.

**Задание**

1. Разработать *UML*-диаграмму иерархии классов, согласно варианта.

2. При наименовании компонентов руководствоваться соглашением о наименовании.

3. При описании иерархии использовать наследование и композицию.

4. На основе *UML*-диаграммы разработать иерархию классов на языке *Java*.

5. Весь код должен быть снабжен элементами документирования.

6. Разработанную иерархию поместить в .*jar* файл для дальнейшего использования в качестве библиотечных классов.

7. Создать консольное приложение для демонстрации работы созданных классов.

8. Составить отчет о проделанной работе.

**Индивидуальное задание**

1.1 Создать иерархию комнатных растений с полями название, высота(размер), продолжительность жизни.

1.2 Растения могут быть нескольких видов: Красивоцветущие,

Суккуленты, Папоротники. Реализовать через наследование. У красивоцветущих добавить поле

продолжительность цветения.

1.3 Создать не менее 10 объектов различных растений.

1.4 Вывести всю информацию по красивоцветущим.

1.5 Вывести все растения, продолжительность жизни которых менее 5 лет.

1.6 Вывести все папоротники, высота которых больше заданной.

**Ход работы**

Разработаем иерархию классов. Создадим абстрактный класс *HousePlant*, который будет содержать поля (имя, высота, время жизни), методы (*GetTotalWeight*, *GetFullInfo*, методы для получения данных полей). От абстрактного класса *HousePlant* наследуем 3 класса: *FernPlant*, *FloweringPlant*, *SicculentPlant*. Создадим *UML*-диаграмму классов средствами *Visual Paradigm*. Результат создания *UML*-диаграммы классов представлен на рисунке 1.

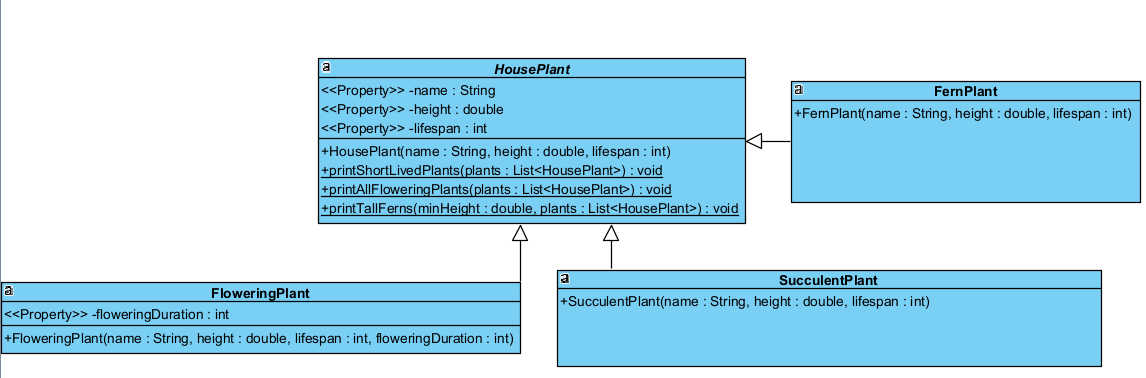
****

Рисунок 1 – *UML*-диаграмма классов

После создания *UML*-диаграммы реализуем иерархию классов в проекте *Java*. Также создадим метод *main*, в котором разместим основную логику приложения.

Результат выполнения метода *printAllFloweringPlants* представлен на рисунке 2:



Рисунок 2 – Результат выполнения метода *printAllFloweringPlants*

Результат выполнения метода *printShortLivedPlants* представлен на рисунке 3:

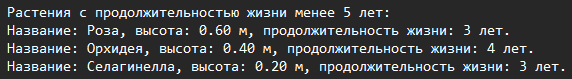
**

Рисунок 3 – Результат выполнения метода *printShortLivedPlants*

Результат выполнения метода *printTallFerns* представлен на рисунке 4:



Рисунок 4 – Результат выполнения метода *printTallFerns*

Для того, чтобы поместить разработанную иерархию в файл *.jar*, необходимо кликнуть ПКМ по *java*-проекту, выбрать пункт *Export*, открыть папку *Java* и выбрать пункт *JAR File*. После этого нужно выбрать необходимые файлы и указать путь к месту хранения файла. Результат создания *.jar* файла представлен на рисунке 5:



Рисунок 5 – Результат создания *.jar* файла

**Вывод:** К основным программным конструкциям *Java* относятся *if*, *for*, *while*, *switch* и т. д. Для создания класса применяется ключевое слово *class*. Класс может содержать поля и методы. Для того, чтобы обработать вероятные ошибки, в программе была проведена работа с исключениями с помощью операторов *try – catch – finally*.