1st International Conference on Applied Soft Computing Techniques
22 & 23.04.2017
In association with
International Journal of Scientific Research in Science and Technology

Diminution of Testcases in Object Oriented Software using JUnit Tool

B.Geetha[*1], V.Bhaskar[2], Dr.D.Jeya Mala[3]
*[1]*Department of Master of Computer Applications, KLN College Of engineering, Madurai, Tamilnadu,India*
bgeet76@gmail.com[1]
[2]*Department of Computer Science Engineering, KLN College Of Engineering, Madurai, Tamilnadu,India*
bhaskarv@gmail.com[2]
[3]*Department of Master Of computer Applications, Thiagarajar College Of Engineering, Madurai,Tamilnadu,India*

ABSTRACT

To ensure the code is working the way the developer intends to test the software.Testing is very costliest phase in software development.The object oriented programming paradigm generating test data is very challenging one.To reach high code coverage search based software testing is used.This paper introduces a novel way to deal with decreases the quantity of experiments utilizing JUnit tool.It likewise tackles the requesting of articles utilizing reflexive strategy in question situated system.This approach comprises of three phases.The first stage distinguishes the kind of protest utilizing reflexive method.In this second stage we build up a need keeping in mind the end goal to keep the abnormality of testing order.In the third phase,we diminish the no of test cases in object oriented software using JUnit tool.

Keywords: Object Oriented, Junit Tool, Software Testing

I. INTRODUCTION

Test-information era is a basic and imperative phase of programming development.Manually producing test-information is exertion devouring task.Revealing concealed mistakes in code execution is exceptionally tedious around half time of the aggregate programming life-cycle[3].Object arranged programming testing is extremely testing task,because of its elements like conceptual class, Inheritance, polymorphism and perceivability.

Many methodologies are accessible to create programmed test cases which incorporate distinctive instruments and strategies which abridged as takes after, Code based test information era techniques.There are fundamentally 3 techniques used to produce test information, Random Test Generation: Random test era produces great outcome in test information era.It can haphazardly choose contribution until data sources are discovered [9]. Irregular test era is simple and also quick technique to produce test information. It can produce enormous number of experiments automatically. Symbolic Execution based procedures:symbolic execution is a method for test-information era proposed by James King in 1976 [10]. Typical execution based system contain dynamic typical execution and concolic testing.Typical test information era methods [11,12] dispense typical qualities to the factors and create mathematical expressions for the numerous imperatives in the program code. An imperative solver is utilized to decide answer for these expressions that cover a test prerequisite.Typical execution can be utilized for various purposes, as discovery of bug, program check, troubleshooting of code, support, and limitation of blame [13].

In inquiry based techniques,search calculation name itself speaks to some sort of scope.It has extensive applications in test information era in light of the fact that the creating programming tests is an undesirable issue [14,15]. It use as enhancement method for test information era and to take care of programming building issues. Advantage of utilizing SBST is that its result demonstrates the productivity of approach just disadvantage is it requires extensive inquiry space.

A. Major issues involved in object oriented Testing

The issues of testing article arranged frameworks are legacy, embodiment and polymorphism. Alternate issues ofquestion situated frameworks are decentralized code,encapsulation,testcases tinguishing proof and raising special cases.Question arranged frameworks are worked out of at least two interrelated items A Determining the accuracy of O-O frameworks requires testing the techniques that change or convey the state ofan protest Å Testing strategies in a protest situated framework is like trying subprograms in process-situated frameworks .The conditions happening in traditional systems are information conditions between factors calling conditions between modules ,practical conditions between a module and the factors it processes, denitional conditions between a variable and its sort. OO Systems have extra conditions: class to class conditions class to strategy conditions; class to message dependencies,class to variable conditions strategy to variable conditions. technique to message conditions and strategy to strategy dependencies.[8]

II. LITERATURE SURVEY

Many Researches have proposed a wide range of solutions to reduce the number of test cases in object oriented system.

a.Soft Computing approaches

Bipin Pandey, Rituraj[1] Jain connected delicate Computing in various sorts of programming testing.By utilizing the systems for programming testing,the result has been upgraded and the general execution of testing was made strides

b.Testing least dependent class

Dr. Reena Dadhich and Sourabh Sehgal [2] favored the test arrange from minimum ward class to most ward class.The outcome demonstrated that most ward class to minimum ward class spares the time and endeavors.

c.Random test case generation

Ingle S.E.1, Mahamune M. R[4] clarified the various types of procedures for producing experiments to satisfy experiment examination standard. The way arranged test perceives way for which experiment must be made, still the way may be found infeasible, and the test information generator improves the info that is go through the way.

d.UML sequence diagram

A.V.K. Shanthi1 and G. Mohan Kumar[7] proposed the experiment era by methods for UML Sequence outline utilizing Genetic Algorithm where test cases are upgraded. This approach is essential to recognize area of a blame in the execution, in this way decreasing testing exertion. Besides this strategy for experiment era proposes the engineers to enhance the outline quality, lessen programming improvement time and discover blames in the execution early phase of programming development[5].

III. PROPOSED SYSTEM

In our proposed framework, we bring the protest under two noteworthy classification.1)Dependent question 2)independent question. We group the question utilizing reflexive technique.We give most elevated need to ward protest while doing programming testing.Because,dependent question covers more number of information members.Then,second need moves to free questions. We test both items utilizing JUnit tool. After ordering the question arranged testing, we proposed another strategy for decreasing the quantity of experiments without trading off the nature of the product. The foundation of our proposed strategy is reflection and JUnit tool.In this area we talk about the quality of reflection,issues in question arranged testing which conveys an ideal answer for protest situated testing. we talk about the parts of our approach one by one..

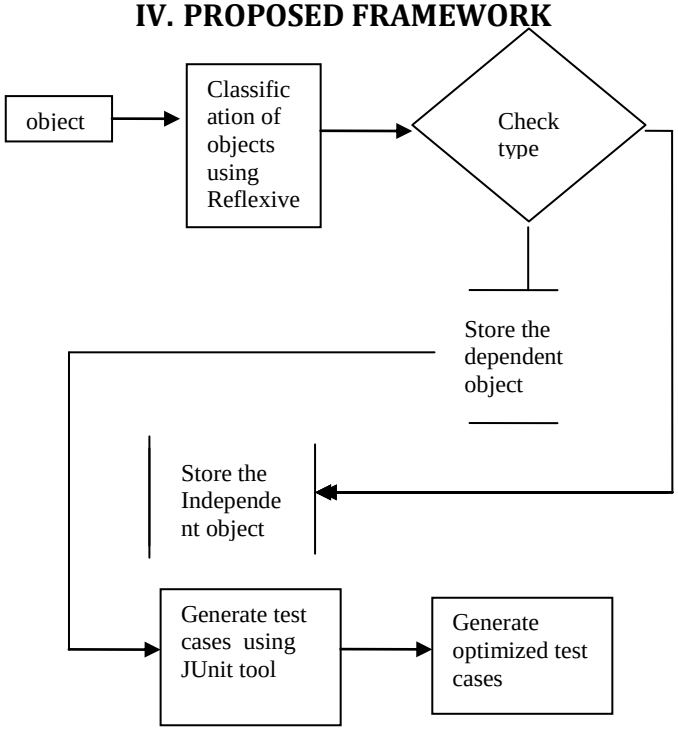
A.Reflection

Reflection is one of advanced features of object oriented languages. This technique has the ability to find the type of objects at run time. It supports various methods which helpful to find out whether an object is a dependent object or independent object. The proposed system uses the reflection technique to classify whether the object is an a independent object or dependent object. This can be used by testers and developers who have a strong grasp of the fundamentals of the object oriented language. since it is a sensitive technology, it requires a run time permission of the security manager.

B.Testing the object using JUnit tool

One of the cost effective software testing is JUnit tool. Many Researchers have utilized the effectiveness of JUnit tool to test the object based software testing. In our proposed approach we use rule based fuzzy logic[8]. To test all kinds of data types of data members of an object, we choose only one data member from each cluster.

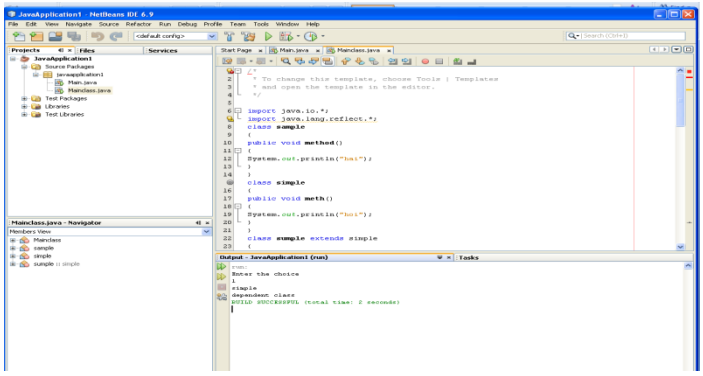
IV. PROPOSED FRAMEWORK

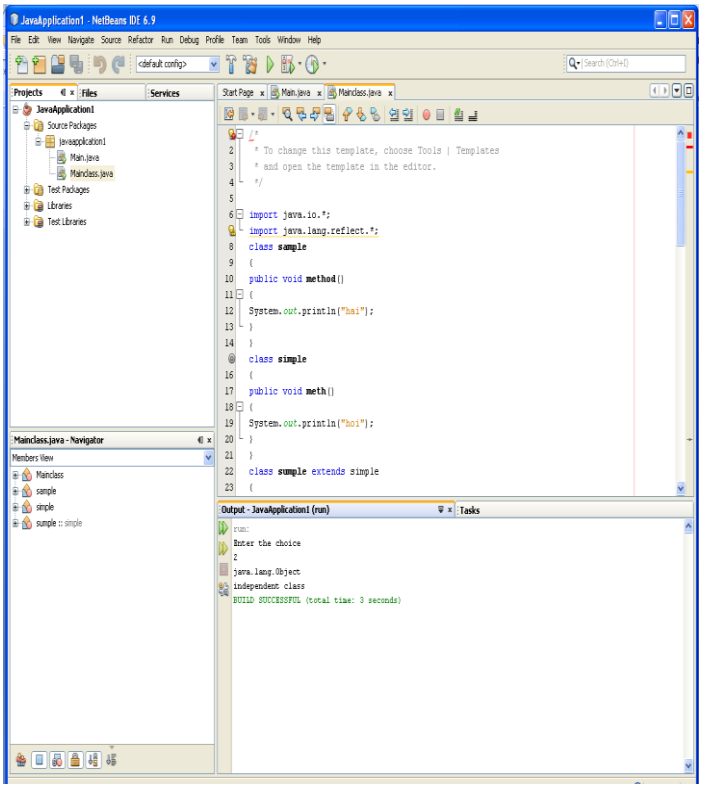


V. EXPERIMENTAL SETUP

Our experimental setup consists of an application which is implemented in java bean.These applications are executed in the Netbeans IDE.We utilized the power of java for classification of objects.The following Table1 shows the implementation code for classification of objects

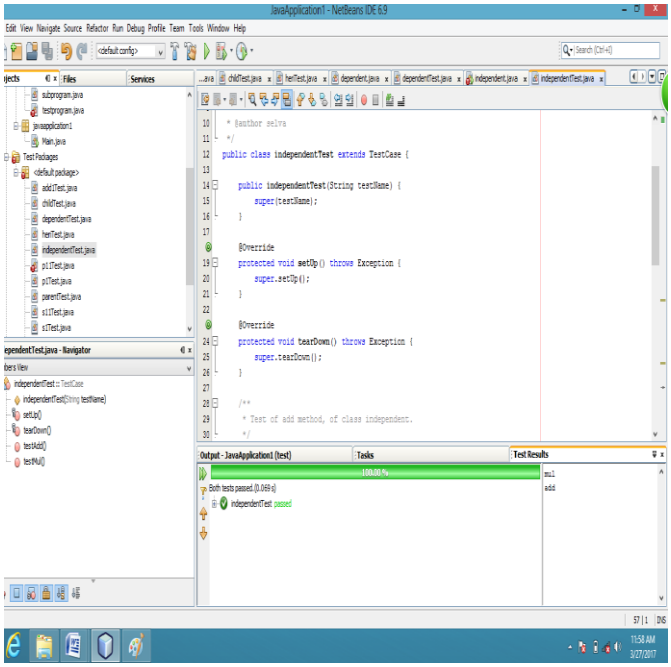
```
import java.io.*;
class sample{
public void method()
{
System.out.println("hai");
}}
class simple
{
public void meth(){
System.out.println("hoi");
}}
class sumple extends simple
{
public void met(){
System.out.println("welcome");
}}
class MainClass{
public static void main(String ar[])throws IOException
{
System.out.println("Enter the choice");
DataInputStream dis=new DataInputStream(System.in);
int x=Integer.parseInt(dis.readLine());
String name=" ";
Class c;
if(x==1)
{sumple s=new sumple();
```

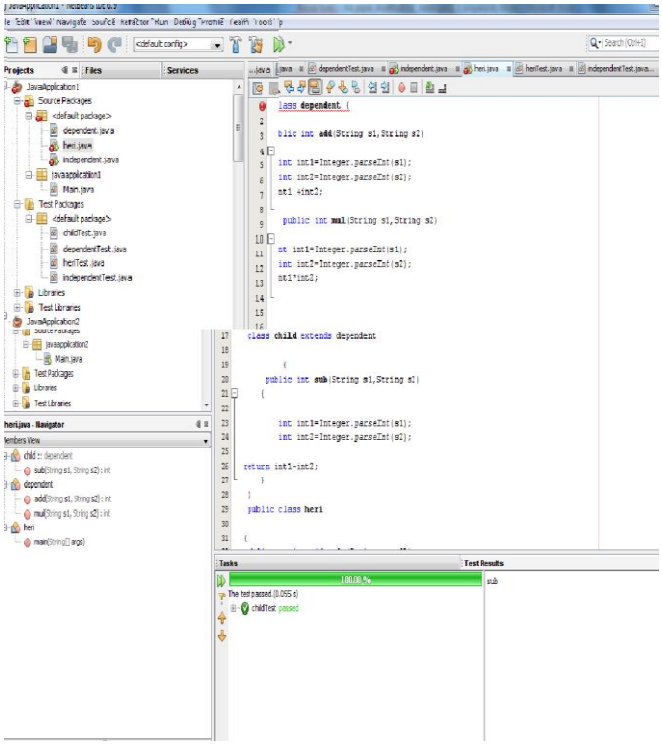




Then the dependent and independent objects are tested using JUnit tool distinctly.The following table shows tested the independent object using JUnit tool

```
public class independent {
    public int add(String s1,String s2)
    {
        int int1=Integer.parseInt(s1);
        int int2=Integer.parseInt(s2);
return int1 +int2;
    }
   public int mul(String s1,String s2)
    {
      int int1=Integer.parseInt(s1);
```





The following table shows the dependent objects are tested using JUnit tool.In this each child class create a distinct test program due to challenging issues of object oriented software.In each class tested distinctly giving the input values.

```
public class dependent {
     public int add(String s1,String s2)
    {
        int int1=Integer.parseInt(s1);
        int int2=Integer.parseInt(s2);
return int1 +int2;
    } public int mul(String s1,String s2)
    {
       int int1=Integer.parseInt(s1);
        int int2=Integer.parseInt(s2);
return int1*int2; } }
class child extends dependent
         {
     public int sub(String s1,String s2)
    {
        int int1=Integer.parseInt(s1);
        int int2=Integer.parseInt(s2);
return int1-int2; } }
public class heri
```

VI. CONCLUSION

In this paper, we exhibited an approach,that is order the protest utilizing reflexive procedure and store it.Then these dependent objects and independent objects are tested using JUnit tool.The dependent object and independent objects are tested using items are assembled utilizing bunching strategy. Bunching procedure gives an approach to choose a test information thing from each gathering and diminishes the quantity of testcases. Using sequence of method calls testing the dependent objects save the time and reduce cost framework resources.The extent of our proposed framework is completely mechanized apparatus to be developed.The device will encourages all levels of question arranged programming testing and spare time and decrease the cost of the product.

VII. REFERENCES

[1]. Bipin Pandey, Rituraj Jain, Soft Computing Based Approaches for Software Testing: A Survey, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-4, Issue-2, May 2014.

[2]. Dr. Reena Dadhich1 and Sourabh Sehgal2 Analyzing The Efficient Test Order For Integration Testing,Advanced Computing: An

International Journal (ACIJ), Vol.3, No.4, July 2012(dependent independent object)

[3]. http://mit.bme.hu/~micskeiz/pages/code_based_test_generation.html (lastly accessed on April 05, 2016).

[4]. Ingle S. E.1, Mahamune M. R., An UML Based software Automatic Test Case Generation: Survey International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 02 Issue: 02 | May-2015(Irjet-v2i276.pdf)

[5]. GurkarandeshKaur, Parminder Singh,Test Case Generation Using UML Diagram, International Journal of Emerging Technologies in Engineering Research (IJETER) Volume 1, Issue 2, July (2015)(Manuscripts-volume-1-issue)

[6]. A.V.K. Shanthi and G. Mohan Kumar, Automated Test Cases Generation from UML Sequence Diagram, 2012 International Conference on Software and Computer Applications (ICSCA 2012)IPCSITvol. 41 (2012) © (2012) IACSIT Press, Singapore(testcasesequences2.pdf)

[7]. Object-Oriented Software Testing | Some Research and Development David C. Kung and Pei Hsia

[8]. Computer Science and Engineering Dept. The Univ. of Texas at Arlington Yasufumi Toyoshima, Cris Chen, Jerry Gao Fujitsu Network Communication Systems

[9]. Jon Edvardsson ,"A Survey on Automatic Test data generation", 2nd Conference on Computer Science and Engineering , Vol. 2, No. 1, (October 1999) , pp. 21-28.

[10]. J. C. King, "Symbolic execution and program testing", Communication ACM, vol. 19, no. 7, (July 1976), pp. 385–394.

[11]. Howden, William E.,"Symbolic testing and the DISSECT symbolic evaluation system", IEEE Transactions on Software Engineering, vol. 3, no. 4, (July 1977), pp. 266-278.

[12]. John Clarke, Mark Harman, Bryan Jones, "The Application of Metaheuristic Search techniques to Problems in Software Engineering", IEEE Computer Society Press Vol. 42, No. 1, (2000), pp. 247-254.

[13]. L. A. Clarke and D. J. Richardson," Applications of symbolic evaluation", Journal of Systems and Software, (Feb 1985), pp. 15–35.

[14]. Tao Feng, Kasturi Bidarkar, " Survey of Software Testing Methodology", vol. 25, no-3 , (2008) , pp. 216-226.

[15]. Voas,Morell and Miller, "Predicting where faults can hide from testing", IEEE vol. 8 , pp. 41-48.