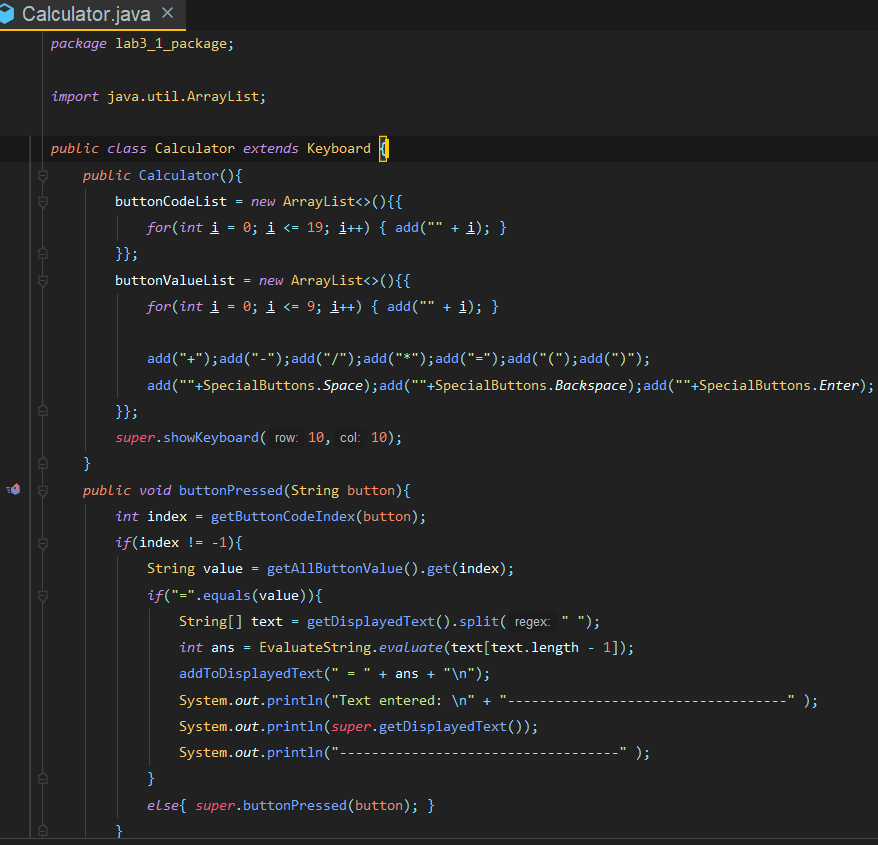
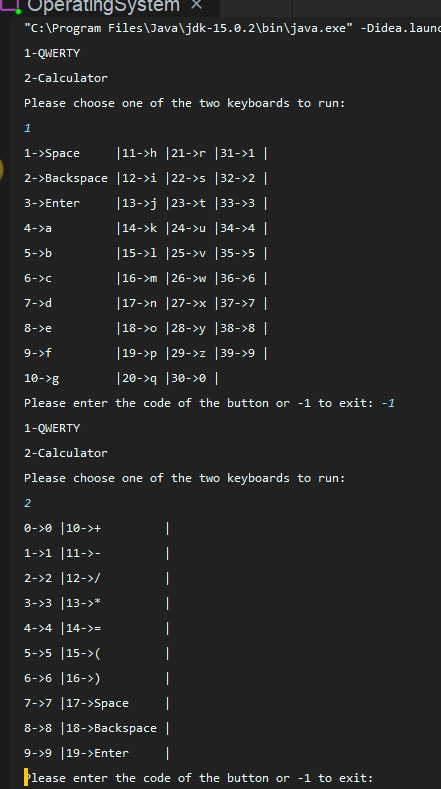
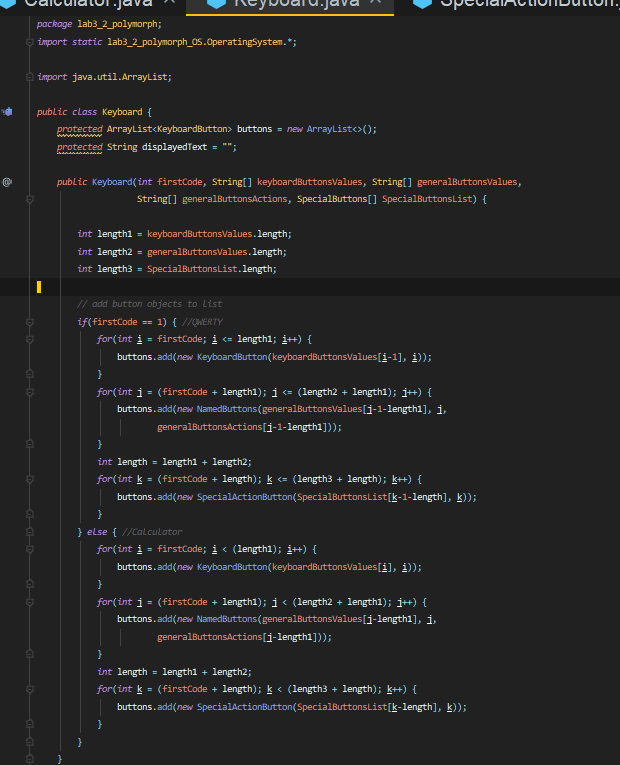
**Part 1:**

**:**



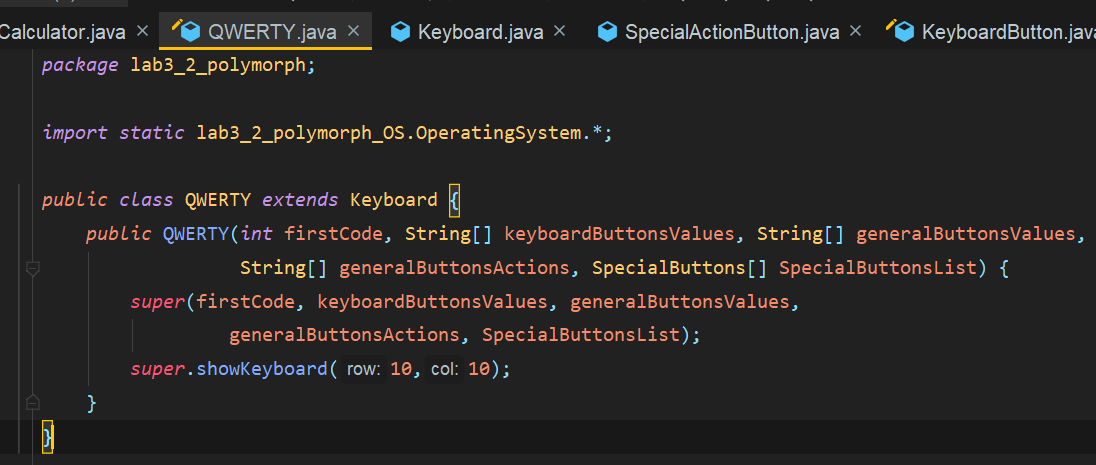
****

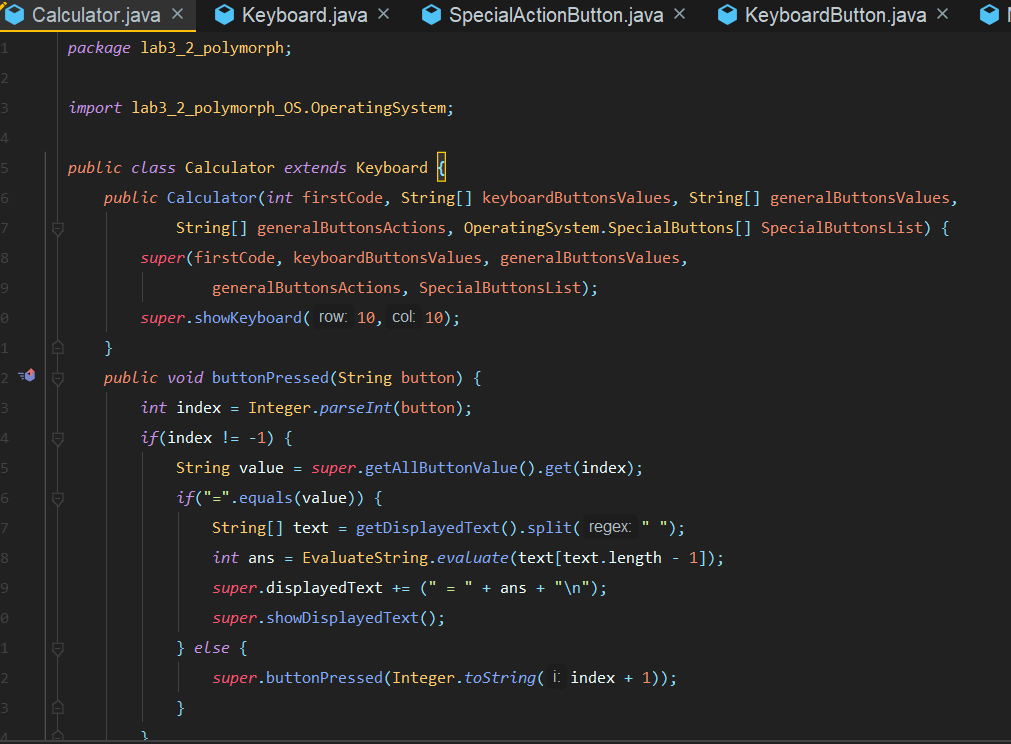
**Part 2:**

**Keyboard Class**

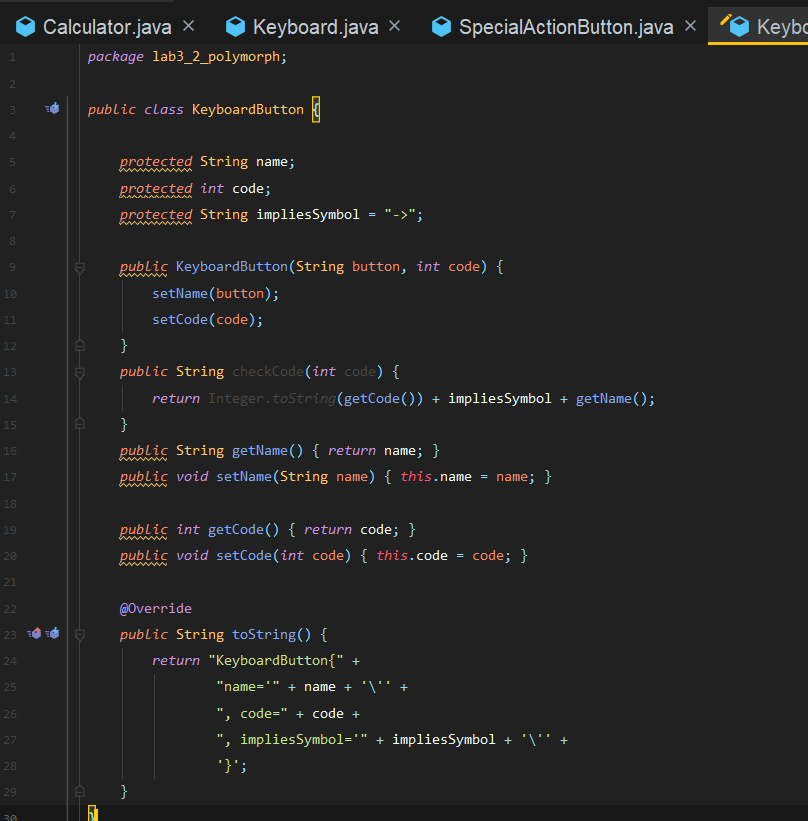


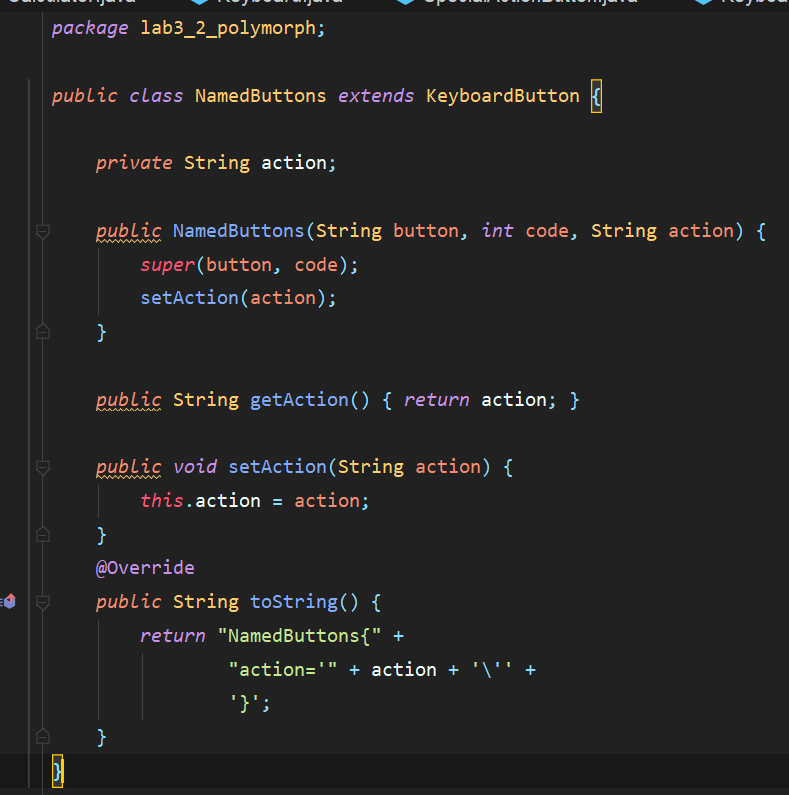


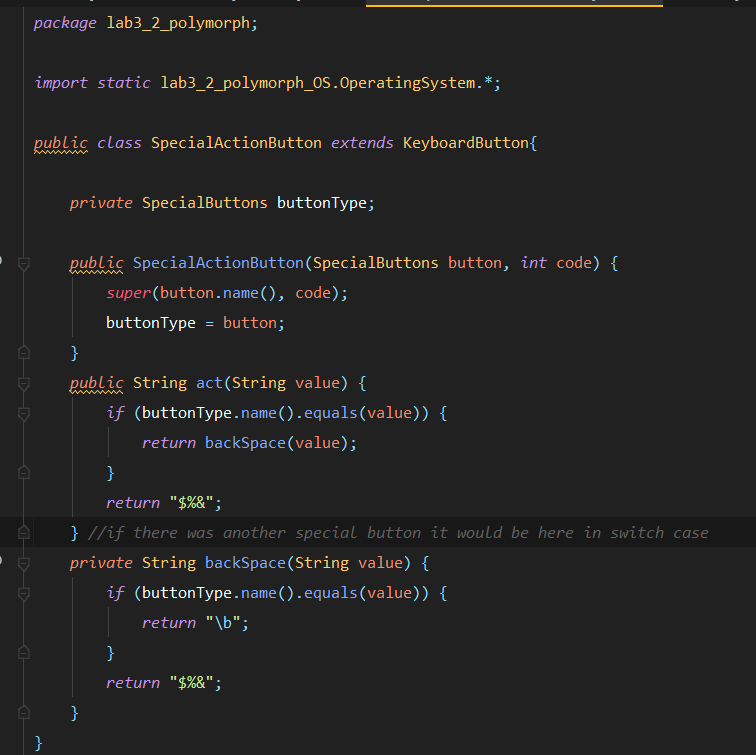
**QWERTY class**

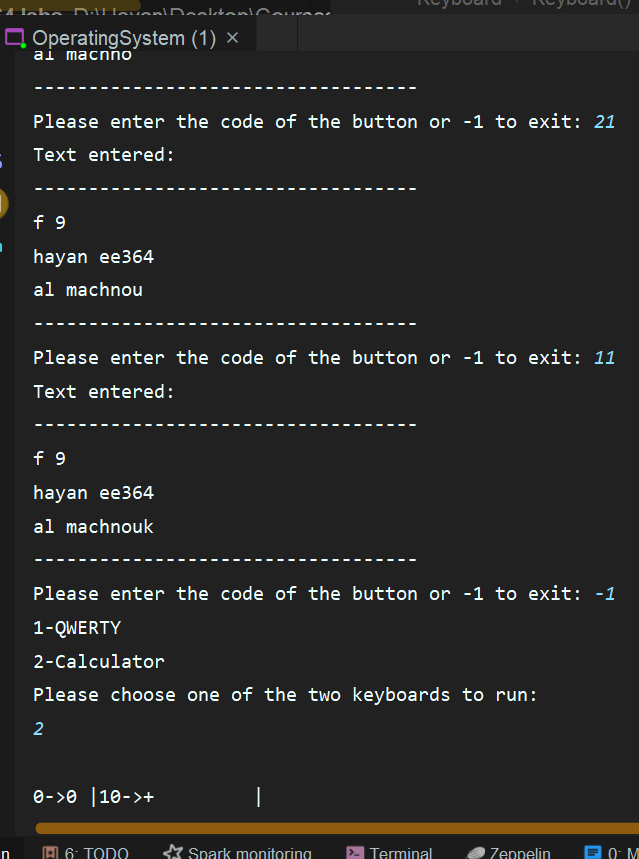
**Calculator class**

**KeyboardButton class**



**NamedButton class**

**SpecialBActionButton class**



**Part 3:**

**Inheritance aids in the reuse of code. The parent class's code can be reused by the child class without having to rewrite it.**

**Because the primary code does not need to be written repeatedly, inheritance can save time and effort.**

**Inheritance gives a straightforward model structure that is simple to comprehend.**

**We can override the parent class's methods with inheritance so that the child class can create a meaningful implementation of the base class method.**

**As a result, all of the above will ultimately result in less maintenance work and lower overhead costs.**

**In addition, the base class can choose to keep some data private so that the derived class cannot access or modify it except if is of protected type.**

**As an example of inheritance, consider the relationship between a parent and a kid. The qualities of parents, such as hands, legs, eyes, and nose, as well as behaviors such as walk, talk, eat, and sleep, are inherited in the child, allowing the child to use/access these properties and behaviors whenever necessary.**

**Both parent and child can have particular or private properties in addition to some common qualities and behavior. For example, parent and child can have specific properties such as blood group, date of birth, and specific behavior such as one playing cricket or another activity while the other does not.**