1.

RatNum(int n):	CREATOR
RatNum(int n, int d):	CREATOR
boolean isNaN():	OBSERVER
boolean isNegative():	OBSERVER
boolean isPositive():	OBSERVER
int compareTo(RatNum rn):	OBSERVER
double doubleValue():	OBSERVER
int intValue():	OBSERVER
float floatValue():	OBSERVER
long longValue():	OBSERVER
RatNum negate():	PRODUCER
RatNum add(RatNum arg):	PRODUCER
RatNum sub(RatNum arg):	PRODUCER
RatNum mul(RatNum arg):	PRODUCER
RatNum div(RatNum arg):	PRODUCER
int gcd(int a, int b):	PRODUCER
int hashCode():	OBSERVER
boolean equals(Object obj):	OBSERVER
String toString():	OBSERVER
RatNum valueOf(String ratStr):	OBSERVER

2.

Because when constructors RatNum(int n) and RatNum(int n, int d) create the object, they already check if the object is qualified. So calling "this" in the member function cannot be null.

3.

Static Method in Java belongs to the class but not its instances. valueOf(String) function does not have a parameter of class. A possible solution is to create a new new for the string to do the similar operations as valueOf(String).

4.

Using new in these functions makes them creators instead of mutators that change what is inside. None of these specifications ask us to change the value itself. So we should use new to make value untouched.

5.

Because the constructor RatNum is immutable. The value of the object cannot be changed after it has been created. So, checking by checkRep() at the end meet also means it is qualified at any point inside the constructor.