



## Uppgift 3

DA339A

Viktor Vallmark, id: aq2560, group 15

Written by *Viktor Vallmark*

February 3, 2025

# 1 Reflections and motivations

## 1.1 Reflections about the design

I used an interface to handle the implementation of some functions including how to calculate the price in the Cake-class. I used an interface instead of abstract class or other forms of generalization because the bakery goods were too different from each other. The Cake class is more intricate than the bakeryitem class so just having an interface to describe how to calculate price was needed. I used polymorphism to handle the ArrayList for the current order by using the interface as a parameter to the ArrayList. This way I could both have bakery items and cakes in the same ArrayList.

The pros of this implementation is that polymorphism is enabled by a single interface which makes the program easy to read. Cons of this implementation is that if people want to add more intricate classes to handle future items in the bakery, the interface have to be extended or an abstract class with some abstract functions, maybe for calculating the price, have to be implemented. Subclasses of this abstract class should reflect the items the bakery is going to sell.

## 1.2 Reflections about the difference between my implementation and other groups implementation

My first implementation included an abstract class to handle the BakeryItems. I didn't use an interface at first. Most other groups used only an interface. I used an abstract class to handle BakeryItems because, contrary to what the task said, I made BakeryItems into several subclasses with enough differences to warrant having a abstract superclass that BakeryItems inherited from. I made a different class for Cookie, Danish and Rye.

After some reflection and talking with others and getting feedback, I made BakeryItem much much more simple with only two fields, name and price. This way I could remove the whole abstract class and the subclasses that inherited from that abstract class and just have every item that is being sold by the bakery implement the interface instead. This way made the class hierarchy way simpler while still allowing polymorphism to help me with my currentOrder ArrayList.

## 1.3 Different questions

1. How did you implement polymorphism? 2. how did you solve customcakeframe? 3. How did you solve update the strings and order arrays so that it updated correctly?

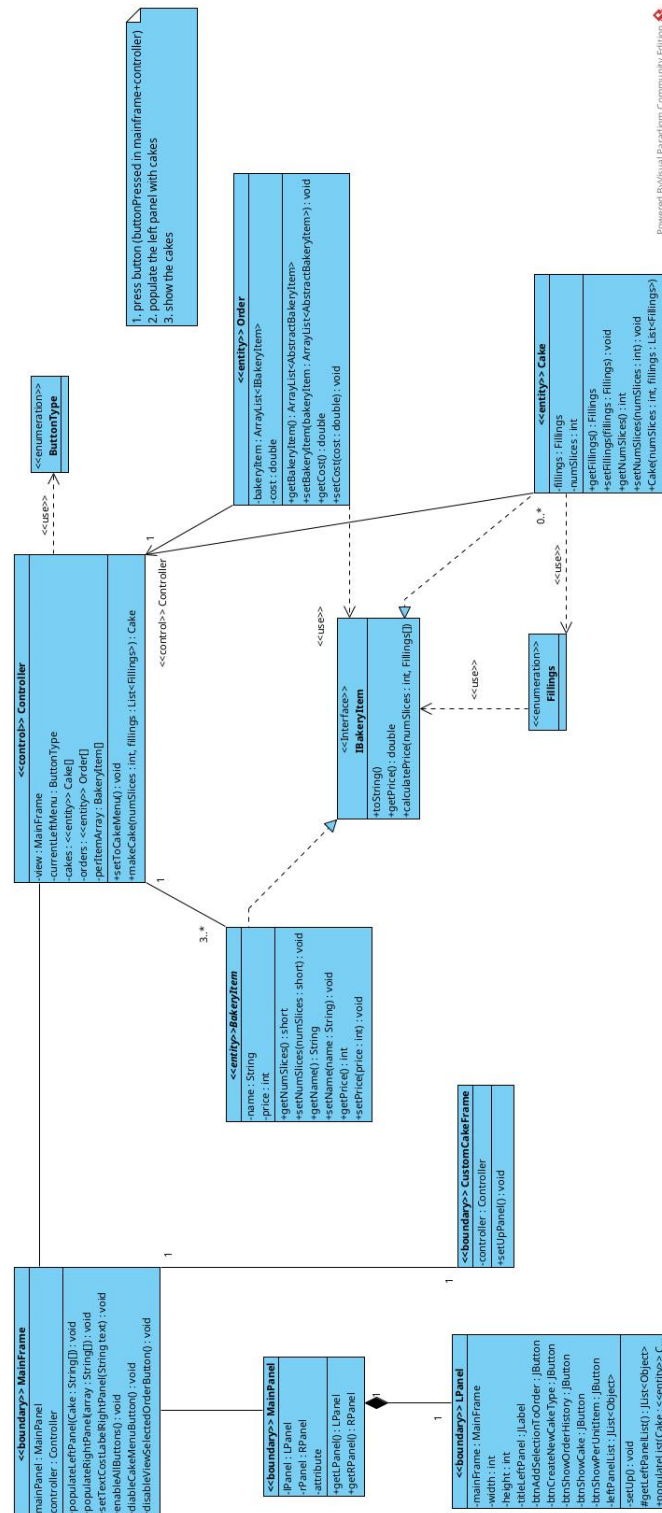


Figure 1: Activity diagram

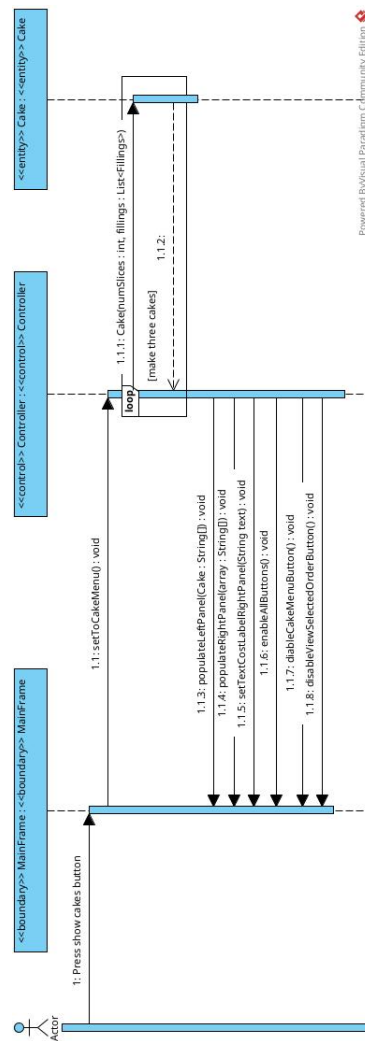


Figure 2: Sequence diagram