Programming 2:
Assignment 4

Jibbe Andringa
s2336219
Group 12

March 10, 2023

1 Package Inheritance Hierarchy

1.1 Introduction

The goal of the package assignment is to implement a simple object-oriented program for calculating the cost of shipping different types of packages. The program offers two different shipping options: TwoDayPackage and OvernightPackage. The program utilizes a Package class as the base class for the different package types, with functions for calculating the cost of shipping and storing information about the sender, receiver, and weight of each package.

1.2 Class Diagram

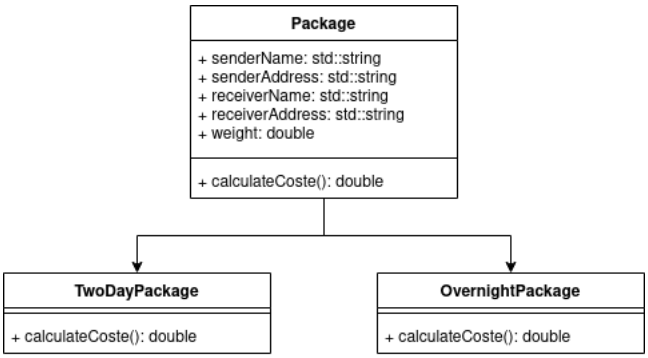

Figure 1: Class diagram for the Package classes

1.3 Design Decisions

- The program uses a class hierarchy to represent the various types of packages, with the base class Package containing virtual functions for calculating the cost of shipping and destructing the object. Having a virtual destructor prevents memory leaks by ensuring that the derived destructor is called first.

- Two derived classes are created, TwoDayPackage and OvernightPackage, which inherit from the base class Package and override the calculateCost function to provide specific cost calculations for each type of package.

- The program creates a list of packages with different shipping options and passes this list to a function printCosts, which loops over the list and prints the cost of shipping for each package by calling the package's member function calculateCost.
- The program utilizes C++11's override specifier to ensure the correct function is called at runtime.

1.4 Compiling and Running the Program

To compile the program, navigate to the directory containing the source code and type the following command into the terminal:

```
g++ main.cpp src/*.cpp -o packages
```

This will compile the program and create an executable file called 'packages'. To run the program, type the following command into the terminal:

```
./packages
```

The program will create a list of packages, calculate the cost of shipping for each package, and print the results to the console.

1.5 Conclusion

In conclusion, the implementation of the package program provides a simple and flexible tool for calculating the cost of shipping different types of packages. The use of a class hierarchy and virtual functions simplifies the implementation of new package types and makes the code more readable and maintainable. The program demonstrates the functionality of the derived classes by providing specific cost calculations for each type of package

2 Customer hierarchy

2.1 Introduction

The goal of the customer hierarchy assignment is to create a system that can manage customer information and their corresponding package subscriptions. The program uses a Customer class to store customer information and a Package class to store package information. Customers can subscribe to multiple packages, and the program should be able to display the customers for a given package.

2.2 Class Diagram

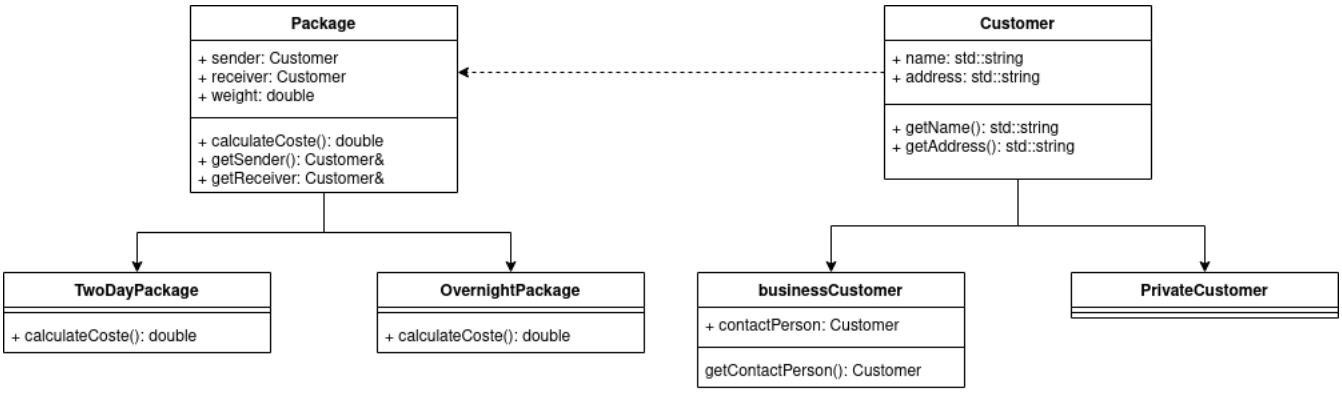

Figure 2: Class diagram for the Customer and Package classes

2.3 Design Decisions

- The program uses two classes: a Customer class and a Package class. The Customer class contains information about each customer, such as their name, address, and account balance. The Package class contains information about each package, such as the sender, recipient, delivery status, and weight.

- The list container was chosen for this program because it allows for efficient insertion and deletion of elements at any position, which is important for a program that needs to add and remove customers and packages frequently.

- Unordered set was used in this program to efficiently store unique elements of a large dataset without any specific order. Since searching for an element in an unordered set has an average time complexity of O(1), it provides faster search operations than other container types like vectors or lists. This makes it a suitable choice for this program that requires frequent lookups of customer IDs and package IDs.

2.4 Compiling and Running the Program

To compile the program, navigate to the directory containing the source code and type the following command into the terminal:

```
g++ -o customers $(find . -name "*.cpp")
```

This will compile the program and create an executable file called 'customers'. To run the program, type the following command into the terminal:

```
./customers
```

The program will prompt the user to enter a key to be searched for in the tree. If the key is present in the tree, the program will output a success message, otherwise, it will output a failure message. The program will then output the tree in a graphical manner.

2.5 Conclusion

In conclusion, the implementation of the customer hierarchy program provides a useful tool for simulating a package delivery service. The program demonstrates the functionality of the Customer and Package classes by allowing printing various information about the customers and packages. The use of lists and unordered set simplifies the implementation of the program and makes the code more readable and maintainable.

3 Questions

a Inheritance is used in the package delivery assignment to allow different types of packages (such as TwoDayPackage and OvernightPackage) to share common attributes and methods of the parent Package class, while also allowing them to have their own unique features.

b The protected access specifier allows derived classes to access the members of the base class, but not other classes outside the hierarchy.

c It is important to be careful because it can lead to the Diamond Problem, where a class derives from two classes that have a common base class.

d Inheritance allows for creating new classes that inherit the properties and behaviors of existing classes, while composition involves creating new objects that contain instances of other classes as components. For example, a computer class could inherit from an electronic device class, while also containing instances of a processor class and a memory class.

e The compiler inserts calls to destructors at the end of the scope where an object is declared, or when the delete keyword is used to free the object from the heap.

f A virtual destructor is necessary when a class has virtual functions and is used as a base class, as it ensures that the destructor of the derived class is called when an object of the derived class is destroyed through a base class pointer.