

I ended up designing and implementing my project a bit differently than I expected to. It is still the same concept. The program is meant to be used to help manage a food vendor business. To begin, I implemented my vendor business project to work in the following way. Everything is centered around the VendorBusiness class, which has-a relationship with several of the other classes. It contains methods to help manage the business. Besides its own unique properties, it has an address, an owner, employees, known event hosts, events, and a menu consisting of different menu items. All the people (owner, employees, hosts) inherit from the Person class. I switched from using array lists to hash tables to store the employees, hosts, and menu items because it is much easier to add and remove specific objects from a hash table. All items that may be stored in hash tables are assigned a random five-digit ID upon their initialization so that it can be found and removed easily. The VendorBusiness class itself uses the design pattern of a singleton because there can only be one vendor business. I also used the design pattern of abstract methods when creating my menu items. Both the FoodItem and DrinkItem implement the abstract methods from their parent class of MenuItem. They also each have an Enum object of a different type. Meaning FoodItem has-a Food Enum and DrinkItem has-a Drink Enum. I used Enums to store the names of food and drink items, but not their prices since they can change per event. The MenuItem class uses generics to adjust the prices of differing items, as well as, to be used polymorphically to work with different unknown kinds of MenuItem objects. One more thing to mention is that all, but the MenuItem, FoodItem, DrinkItem, and Enum classes have an address from the Address class.

The driver program creates a vendor business with all of the things mentioned above and contains comments that tell scenarios of what happens at the events the business attends and what the business does after each one.