UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Validação e Teste de Software - 2020
Profa Simone Senger Souza

Exercício - Teste Mutação

1) Considerando o método *findVal*, responda as questões seguintes:

Função Original	Função Mutante
//Effects: If numbers null throw NullPointerException *//else return LAST occurrence of val in numbers[]* *//if val not in numbers[] return -1* /*1*/ public static int findVal(int numbers[], int val) /*2*/ { /*3*/ int find = -1; /*4*/ for (int i=0; i < numbers.length; i++) /*5*/ if(numbers[i] == val) /*6*/ find = i; /*7*/ return(find); /*8*/ }	*//Effects: If numbers null throw NullPointerException* *//else return LAST occurrence of val in numbers[]* *//if val not in numbers[] return -1* /*1*/ public static int findVal(int numbers[], int val) /*2*/ { /*3*/ int find = -1; /*4*/ for (int i=(0+1); i < numbers.length; i++) *//Mutant* /*5*/ if(numbers[i] == val) /*6*/ find = i; /*7*/ return(find); /*8*/ }

a) Crie um caso de teste que "mata" esse mutante.

b) Crie um caso de teste que não mata esse mutante.

c) Considerando a versão original do programa (sem a mutação), crie um mutante equivalente e explique por que ele é equivalente. Indicar linha e mutação feita.

d) Considerando a versão original do programa, escolha um operador de mutação do slide 11, os quais são os operadores essenciais, e crie um mutante. Mostre a mutação (linha e mutante criado) e a seguir encontre um caso de teste que mata esse mutante ou justifique caso ele seja equivalente.

e) Considerando os operadores essenciais (slide 11), aproximadamente, quantos mutantes você acredita que seriam gerados para essa função? Apresente a descrição de como fez a análise por operador de mutação.