

ЛАБОРАТОРНА РОБОТА №2. СПАДКУВАННЯ ТА ПОЛІМОРФІЗМ JAVA

Мета:

- Ознайомитися з реалізацією спадкування в Java.
- Ознайомитися зі шляхами реалізації поліморфізму в Java.
- Створити додаток з використанням спадкування і реалізацією поліморфізму.

Створені класи та інтерфейс:

AbstractForm:

```
package com.excore.java_lab_2.model;

public abstract class AbstractForm implements IWeight {
    protected final Wood wood;

    public AbstractForm(Wood wood) {
        this.wood = wood;
    }

    public Wood getWood() {
        return wood;
    }

    public abstract float volume();

    @Override
    public float weight() {
        return volume() * wood.getDensity();
    }
}
```

IWeight:

```
package com.excore.java_lab_2.model;

public interface IWeight {
    float weight();
}
```

Cylinder:

```
package com.excore.java_lab_2.model;

public class Cylinder extends AbstractForm {
    private final float length;
    private final float diameter;

    public Cylinder(Wood wood, float length, float diameter) {
        super(wood);
        this.length = length;
        this.diameter = diameter;
    }

    public float getLength() {
        return length;
    }

    public float getDiameter() {
        return diameter;
    }

    @Override
    public String toString() {
        return "Cylinder{" +
            "wood=" + wood.getName() +
            ", weight=" + weight() +
            '}';
    }

    @Override
    public float volume() {
        return (float) (Math.PI * diameter * diameter / 4 * length);
    }
}
```

Waste:

```
package com.excore.java_lab_2.model;

public class Waste implements IWeight{
    private final float weight;

    public Waste(float wasteWeight) {
        this.weight = wasteWeight;
    }

    public float weight() {
        return weight;
    }

    @Override
    public String toString() {
        return "Waste{" +
            "weight=" + weight +
            '}';
    }
}
```

AbstractStore:

```
package com.excore.java_lab_2.store;

import java.util.Arrays;

public abstract class AbstractStore<T> {
    protected int count = 0;
    protected Object[] arr = new Object[3];

    public int getCount() {
        return count;
    }

    public abstract T[] getArr();

    protected T get(int idx) {
        if (idx >= 0 && idx < count) {
            return (T)arr[idx];
        }
        return null; // Out of range
    }

    protected void add(T newItem) {
        if (arr.length == count) {
            arr = Arrays.copyOf(arr, count + count / 2 + 1);
        }
        arr[count++] = newItem;
    }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < count; i++) {
            sb.append(arr[i]).append('\n');
        }
        return sb.toString();
    }
}
```

Класи які були змінені:

Timber:

```
package com.excore.java_lab_2.model;

public class Timber extends AbstractForm {
    private final float length;
    private final float height;
    private final float width;

    public Timber(Wood wood, float length, float height, float width) {
        super(wood);
        this.length = length;
        this.height = height;
        this.width = width;
    }

    public float getLength() {
        return length;
    }

    public float getHeight() {
        return height;
    }

    public float getWidth() {
        return width;
    }

    @Override
    public float volume() {
        return length * height * width;
    }

    @Override
    public String toString() {
        return "Timber{" +
            "wood=" + wood.getName() +
            ", weight=" + weight() +
            '}';
    }
}
```

ProductStore:

```
package com.excore.java_lab_2.store;

import com.excore.java_lab_2.model.IWeight;

import java.util.Arrays;

public class ProductStore extends AbstractStore<IWeight> {
    public IWeight[] getArr() {
        return Arrays.copyOf(arr, count, IWeight[].class);
    }

    @Override
    public IWeight get(int idx) {
        return super.get(idx);
    }

    @Override
    public void add(IWeight newItem) {
        super.add(newItem);
    }

    @Override
    public String toString() {
        return "Перелік виробів:\n" + super.toString();
    }
}
```

WoodDirectory:

```
package com.excore.java_lab_2.store;

import com.excore.java_lab_2.model.Wood;

import java.util.Arrays;

public class WoodDirectory extends AbstractStore<Wood> {

    {
        arr[0] = new Wood(1, "Модрина", 1.1f);
        arr[1] = new Wood(2, "Ялина", 0.9f);
        arr[2] = new Wood(3, "Сосна", 0.7f);
        count = 3;
    }

    public Wood[] getArr() {
        return Arrays.copyOf(arr, count, Wood[].class);
    }

    @Override
    public Wood get(int id) {
        for (int i = 0; i < count; i++) {
            if (((Wood)arr[i]).getId() == id) {
                return (Wood)arr[i];
            }
        }
        return null; // Not found
    }

    public boolean addChecked(Wood newWood) {
        if (get(newWood.getId()) != null) {
            return false; // Id already present
        }

        super.add(newWood);
        return true;
    }

    @Override
    public String toString() {
        return "Каталог деревини:\n" + super.toString();
    }
}
```

calcWeight() в класах TestApp та TestAppConsole:

```
private float calcWeight() {  
    float result = 0f;  
    for (IWeight t :  
        ps.getArr()) {  
        result += t.weight();  
    }  
    return result;  
}
```

Також тестові класи були змінені щоб тестувати нові можливості. Код надто довгий щоб сюди вставляти.

Тестування:

За допомогою класу TestApp:

```
Каталог деревини:
Wood{id=1, name='Модрина', density=1.1}
Wood{id=2, name='Ялина', density=0.9}
Wood{id=3, name='Сосна', density=0.7}

Перелік виробів:
Timber{wood=Модрина, weight=1.1}
Timber{wood=Ялина, weight=1.8}
Waste{weight=3.6}
Cylinder{wood=Сосна, weight=8.24668}
Timber{wood=Сосна, weight=9.702}
Cylinder{wood=Модрина, weight=0.31101772}
Waste{weight=9.2}

Загальна вага: 33.960
```

За допомогою класу TestByConsole:

```
/ / / Timber management system \ \ \

1: Add Wood
2: Add Product
3: Calculate total weight
4: Exit
1

Id: 12
Name: Quercas
Density: 3.3

Wood added
Каталог деревини:
Wood{id=1, name='Модрина', density=1.1}
Wood{id=2, name='Ялина', density=0.9}
Wood{id=3, name='Сосна', density=0.7}
Wood{id=12, name='Quercas', density=3.3}

1: Add Wood
2: Add Product
3: Calculate total weight
4: Exit
2

What do you want to add?
1: Timber
2: Cylinder
3: Waste
2

Wood Id: 12
Length: 10.2
Diameter: 3.1

Cylinder added
Перелік виробів:
Cylinder{wood=Quercas, weight=254.05475}
```

Продовження:

```
1: Add Wood
2: Add Product
3: Calculate total weight
4: Exit
2

What do you want to add?
1: Timber
2: Cylinder
3: Waste
3

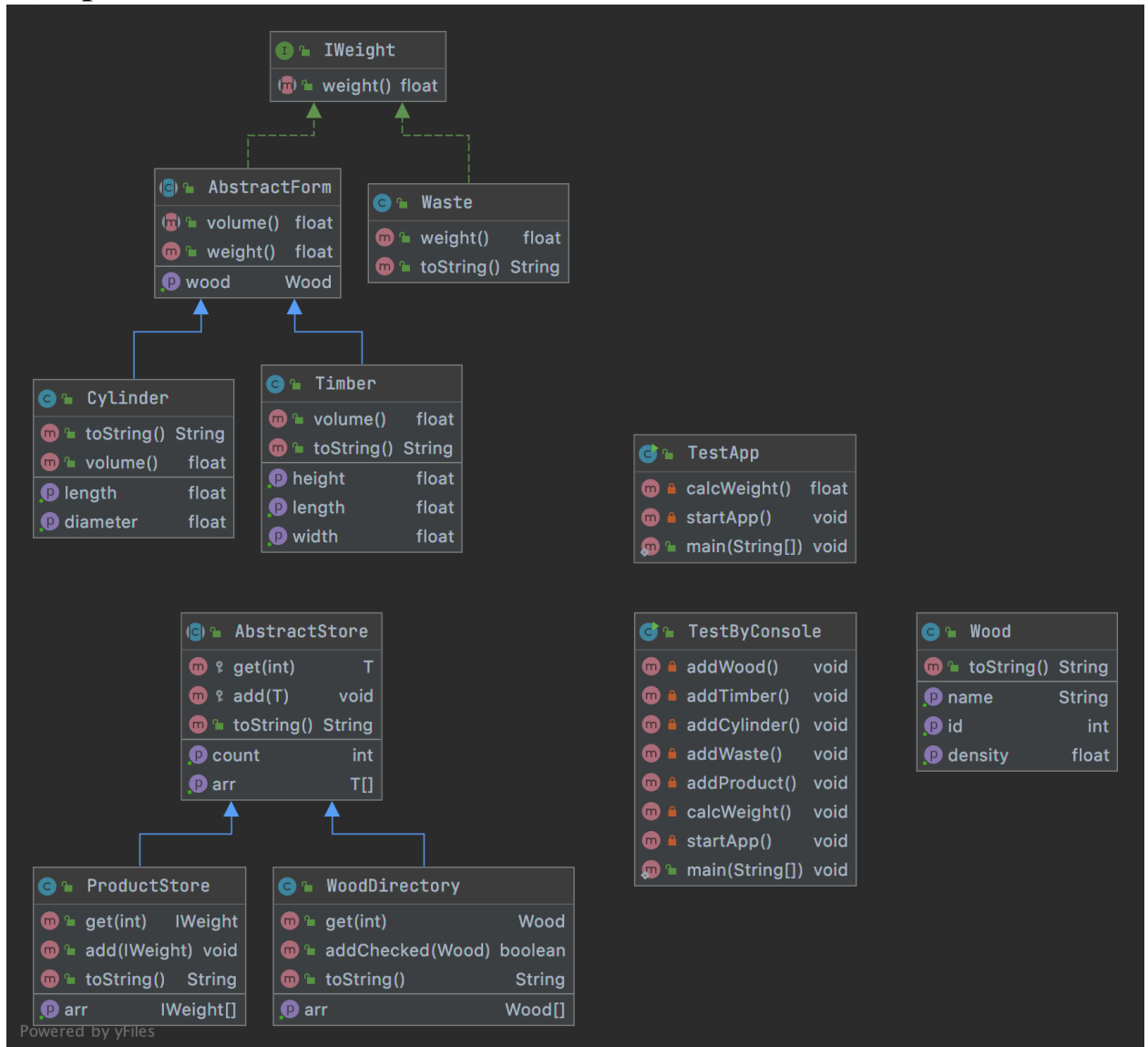
Weight: 1.44

Waste added
Перелік виробів:
Cylinder{wood=Quercas, weight=254.05475}
Waste{weight=1.44}


1: Add Wood
2: Add Product
3: Calculate total weight
4: Exit
3

Total weight: 255.495


1: Add Wood
2: Add Product
3: Calculate total weight
4: Exit
4
```

Діаграма класів:**Висновки:**

На цій лабораторній роботі я модифікував додаток для обліку інформації про виробу, додавши можливість зберігати циліндри та відходи. Під час виконання роботи я навчився працювати з наслідуванням та поліморфізмом у Java.