Министерство образования и науки Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

«СЕВЕРО-ВОСТОЧНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ М.К. АММОСОВА»

Институт математики и информатики

Лабораторная работа №4

Наследование. Обработка исключительных ситуаций

Выполнил студент 1-го курса гр. ПИ-19-1

Макаров Максим Егорович

Якутск 2020

Задание 4.1

Пример 4.1

Сначала выводит “0”;

Потом вызывает ошибку RuntimeException(“Непроверяемая ошибка”);

Эту ошибку ловит catch и выводит “1” и “java.lang.RuntimeException: Непроверяемая ошибка”;

И в конце выводит “2”;

Пример 4.2

Выводится “0”;

Затем вызывается ошибка RuntimeException(“Непроверяемая ошибка”);

Затем пытается вывести “1”, затем ломается программа, выводя ошибку java: unreachable statement;

Пример 4.3

Выводится “0”;

Затем вызывается ошибка RuntimeException(“ошибка”);

После этого она перехватывается перехватчиком предка catch (RuntimeException e), выводя “2”, игнорируя NullPointerException и пропуская Error;

И в конце выводится “4”

Пример 4.4

Выводится “0”;

Затем вызывается ошибка RuntimeException(“ошибка”);

После этого она перехватывается перехватчиком предка catch (Exception e), выводя “2”, игнорируя NullPointerException и пропуская Error;

И в конце концов выводится “4”;

Пример 4.5

Выводится “0”;

Затем вызывается ошибка RuntimeException(“ошибка”);

Ошибка не перехватывается (потому что NullPointerException не подходит) и вызывает ошибку RuntimeException: ошибка;

Пример 4.6

Выводится “0”;

Затем вызывается ошибка RuntimeException(“ошибка”);

Она не перехватывается, так как последовательность перехвата не соответствует иерархии исключений;

Программа ломается;

Пример 4.7

Выводится “0”;

Вызывается ошибка NullPointerException(“ошибка”);

Ошибка перехватывается, выводит “1” и вызывается ошибка ArithmeticException;

Программа выдает ошибку, так как нельзя вызывать ошибки в catch;

Пример 4.8

Выводится “0”;

Вызывается ошибка RuntimeException;

В любом случае выводится “1” так как он находится в finally;

Программа выдает ошибку, так как RuntimeException не перехватывается;

Пример 4.9

Выводится “0”;

Возвращается 55, то есть выходит из метода выводя 55, но так как в программе имеется finally с sout(“1”) сначала выведется 1, а затем 55;

Пример 4.10

Выводится “0”;

Возвращается 15;

Так как тут имеется finally, то выводится “1” и возвращается 20;

В итоге выводится:

0

1

20

Потому что возврат 15, замещается возвратом 20 из finally;

Пример 4.11

Выводится “0”;

Вызывается ошибка NullPointerException(“ошибка”);

Ошибка перехватывается и выводит “1”;

Затем выводится “2” из finally;

После всего этого выводится “3”;

Пример 4.12

Из main в m передается 2 значения – это str = null и сhislo = 0.000001;

Если str == null, то вызывается ошибка IllegalArgumentException(“Строка введена неверно”);

Выдается ошибка, потому что IllegalArgumentException не перехватили, и программа останавливается;

Пример 4.13

Задается переменная l = длине массива args;

Выводится “Размер массива= ” длинна массива args = 0;

h/l – получается ошибка ArithmeticException(потому что делится на ноль);

Присваивается String значение “10” элементу массива args с индексом l +1 (0 + 1);

Перехватывается ArithmeticException и выводится “Деление на ноль”;

Выходит из try;

Пример 4.14

Задается переменная l = длине массива args;

Выводится “Размер массива= ” длинна массива args = 0;

Вызывается метод m(x = l);

Там задается h = 10/x;

Ошибка перехватывается и выводится “Ошибка: Деление на ноль”;

Задание 4.2

Вариант 2

В программе, где требуется из матрицы вывести столбец с номером, заданным с клавиатуры, могут возникать ошибки в следующих случаях: – ввод строки вместо числа; – нет столбца с таким номером

1.1

Сначала вводим с клавиатуры инициализированные переменные x и y;

Создаем двухмерный массив с [x][y];

try{создаем цикл который будет заполнять массив случайными числами и выводить}

catch(ошибка InputMismatchException e){выводится строка “Вы ввели не целое число ” и сама ошибка};

catch(ошибка ArrayIndexOutOfBoundsException e){выводится “Такого столбца нет ” и сама ошибка };

import java.util.InputMismatchException;  
import java.util.Scanner;  
  
public class ex2\_var2 {  
 public static Scanner *sc* = new Scanner(System.*in*);  
  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("Введите кол-во строк: ");  
 int x = *sc*.nextInt();  
 System.*out*.println("Введите кол-во столбцов: ");  
 int y = *sc*.nextInt();  
 int[][] arr = new int[x][y];  
 for (int i = 0; i < x; i++) {  
 for (int j = 0; j < y; j++) {  
 arr[i][j] = (int) (Math.*random*() \* 10);  
 System.*out*.print("[" + arr[i][j] + "] ");  
 }  
 System.*out*.println();  
 }  
 System.*out*.println("Какой столбец вывести?");  
 int k = *sc*.nextInt();  
 for (int i = 0; i < arr.length; i++) {  
 System.*out*.println("[" + arr[i][k - 1] + "]");  
 }  
  
 } catch (InputMismatchException e) {  
 System.*out*.println("Вы ввели не целое число: " + e);  
 } catch (ArrayIndexOutOfBoundsException e) {  
 System.*out*.println("Такого столбца нет: " + e);  
 }  
 }  
}

1.2

Метод main

try{

вызывается метод fillArr()

Метод fillArr() создает и заполняет массив arr случайными числами и отправляет этот массив в метод str()

Метод str() получает массив arr и выводит нужный нам столбец

}

catch(ошибка java.util.InputMismatchException e){выводится строка “Вы ввели не целое число”};

catch(ошибка java.lang.ArrayIndexOutOfBoundsException e){выводится “Такого столбца нет”};

import java.util.InputMismatchException;  
import java.util.Scanner;  
  
public class ex2\_2\_var2 {  
 public static Scanner *sc* = new Scanner(System.*in*);  
 public static void main(String[] args) {  
 try{  
 *fillArr*();  
 }catch (InputMismatchException e){  
 System.*out*.println("Вы ввели не целое число: " + e);  
 }catch (ArrayIndexOutOfBoundsException e){  
 System.*out*.println("Такого столбца нет: " + e);  
 }  
 }  
 public static void fillArr(){  
 System.*out*.println("Введите кол-во строк: ");  
 int x = *sc*.nextInt();  
 System.*out*.println("Введите кол-во столбцов: ");  
 int y = *sc*.nextInt();  
 int[][] arr = new int[x][y];  
 for(int i = 0; i < x; i++){  
 for(int j = 0; j < y; j++){  
 arr[i][j] = (int)(Math.*random*()\*10);  
 System.*out*.print("["+arr[i][j] + "] ");  
 }  
 System.*out*.println();  
 }  
 *str*(arr);  
 }  
 public static void str(int[][] arr){  
 System.*out*.println("Какой столбец вывести?");  
 int k = *sc*.nextInt();  
 for(int i = 0; i < arr.length; i++){  
 System.*out*.println("["+arr[i][k-1]+"]");  
 }  
 }  
}

Контрольные вопросы

1. Наследование позволяют дополнить один класс другим, который унаследовал его.
2. …RuntimeException(unchecked) > Exception(checked) > Throwable(checked) > java.lang.Object

…Error(unchecked) > Throwable(checked) > java.lang.Object

1. try{часть кода, которая может вызвать ошибку}catch(ошибка e){при перехвате ошибки, выполняется эта часть кода}
2. По иерархии снизу вверх(вопрос 2)
3. throw