|  |  |  |
| --- | --- | --- |
| Class | Fields (Variables) | Methods |
| Shoe | 1. protected FINAL int productNumber; 2. protected FINAL String name; 3. protected ShoeSizeList shoeSizeList; 4. protected int totalQuantity | **This class will not have any methods, just the three-argument constructor so that data can be accessed directly instead of using getters or setters.** |
| ShoeSizeList | 1. float shoeSize; 2. int quantity; | 1. **boolean add(float shoeSize, int quantity)** 2. **boolean decrease(float shoeSize, int quantity)** 3. **String traversal()** 4. **int get (float shoeSize)** |
| ShoeTable | 1. private HashTable shoeTable; | 1. void addShoe(Shoe shoe) 2. String getShoeName(Shoe shoe) 3. int getShoeSize(Shoe shoe) 4. int getProductNumber(Shoe shoe) |
| HashTable  (Implements HashTableADT) | 1. private int numItems; 2. private Shoe<generics>hashArray[]; | 1. public double getLoadFactorThreshold() 2. public double getLoadFactor() 3. public int getCapacity() 4. public int collisionResolution() 5. int hash(Shoe shoe) 6. int numItems() |
| Interfaces | Fields (Variables) | Description |
| ShoeTableADT | 1. void addShoe(int productNumber, String name, float shoeSize, int quantity) 2. ShoeInfo lookupShoe(int productNumber) 3. String checkSize(int productNumber) 4. boolean deleteShoe(int productNumber, float shoeSize, int quantity) | 1. In addShoe methods, if productNumber already exits, add shoeSize and quantity to that Shoe class. If productNumber does not exit, consturuct a new Shoe class 2. Class ShoeInfo: contains three parameters, int productNumber, String name, int totalQuantity 3. checkSize returns a String like “7.5(3) 9.5(8) 10(4) 10.5(2)” 4. delete just decrease quantity of that shoeSize. If quantity to be deleted is larger than current, it will return false |
| HashTableADT | 1. public double getLoadFactorThreshold() 2. public double getLoadFactor() 3. public int getCapacity() 4. public int collisionResolution() 5. int hash(Shoe shoe) 6. int numItems() | N/A |