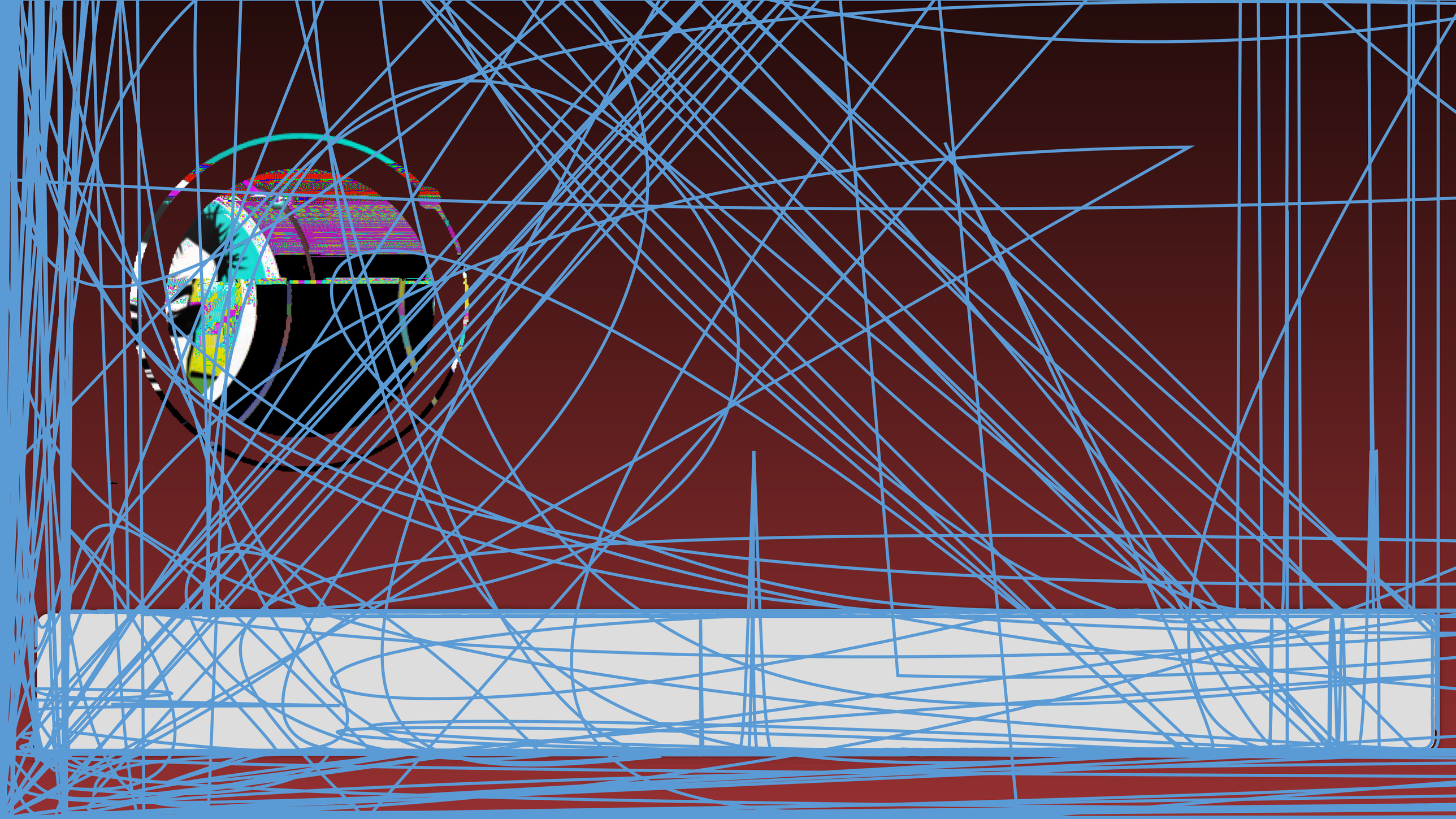
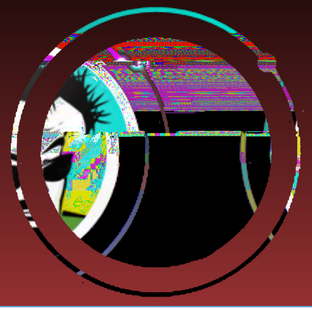
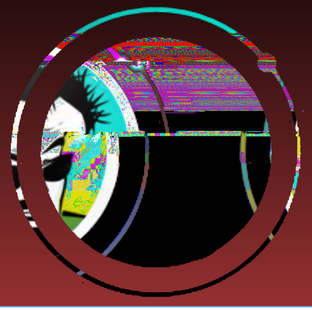




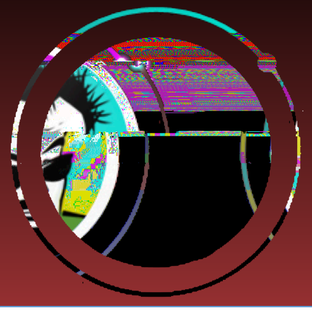
JPA Inheritance

- Inheritance is a Object Oriented concept where types can inherit properties and behaviors from other types
- The Relational paradigm used by Relational Databases does directly not support the concept of Inheritance
- In a JPA context we are looking at how inherited properties can be mapped into the relational model of the database



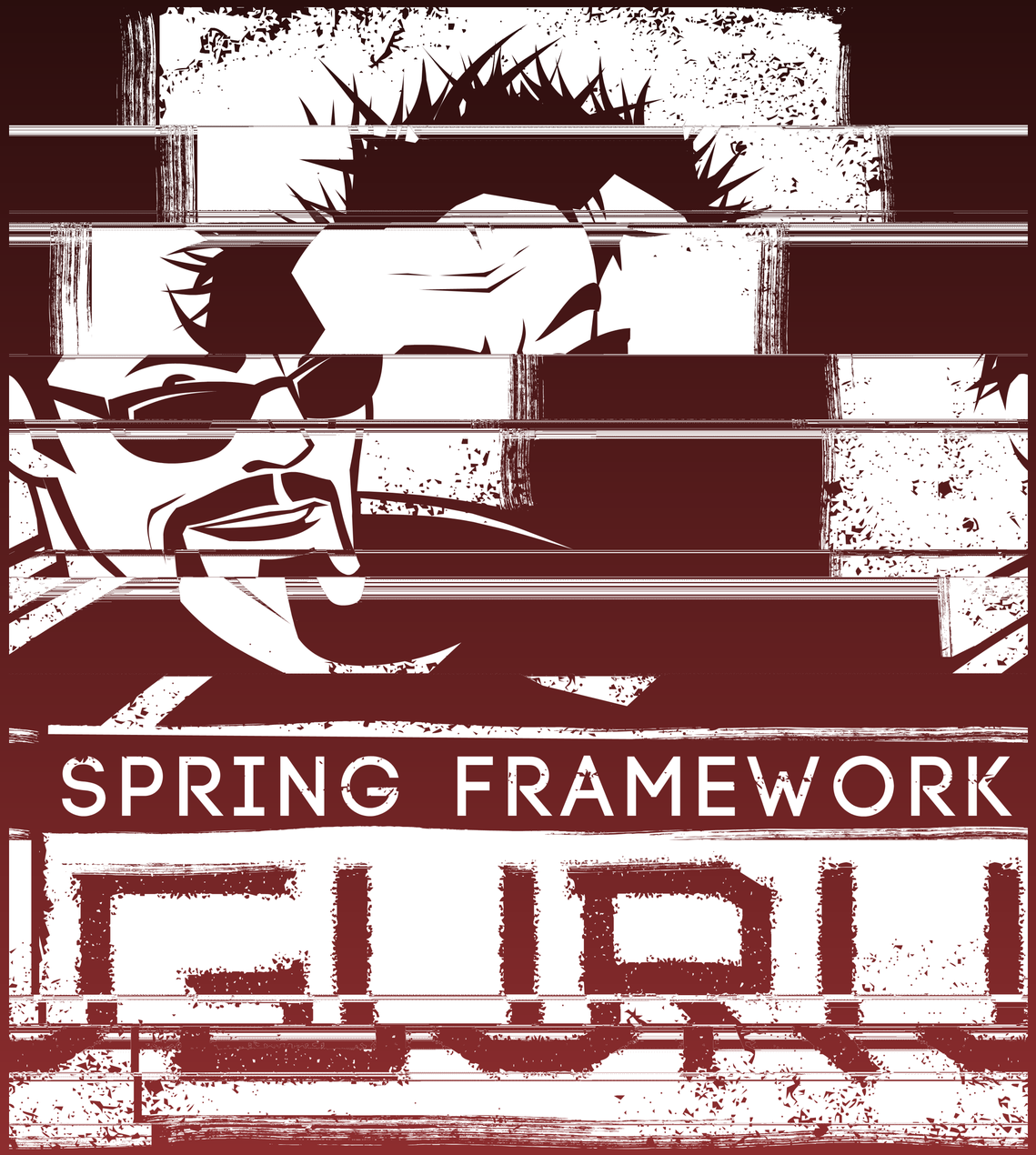
JPA Inheritance Strategies

- **MappedSuperclass** - Abstract parent class, parent is not in the database
- **Table per Class** - Similar to MappedSuperClass, but each type is assigned a database table
- **Single Table** - Single table is used, has properties for all types, uses discriminator value to determine the type database row is mapped to
- **Joined Table** - Each type has a table, sub-types are joined



JPA Inheritance Which to Use?

- **MappedSuperclass** - Good for common entity properties
- **Table per Class** - Best option for performance, optimal for most use cases
- **Single Table** - Best when you need to make polymorphic queries, but can't use not null constraints on subclass attributes & can lead to data inconsistencies
- **Joined Table** - When you need polymorphic queries and need data consistency (you **can** use Not Null constraints on subtypes)


SPRING FRAMEWORK
GURU