###### Table C37 Test Report Template – Las pruebas las acomodé por números de programa a probar por lo que los errores corregidos no llevan un orden

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Daniel Alejandro Morales Castillo | Date | 6/06/2022 |
| Instructor | Dra. María de León Sigg | Program # | 5 |

|  |  |
| --- | --- |
| **Test Name/Number:** | **Prueba 1 – 15\_Code\_Programa1\_E6\_E22(nombrado Archivos Operaciones por comodidad)** |
| **Test Objective:** | **Hacer que el programa calcule adecuadamente CBO** |
| **Test Description:**  **Programa para calcular el CBO en el programa 1 de PSP** | |
| **Test Conditions:**  **El archivo utiliza el estandar de codificación y los criterios de aceptacion**  **Criterio de aceptación: Introducir los //comentarios que indicant el final de cada método y clase para poder realizar el cálculo correcto de CBO**  **Criterio de aceptación: introducir una ruta valida con el archivo modificado anteriormente mencionado con sus comentarios.** | |
| **Expected Results:**  **1 sola clase con valores de 0.0 CBO** | |
| **Actual Results:**  **Salida no correcta**    **La salida no es la esperada ya que no lee el nombre de la clase**  **Salida de CBO: 0.0 sin nombre de clase** | |
| **Test status (Passed/Failed):**  **FAILED** | |

|  |  |
| --- | --- |
| **Test Name/Number:** | **Prueba 2 – 15\_Code\_Programa1\_E6\_E22(nombrado Archivos Operaciones por comodidad)** |
| **Test Objective:** | **Hacer que el programa calcule adecuadamente CBO** |
| **Test Description:**  **Programa para calcular el CBO en el programa 1 de PSP** | |
| **Test Conditions:**  **El archivo utiliza el estandar de codificación y los criterios de aceptacion**  **Criterio de aceptación: Introducir los //comentarios que indicant el final de cada método y clase para poder realizar el cálculo correcto de CBO**  **Criterio de aceptación: introducir una ruta valida con el archivo modificado anteriormente mencionado con sus comentarios.** | |
| **Expected Results:**  **1 sola clase con valores de 0.0 CBO** | |
| **Actual Results:**  **Clase ArchivosOperaciones con 0.0 de CBO, ahora ya con el nombre de la clase** | |
| **Test status (Passed/Failed):**  **PASSED**  **ERROR CORREGIO #8** | |

|  |  |
| --- | --- |
| **Test Name/Number:** | **Prueba 3 -15\_Code\_Programa2** |
| **Test Objective:** | **Hacer que el programa calcule adecuadamente LCOM** |
| **Test Description:**  **El archivo utiliza el estandar de codificación y los criterios de aceptacion**  **Criterio de aceptación: Introducir los //comentarios que indicant el final de cada método y clase para poder realizar el cálculo correcto de CBO**  **Criterio de aceptación: introducir una ruta valida con el archivo modificado anteriormente mencionado con sus comentarios.** | |
| **Test Conditions:**  **El archivo utiliza el estandar de codificación y los criterios de aceptacion** | |
| **Expected Results:**  **3 clases con porcentajes CBO de 0.0 0.0 y 2.0 con un CBO total de 2.0** | |
| **Actual Results:**  **Solo leyó dos clases y el cálculo no es el esperado** | |
| **Test status (Passed/Failed):**  **FAILED** | |

|  |  |
| --- | --- |
| **Test Name/Number:** | **Prueba 4 -15\_Code\_Programa2** |
| **Test Objective:** | **Hacer que el programa calcule adecuadamente LCOM** |
| **Test Description:**  **Archivo para identificar el porcentaje LCOM en el programa 2 de PSP.** | |
| **Test Conditions:**  **El archivo utiliza el estandar de codificación y los criterios de aceptacion**  **Criterio de aceptación: Introducir los //comentarios que indicant el final de cada método y clase para poder realizar el cálculo correcto de CBO**  **Criterio de aceptación: introducir una ruta valida con el archivo modificado anteriormente mencionado con sus comentarios.** | |
| **Expected Results:**  **3 clases con porcentajes CBO de 0.0 0.0 y 2.0 con un CBO total de 2.0** | |
| **Actual Results:**  **3 clases con porcentajes CBO de 0.0 0.0 y 2.0 con un CBO total de 2** | |
| **Test status (Passed/Failed):**  **PASSED**  **ERROR CORREGIDO #6 #7 Y #9** | |

|  |  |
| --- | --- |
| **Test Name/Number:** | **Prueba 5 – 15\_Code\_Programa3\_E6\_E22** |
| **Test Objective:** | **Hacer que el programa calcule adecuadamente CBO** |
| **Test Description:**  **El archivo utiliza el estandar de codificación y los criterios de aceptacion**  **Criterio de aceptación: Introducir los //comentarios que indicant el final de cada método y clase para poder realizar el cálculo correcto de CBO**  **Criterio de aceptación: introducir una ruta valida con el archivo modificado anteriormente mencionado con sus comentarios.** | |
| **Test Conditions:**  **El archivo utiliza el estandar de codificación y los criterios de aceptacion** | |
| **Expected Results:**  **3 clases con 0.0, 0.0 Y 1**  **Ttoal de 1** | |
| **Actual Results:**  **Salida CORRECTA** | |
| **Test status (Passed/Failed):**  **PASSED**  **DEFECTO PASADO ARREGALDO PARA OTROS PROGRAMAS #6 #7 Y #9** | |

|  |  |
| --- | --- |
| **Test Name/Number:** | **Prueba 6 – 15\_Code\_Programa4\_E3\_E22** |
| **Test Objective:** | **Hacer que el programa calcule adecuadamente CBO** |
| **Test Description:**  **El archivo utiliza el estandar de codificación y los criterios de aceptacion**  **Criterio de aceptación: Introducir los //comentarios que indicant el final de cada método y clase para poder realizar el cálculo correcto de CBO**  **Criterio de aceptación: introducir una ruta valida con el archivo modificado anteriormente mencionado con sus comentarios.** | |
| **Test Conditions:**  **El archivo utiliza el estandar de codificación y los criterios de aceptacion** | |
| **Expected Results:**  **4 clases con valores de 3.0 , 1.0, 0.0 y 0.0 y Total de 4** | |
| **Actual Results:**  **Salida correcta**    **La salida es la correcta haciendo el calculo ahora para el programa 4 y solucionado el problema de antes** | |
| **Test status (Passed/Failed):**  **FAILED** | |

|  |  |
| --- | --- |
| **Test Name/Number:** | **Prueba 7 –CBO.java** |
| **Test Objective:** | **Hacer que el programa calcule adecuadamente CBO** |
| **Test Description:**  **El archivo utiliza el estandar de codificación y los criterios de aceptacion**  **Criterio de aceptación: Introducir los //comentarios que indicant el final de cada método y clase para poder realizar el cálculo correcto de CBO**  **Criterio de aceptación: introducir una ruta valida con el archivo modificado anteriormente mencionado con sus comentarios.** | |
| **Test Conditions:**  **El archivo utiliza el estandar de codificación y los criterios de aceptacion** | |
| **Expected Results:**  **4 clases con valores de 2, 0 y 0 con total de 2** | |
| **Actual Results:**  **Salida correcta**    **La salida es la correcta haciendo el calculo ahora para el programa 5 y solucionado el problema de antes** | |
| **Test status (Passed/Failed):**  **FAILED** | |

Estándar de códificación

**Table Java Coding Standard – Daniel Castillo**

|  |  |
| --- | --- |
| Purpose: | To guide the development of Java programs |
| Program Headers | Begin all programs with a descriptive header. |
| Header Format | /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\* Program Assignment: the program number \*/  /\* Name: your name \*/  /\* Date: the date program development started \*/  /\* Description: a short description of the program \*/  /\* function \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ |
| Identifiers | Use descriptive names for all variables, function names, constants, and other identifiers. Avoid abbreviations or single letter variables. |
| Identifier Example | int number\_of\_students; /\* This is GOOD \*/  float x4, j, ftave; /\* These are BAD \*/ |

|  |  |
| --- | --- |
| Comments | • Document the code so that the reader can understand its operation.  • Comments should explain both the purpose and behavior of the code.  • Comment variable declarations to indicate their purpose. |
| Good Comment | if (record\_count > limit) /\* have all the records been processed? \*/  if(record\_count > limit) /\* check if record\_count is greater than limit \*/  If(x > y) // This comment could be good too, using only // |
| Bad Comment | if(record\_count > limit) /\* Create a if statement \*/ |
| Major Sections | Precede major program sections by a block comment that describes the processing that is done in the next section. |
| Example | /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\* This program section will examine the contents of the array “grades” \*/  /\* and will calculate the average grade for the class. \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ |
| Blank Spaces | • Write programs with sufficient spacing so that they do not appear crowded.  • Separate every program construct with at least one space. |
| Indenting | • Each opening brace must contain its closing brace, it doesn't matter if the indentation is on a different line as long as it is closed. |
| Indenting Example | while (miss\_distance > threshold){  success\_code = move\_robot (target\_location);  if (success\_code == MOVE\_FAILED){  printf(“The robot move has failed.\n”);  }  } |
| Capitalization | • Capitalized all defines.  • Lowercase all other identifiers and reserved words.  • Messages being output to the user can be mixed-case so as to make a clean user presentation. |
| Capitalization Example | #define DEFAULT-NUMBER-OF-STUDENTS 15  int class-size = DEFAULT-NUMBER-OF-STUDENTS; |
| Package names | * Package names must be lowercase with no underscores or other special characters. * Package names start with the overridden authority part of the developer's company web address. This part can be followed by a package substructure that depends on the project/program structure. |
| Package example | com.yourcompany.widget.button;  com.yourcompany.core.api;  java.util.\*; |
| Class, interface, and enumeration names | * Class and enum names must be names. * Interface names should normally be nouns or adjectives ending with ... capable. * Use upper and lower case with the first letter of each word capitalized (ie CamelCase ). * Use full words and avoid using abbreviations unless the abbreviation is more widely used than the long form. * Format an abbreviation as a word if it is part of a longer class name. |
| Class, Interface and enum names Examples | ArrayList  BigInteger  ArrayIndexOutOfBoundsException  Iterable |
| Method names | * Use uppercase and lowercase with the first letter in lowercase. |
| Method Examples | toString  hashCode  createPSP  deleteNode |
| Variables | * Variable names must be case sensitive with the first letter lowercase * The definition of variables with their data type can always go on the same line |
| Variables Examples | element  aux  varAux  int run, counter, mathAux; |
|  |  |
|  |  |
|  |  |