

**Course Experiment Report**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Course:** | Java Language | | | | | | |
|  |  | | | | | | |
| **Semester:** | 1-18th | **week** | 12th | **year** | | 1st | **term** |
|  |  |  |  |  | |  |  |
| **Major:** | Software Engineering | | | | | **Class:** | 2019 |
|  |  | | | | |  |  |
| **Student name:** | 冯春霖 | | **Student No.:** | | 222019321062074 | | |
|  |  | |  | |  | | |
| **Teacher:** | Wang Xiaomeng | | | | | | |

College of Computer and Information Science

|  |  |  |  |
| --- | --- | --- | --- |
| Project | Exp5 Exception Handling | | |
| Time | 2020.11.23 | Type | □Verification □Design □Synthetical |
| 1. Answer the questions  (1) Under what conditions will the finally block be executed?  A: If the program executes to the try block, the statement in finally is executed regardless of whether the program is terminated by an exception or some other means.  (2) What is the difference between the keyword throw and throws?  A: throw is an exception throwing action used in method implementations to throw a single exception. throws declares that the method is in a possible exception throwing state, with the possibility of throwing multiple exceptions.  (3) Can the main method declare an exception?  A: Yes, but not recommended. The throw exception is generally handled by a higher-level method after the exception is thrown. Exceptions declared in the main method may be difficult to handle since the main method is already at the top of the program.  (4) Other experience.  A: Through this experiment, I have a basic grasp of how to handle exceptions.  2. All Codes  Code of sunTwoIntegers.java:  **package** week12;  **public** **class** Triangle {  **double** side1 = 1.0;  **double** side2 = 1.0;  **double** side3 = 1.0;    **public** **Triangle**() {};    **public** **Triangle**(**double** \_s1, **double** \_s2, **double** \_s3) **throws** Exception  {  **if**(\_s1 <= 0)  **throw** **new** IllegalTriangleException("\_s1 <= 0");  **else** **if**(\_s2 <= 0)  **throw** **new** IllegalTriangleException("\_s2 <= 0");  **else** **if**(\_s3 <= 0)  **throw** **new** IllegalTriangleException("\_s3 <= 0");  **else** **if**(\_s1 + \_s2 <= \_s3)  **throw** **new** IllegalTriangleException("\_s1 + \_s2 <= \_s3");  **else** **if**(\_s1 + \_s3 <= \_s2)  **throw** **new** IllegalTriangleException("\_s1 + \_s3 <= \_s2");  **else** **if**(\_s2 + \_s3 <= \_s1)  **throw** **new** IllegalTriangleException("\_s2 + \_s3 <= \_s1");  side1 = \_s1;  side2 = \_s2;  side3 = \_s3;  }    **public** **double** **getSide1**()  {  **return** side1;  }  **public** **double** **getSide2**()  {  **return** side2;  }    **public** **double** **getSide3**()  {  **return** side3;  }    **public** **double** **getArea**()  {  **double** **p** = (side1 + side2 + side3) / 2;  **return** Math.*sqrt*(p \* (p - side1) \* (p - side2) \* (p - side3));  }    **public** String **toString**()  {  **return** "Three sides of the triangle are " + side1 + ", " + side2 + ", " + side3;  }    **class** IllegalTriangleException **extends** Exception  {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;    **public** **IllegalTriangleException**(String msg)  {  **super**(msg);  }    }  } | | | |

|  |  |  |
| --- | --- | --- |
| Evaluation | Code Correctness (60%): |  |
| Experience (40%): |  |
| Score： | |