How to invoke a method on a class dynamically using reflection api?

```
class Random {
  public int nextInt(int min) {
    return 93;
  }

  public int nextInt(int min, int max) {
    return 33;
  }
}
```

```
class Mapper {
  public static void mapRecord(String fqnClass) {
    Class<?> clazz = Class.forName(fqnClass);
    Object obj = clazz.newInstance();

    Method method = clazz.getDeclaredMethod("nextInt", new Class<?>[]
{Integer.class});
    Object retValue = method.invoke(obj, new Object[]{30});
    sop(retValue);
  }
}
```

```
Random random = new Random();
int r = random.nextInt(10);
returnValue = object.method(params);
```

Class object holds the definition information about the class like
1. Fields
2. Constructors
3. Methods

we can understand:
always a method will invoked on a object with parameters.

In the above we cannot invoke the methods on the Random class using object.methodName(params), because the referenceType is generic. So instead of calling object.method(params), we can call method.invoke(obj, params);

we can get reference of Method of that class using Class object. So we can access the reference of a Method uniquely on a Class using methodName & argumentTypes