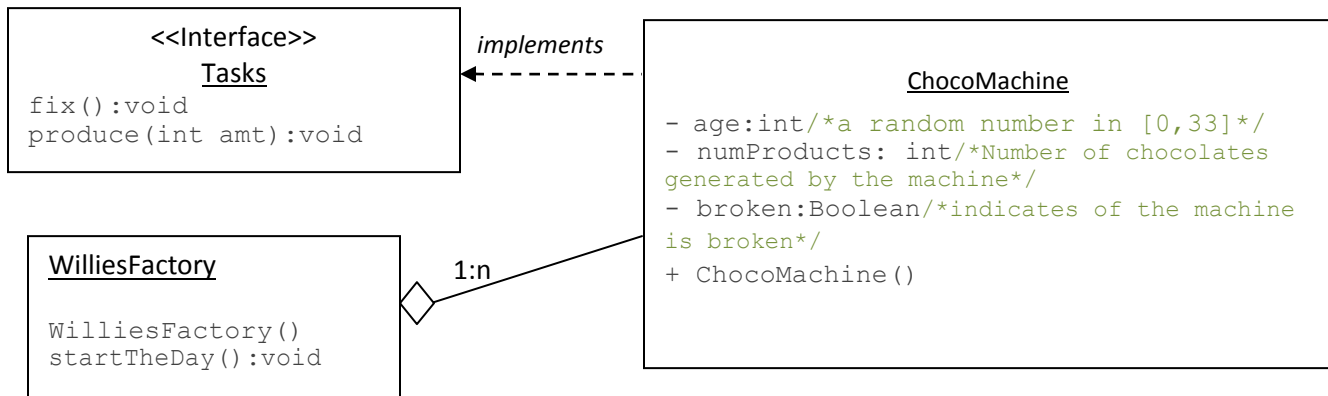


Question: Willie's Chocolate factory has different chocolates machines at work in the factory. There are 10 chocolate machines, each machine produces chocolates until it gets broken.



PartA: Read the below details to create the above classes and interfaces.

- The constructor **ChocoMachine()** creates a new chocolate machine. Its age is a random number in [0,33].
- Fix()** method fixes a Machine. It changes the broken value to false.
- Produce(int amt)** method takes the number of chocolates to be produced and increases the "numProducts" if the machine is not broken. If the number of products becomes multiple of 7, the machine is broken.
- startTheDay()** method generates for each machine in the factory, a random number that is the number of products to be produced then calls the produce method.

PartB: Like in real life, some problems are worth fixing while others are not. Willie looks after his machines, and if there's something worth fixing he fixes it and gets it up and running again, otherwise the machine is thrown away. Note the **BrokenMachineException**:

- It extends Exception so we can throw it. It holds some extra data (the machine's age) and an extra method (`isItWorthFixing()`) that returns true if the age of the machine is less than 10.
- The constructor **BrokenMachineException(int age)** creates its own error message that is more explanatory than the standard. The message is "This machine is broken and its age is ...".
- Note that the method "**produce**" throws an exception if the machine is broken. The method "**startTheDay**" handles the exception and tries to fix the machine if it is worth fixing.

PartC: Write a main program to create a factory and start the day.