BÁO CÁO

1. 1 vài kí tự

A screenshot of a computer

Description automatically generated with medium confidence

1. Thiết kế

Calculator được thiết kế dựa trên mô hình MVC(model view controller).

* View bao gồm view của old display và entry display, menuBar và keyboard. Tớ tạo ra interface ResultDisplay bao gồm 2 method: getOldDisplay() và getDisplayEntry sẽ trả về 2 Label vẽ lên trên màn hình. Tớ tạo ra 2 class implements cái interface này cho 2 chế độ : chế độ standard và chế độ sciencific nhưng tớ không đủ time để làm cái sciencific. Interface ResultDisplay này extends ModelObserver interface, để mà mỗi khi có sự thay đổi từ model( class Model expression) sẽ thông báo cho display và cập nhật dựa vào message.

Diagram

Description automatically generated

Diagram

Description automatically generated

Phần view của keyboard do interface Keyboard bao gồm method: buildButton(): MyButton[], method này sẽ trả về toàn bộ những button cần thiết cho máy tính theo thứ tự từ trái qua phải, từ trên xuống dưới. Bởi vì người ta cũng xây dựng button hoàn chỉnh rồi nên tớ extends từ JButton(java.swing ) và implements KeyboardPublisher, ActionListener interface. KeyboardPublisher có method notifyObservers() sẽ thông báo bất cứ cử chỉ bấm nào từ bàn phím của máy tính đến ModelExpression( implements KeyboardObserver) cho việc xử lý.

Phần view về menuBar tớ chưa làm kịp, chỉ cho thêm vào cho đẹp thôi :>

- Ở đây, tớ áp dụng strategy pattern vào calculator này, để sau này nếu mình muốn mở rộng thêm như là tạo thêm máy tính chế độ sciencific, hay programming thì không cần phải sửa code cũ mà chỉ cần implements các interface rồi có thể thay đổi chế độ ở run time được.

Diagram

Description automatically generated

+ Phần model có interface là modelExpression, interface này extends từ 2 interfaces khác là KeyboardObserver và modelPublisher để lắng nghe sự thay đổi từ keyboard sau đó truyền đến màn hình (resultDisplay).

+ Phần model này tớ mới chỉ làm kịp cho chế độ standard. Vì là nếu để mỗi class StandardExpression xử lý hết phần logic toán tử và toán hạng thì rất không hợp lý nên tớ nghĩ đến việc chia công việc cho các class khác nên tớ sử dụng State pattern ở đây.

Diagram

Description automatically generated

IntegerState: state khi mà user click vào cái chữ số, và là state đầu tiên.

OperationState: state khi mà user click vào cái dấu +, -, \*, /, sqrt, 1/x, negative

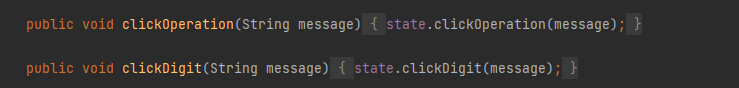
ResultState: state khi user click =.

DecimalState: state khi user bắt đầu click dot.

+ Khi user sử dụng máy tính, họ phải ở 1 trong 4 state này cũng giống như là việc StandardExpression phải ở 1 trong 4 state này :

Text

Description automatically generated with low confidence

Tính đa hình được thể hiện trong các method clickDot(), clickDigit(), … này 

Nó sẽ thay đổi được state ở runtime tùy vào việc user click.

1. Exception:

Khi user cố gắng chia 1 số cho 0, thì sẽ throws ArithemticException

Khi user cố gắng lấy sqrt của một số âm sẽ throws ra NotANumberException.

Text

Description automatically generated

1. Vận dụng các kiến thức đã học như thế nào?

* Tính đóng gói được thể hiện ở chỗ mọi field trong các class đều được để private và dùng getter và setter để lấy dữ liệu
* TÍnh kế thừa thể hiện ở chỗ các interface kế thừa interface khác hoặc MyButton kế thừa JButton
* Tính đa hình thể hiện trong sự thay đổi của state trong class standardExpression hay là sự thay đổi cấu hình của các display, keyboard
* Tính trừu tượng thế hiện trong sự hiện diện của các interface.