UNIVERSITÄT OSNABRÜCK

Dr. Thomas Wiemann

9. Übungsblatt zur Vorlesung „Einführung in die Programmiersprache C++“

Wintersemester 2014 / 2015

Aufgabe 1: Exception-Handling für Mathe-Klassen (30 Punkte)

Erweitern Sie die Klassen `Vertex`- und `Quaternion`-Klassen um Exceptions. Implementieren Sie dazu eine Klasse `BaseException` und leiten Sie davon die Klassen `OutOfBoundsException` und `DivisionByZeroException` ab, die dann geworfen werden, wenn im []-Operator der Vertex-Klasse auf ein Feld > 2 zugegriffen wird bzw. wenn irgendwo im Code durch Null geteilt werden soll (20 Punkte). Speichern Sie Ihre Exceptions im Unterorder `exceptions` des Programmgerüsts. Ergänzen Sie den Code zum Abfangen eventuell geworfener Exceptions im zur Verfügung gestellten Programm (10 Punkte).

Aufgabe 2: Timestamps und Logging (70 Punkte)

Implementieren Sie die Klasse Timestamp zu folgendem Header:

```
/**
 * @brief A helper class for automated time stamping. Timing is
 *                  started as soon as an object of this class is created.
 *                  To time some parts of a program, just create a new object
 *                  and use the provided output operator to display the elapsed
 *                  time.
 */
class Timestamp
{
public:
        /**
         * @brief   Constructor.
         */
        Timestamp();

        /**
         * @brief   Returns the current system time in milliseconds
         */
        unsigned long  getCurrentTimeInMs() const;

        /**
         * @brief   Returns the milliseconds since object creation
         */
        unsigned long  getElapsedTimeInMs() const;

        /**
         * @brief   Returns the current system time in seconds
         */
        double             getCurrentTimeinS() const;

        /**
         * @brief   Returns the time since instantiation in seconds
         */
        double             getElapsedTimeInS() const;

        /**
         * @brief   Resets the internal timer
         */
        void               resetTimer();

        /**
         * @brief   Returns a string representation of the current
         *          timer value
         */
        string                   getElapsedTime() const;

private:

        /// The system time at object instantiation
        unsigned long         m_startTime;
};
```

Fügen Sie einen Output-Operator hinzu, der einen Timestamp in einen Output-Stream (`ostream`) ausgibt. Die Ausgabe soll dabei folgendermaßen aussehen (20 Punkte):

```
[00:00:12 - 000]
```

Dabei geben die ersten drei Stellen Stunden, Minuten und Sekunden seit Instanzierung der Klasse an. Die Nachkommastellen sind Millisekunden. Nutzen Sie zum Bestimmen der Systemzeit die Funktion `gettimeofday`, die Sie im Header `<sys/time.h>` finden. Diese Funktionalität ist teil der POSIX-Spezifikation, d.h., Sie können diese Funktion unter Linux und MacOS direkt nutzen. Für Windows-Nutzer stellen wir bei Bedarf eine Nachimplementierung des Interfaces zur Verfügung. Die Dokumentation von `gettimeofday` finden Sie z.B. unter `http://linux.die.net/man/2/gettimeofday`. Nutzen Sie Ihre Klasse, um zu bestimmen, wie lange das Laden der Daten und ein Render-Vorgang im Viewer dauern (10 Punkte).

Implementieren Sie neben der Timestamp-Klasse eine Logger-Klasse, mit der Sie wichtige Programmereignisse (z.B. Programmstart, das Laden eines Modells usw.) mit Zeitstempel versehen loggen können (15 Punkte). Die Ausgabe soll wahlweise in einer Datei oder auf der Konsole erfolgen.

Standardmäßig soll die Ausgabe auf der Konsole erfolgen. Wird mittels der Methode

```
void Logger::setOutputFile(string filename)
```

eine Ausgabedatei spezifiziert, soll die Ausgabe in die entsprechende Datei umgeleitet werden. Mittels

```
void Logger::setOutputToStdout()
```

soll die Ausgabe wieder auf die Konsole umgeleitet werden können. Nutzen Sie intern ein `ostream`-Objekt, um die Ausgabe umlenken zu können (10 Punkte).

Implementieren Sie einen Output-Operator, der die folgenden Aufrufe auf einer Instanz `logger` der `Logger`-Klasse ermöglicht:

```
logger << "Programmereignis";
```

Ein solcher Aufruf soll mit Hilfe eines Timestamp-Objektes formatiert mit Zeilenumbruch im Output-Ziel ausgegeben werden.:

```
[00:00:12 - 000] - Programmereignis
```

Demonstrieren Sie die Funktionalität, indem Sie Log-Ereignisse an passenden Stellen im Code ausgeben. Deklarieren Sie eine globale Logger-Variable und demonstrieren Sie Ihrem Tutor die Ausgabe auf Konsole und in eine Datei, indem Sie die Log-Ereignisse in die gegebene Datei umleiten, wenn diese mittels des Programmparameters `--outputFile=filename` definiert wird (15 Punkte).

Beispiel:

```
./viewer arrow.3ds
```

zeigt die Ausgaben auf der Konsole an

```
./viewer arrow.3ds --outputFile="log.txt"
```

leitet die Ausgabe in die Datei "log.txt" um.

Abgabe:

Ihre Dateien sind bis Montag den 15.12.2014 08:00 Uhr mit der Angabe Ihrer Gruppe an die Adresse `cpp@informatik.uni-osnabrueck` zu mailen und die Ausdrucke in den Testaten abzugeben.