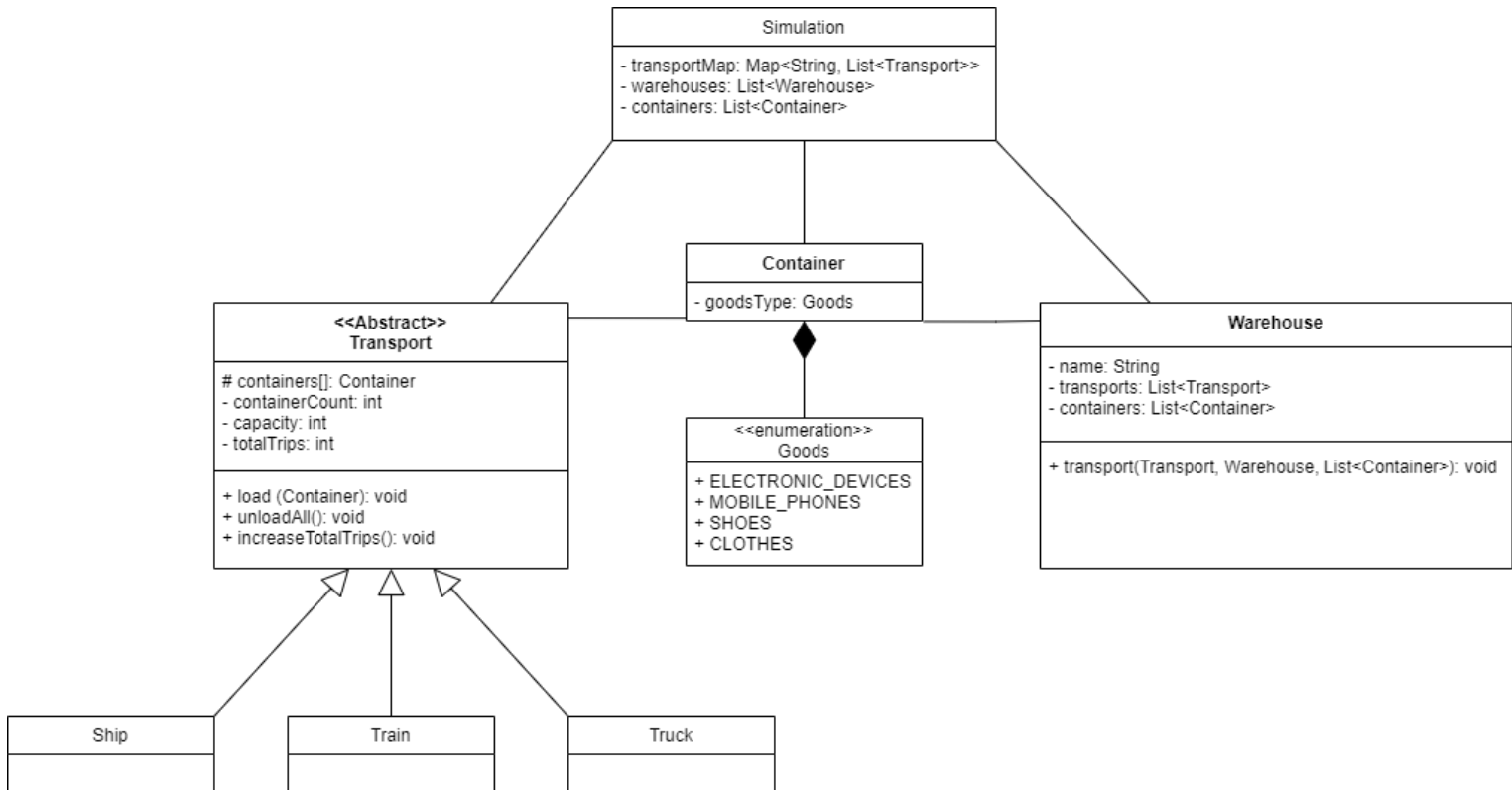


Programowanie obiektowe. Zadanie 2 – Firma transportowa

Viktoryia Semianiuk 46479

Do rozwiązania tego zadania zostały stworzone 8 klas, które opisuje poniższy diagram:



Najważniejsze metody stworzone w tym zadaniu:

- metoda transportująca w klasie Warehouse

```
public void transport(Transport transport, Warehouse destination,
List<Container> containers) {
    System.out.println("Transport containers from " + this.name + " to "
+ destination.getName() + ".");
    for (Container container : containers) {
        transport.load(container);
    }
    transport.unloadAll();
    destination.setTransports(transports);
    this.setTransports(null);
    destination.getContainers().addAll(containers);
    Transport.increaseTotalTrips();
}
```

- metody załadunkowe w klasie Transport

```
public void load(Container container) {
    if (containerCount < capacity) {
        containers[containerCount] = container;
        containerCount++;
        System.out.println("Loading container with goods of type " +
            container.getGoodsType() + " on a " + getClass().getName());
    } else {
        System.out.println("Transport is full, cannot load more
            containers.");
    }
}
```

oraz

```
public void loadAll(List<Container> containers) {
    int containersToLoad = Math.min(containers.size(), capacity -
        containerCount);
    for (int i = 0; i < containersToLoad; i++) {
        Container container = containers.get(i);
        this.containers[containerCount] = container;
        containerCount++;
        System.out.println("Loading container with goods of type " +
            container.getGoodsType() + " on a " + getClass().getName());
    }
    System.out.println("Transport is full, cannot load more
        containers.");
}
```

- metody rozładunkowe w klasie Transport

```
public void unload(Container container) {
    if (containerCount > 0) {
        for (int i = 0; i < containerCount; i++) {
            if (containers[i] == container) {
                for (int j = i; j < containerCount - 1; j++) {
                    containers[j] = containers[j + 1];
                }
                containerCount--;
                System.out.println("Unloading container with goods of
                    type " + container.getGoodsType() + ".");
                break;
            }
        }
    } else {
        System.out.println("Transport is empty, cannot unload any
            containers.");
    }
}
```

oraz

```
public void unloadAll() {
    if (containerCount > 0) {
        for (int i = 0; i < containerCount; i++) {
            System.out.println("Unloading container with goods of
```

```
type " + containers[i].getGoodsType() + ".");
    }
    containerCount = 0;
} else {
    System.out.println("Transport is empty, cannot unload any
containers.");
}
}
```

- statyczna metoda licząca kursy za pomocą statycznego pola

```
public static void increaseTotalTrips(){
    totalTrips++;
}
```

Wynik kompilacji programu:

```
Total trips: 83
```

```
Process finished with exit code 0
```