



## Testat 0 – Test the Test

Willkommen zur Vorbereitung zum ersten Testat OOP.

Sie ändern im Testat ein vorhandenes Code-Fragment so, dass dann 10 Tests durchlaufen. Sobald alle Tests durchlaufen haben Sie 10 Punkte. Je positiv durchlaufenen Test ein Punkt.

Starten Sie Linux, loggen Sie sich ein und starten Sie Eclipse. Entpacken Sie das Projekt Testat und erstellen Sie darüber ein neues Projekt (New C++ Project -> Location: .... /Testat). Implementieren Sie die Funktionen und Methoden in den folgenden Dateien: Studi.cpp, simple\_math.cpp.

Sie dürfen/sollen in den Dateien neue Funktionen/fehlende Methoden realisieren. Sie dürfen, müssen aber nicht, *private* Methoden in den Klassen (in den Header-Dateien) hinzufügen. Sonst darf in den Header-Dateien nichts verändert werden. Verändern Sie keine anderen Dateien. Andere Dateien werden vor der Auswertung auf Ihren ursprünglichen Zustand zurückgesetzt. „Builden“ Sie das Projekt (Project -> Build Project) oder mit Strg-B. Lassen Sie die Unit-Tests im Projekt laufen.

- Im Projekt auf Binaries -> Testat
- Rechte Maustaste und Run as -> Run Configurations
- Auf C/C++ Unit Doppelklicken oder vorhandenes Testat auswählen. Im Tab Rechts C/C++ Testing als Tests Runner wählen wir Goggle Tests Runner.
- Run

Sie sehen in einem C/C++ Unit Tab die Ergebnisse der einzelnen Tests (die Gruppen aufklappen). Die Anzahl der Tests, die korrekt (grün) laufen entsprechen Ihren Punkten. Es gibt keinen Punkt bei Speicherfehlern. Die Punkte, die Sie sehen sind die Maximalpunkte. Wer gegen den Test programmiert, zum Beispiel `return 42;` weil in dem Beispiel eine 42 erwartet wird, erhält im Nachhinein keine Punkte. Schauen Sie sich die Tests in `test_*.cpp` an. Die Inhalte der Tests helfen Ihnen zu verstehen was klappt, beziehungsweise nicht klappt.

**Aufgabe 1.** Es sind drei einfache mathematische Funktionen zu realisieren. Die Prototypen sind in der Datei `simplemath.h` zu finden.

```
long quadrat(long n);  
long kubik(long n);  
long hoch4(int n);
```

Sie realisieren die Funktionen in `simplemath.cpp`. Probieren Sie mal eine Hilfsfunktion zu realisieren und debuggen Sie diese während die Tests durchlaufen. Die Tests sind in der Datei `test_simplemath.cpp`. Sie dürfen Dateien der Form `test_*.cpp` nicht ändern.

Schauen Sie sich die Nachrichten (Messages) der Tests gut an. Sie sehen zu jedem Test, ob es geklappt hat (grün) oder ob nicht (rötlich) und dann den Grund warum nicht. Sie können auf Meldung klicken und springen im Test auf die Zeile, die nicht den Erwartungen entsprach. Wenn Sie sehr böse Fehler machen, dann bricht das Programm ab und die Tests laufen nicht durch. Vermeiden Sie sehr böse Fehler. Wenn Sie Syntaxfehler bei der Abgabe im Code haben, dann kompiliert das Programm nicht. Das sind dann 0 Punkte.



**Aufgabe 2.** Der Prototyp einer Klasse `Studi` ist vorgegeben in der Datei `Studi.h`.

```
// Eine Studi-Klasse, die einen Studierenden repräsentiert
class Studi {
public:
    Studi();
    Studi(long matnr, std::string vorname, std::string nachname);
    Studi(const Studi&);
    ~Studi();
    long matnr() const;
    std::string vorname() const;
    std::string nachname() const;
    // Akzeptiert nur 7-stellige Matrikelnummern ohne führende Nullen,
    // false wenn nicht akzeptiert (alter Wert bleibt), true sonst
    bool set_matnr(long matnr);
    // Das übergebene Bild wird als Bytearray kopiert in einen eigenen
    // Speicherbereich, vorhandener Speicher wird gelöscht
    void set_image(char *image, int size);

    // get_* ist nur zur Verwendung im Test, nicht ändern
    char *get_image() const { return _image; }
    int get_image_size() const { return _image_size; }
private:
    long _matnr;
    std::string _vorname;
    std::string _nachname;
    char *_image;
    int _image_size;
};
```

Sie realisieren die passenden Funktionen und Methoden in der Datei `Studi.cpp`. Als Matrikelnummer sollen nur 7stelligen Zahlen (ohne führende Nullen) akzeptiert werden. Bei Ablehnung bleibt der alte Wert und es wird `false` zurückgeben. Wir müssen Arrays von Studis anlegen können, die nicht initialisiert werden. In dem Fall ist die Matrikelnummer 0 ok. Beim Kopieren eines Studis wird die Matrikelnummer nicht mitkopiert. Achten Sie darauf den Speicher von `_image` richtig zu verwalten beim Kopieren, Zuweisen und Löschen. Ein Bild im Freispeicher muss in der Kopie seinen eigenen Speicherbereich haben.

Viel Spaß und viel Erfolg!

moodle OOP