# Assignment 3

Name: Milind Shaileshkumar Parvatia

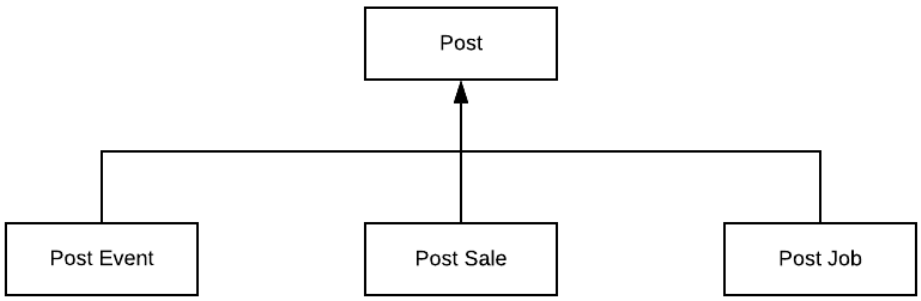
Id: s3806853

## Task 1

### A.) Inheritance is an important and powerful technique in Object-oriented programming. Describe all important inheritance relationships in your Unilink GUI application which you have implemented in Assignment 2. Draw simple class diagrams using the classes you implemented in Assignment 2 to illustrate the inheritance relationships you have described.

In my Assignment 2 main I have used following in heritances:

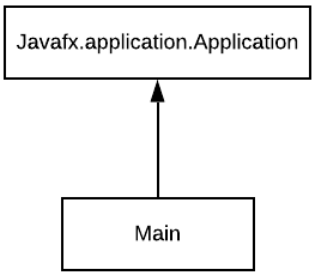
**Model Inheritance Diagram**




**Relationship 1:**

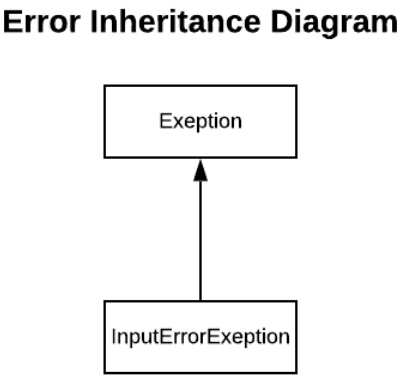
- PostEvent extends Post
- PostJob extends Post
- PostSale extends Post

**Javafx Inheritance Diagram**



**Relationship 2:**

- Main extends javafx.application.Application




**Relationship 3:**

- InputErrorExeption extends Exception class

## B.) What are the benefits of inheritance? Give specific examples from your implementation to show the benefits of using inheritance in your application.

- In my project using inheritance I'm creating loosely coupling and high cohesive design.
- I'm taking common variables from all the three classes and assigning them to post class and inheriting post class.
- This make managing types of class's goal very easy.
- Example, since all types of event, sale, jobs are inherited by post I was able to create HashMap of single post class to manage all three post types.

# Task 2

## A.) Polymorphism is another key conception programming language, especially in Object - Oriented programming languages. Describe how polymorphism is implemented in your Unilink GUI application. Give specific examples from the code you have implemented to show the polymorphic method calls. Explain your examples.

1. In my project I have created class level polymorphism with Post model class, which is being extends by postEvent, postJob, postSale.

2. I'm creating HashMap with key as string and value as **Post** calling it PostMap in mainWindowController.

3. Using this HashMap, I can add new posts using any **postEvent, postJob, postSale** type of object to object of post and to get them I can use type cast.

## B.) What are the benefits of polymorphism in your implementation?

- Main benefit of using polymorphism is its easier when you have to work with subclass and have to relate to each other, it reduces code redundancy.

- In my example if I didn't apply class level polymorphism on HashMap then I have to create 3 HashMap's with each and every post's type and have to manage 3 different HashMap which I can do easier with single variable.

# Task 3

## A.) Explain all the pros and cons of the design of your Unilink GUI application in terms of extensibility and maintainability. What are the features in your code that make it easier to extend your application in the future, for example, to add a new type of post.

- I have made post class extends all the post type class so if it is required to add new type of post to project it will be successfully done by extending Post class.

- In GUI, I have created mainGrid variable generic so if new type of post is added then it will be function as for function I'm using for event, job and sales post.

- I have designed whole project with high cohesive and low coupling in mind so when extending new feature or creating whole new file we can easily attach new module to working project and make that easier though private variables and their public getter setter methods.

- Adding new component to any of post will be also easier since all types of posts are independent class extending Post class, they don't have any dependency.

- Adding new feature in GUI will be also easier since all the java files I have created are based on name of window screen and it is well managed file with their package strucutre.

## B.) Describe the trade-off between coupling and cohesion (refer to Week 4 lecture) in your design. You need to include specific examples from your code in your answers.

### Coupling:

- In my assignment I have used loose coupling every time possible like encapsulating data like,

- Declaring every variable private in their class and creating public getter and setter method possible for those variable calling.

- In class Post when I have created getter setter and In their inherited classes Event, Jobs, Sales I am using those getter setter to assign values and using super( ) method.

- Other than I'm using getter setter throughout my assignment whenever I'm assigning values to any class variable to make coupling as loose as possible.

### Cohesion:

- I have designed my project with high cohesion in mind and assigns all class to their specific class goals, like Database, mainWindowController, login, CreateNewPost etc.

- I have created all the methods within those class and using getter setter methods to call and pass variables to maintain high Cohesive ability throughout project.

- This way in future if I have to add new goal, I can easily attached new class like a module to this project and manage it esily.