

## **OOP – JAVA - Prof. Kaendl - Ausgearbeitete Theoriefragen**

### **Data Encapsulation ?**

Die Kapselung in Java dient im allgemeinen zur Daten & Manipulationsicherung von aussen. Nur das Objekt kennt seinen Zustand. Um den Status (Variablen, etc) des Objekts zu verändern & abzufragen werden im Objekt eigene Methoden dafür entworfen. Der Aufrufer kann weiters die Implementierungen des Objekts nicht.

### **Was man für Testing braucht ?**

- 1.) Testfälle - Mengen von Test-Inputs, Ausführungsbedingungen und erwarteten Resultaten
- 2.) Testkriterien - Kriterien, die erfüllt werden müssen, um einen vorgegebenen Test zu bestehen
- 3.) Testdokumentation - Spezifikation des Testfalles, Testplan, Testbericht
- 4.) Tester – Person die diese Aufgabe bewältigt (oft ein Team )☺

### **Zweck des Testens ?**

- 1.) Um Fehler zu finden
- 2.) Für die Verifikation des spezifizierten Verhaltens
- 3.) Um die Qualität zu bewerten

### **Erkläre das Observer Pattern.**

Das Observer Pattern wird als Interface implementiert und arbeitet mit dem Subject Interface zusammen. Idee : Definiere eine 1:n Abhängigkeit zwischen Objekten, so dass die Änderung des Zustands eines Objekts dazu führt, dass alle Objekte benachrichtigt und automatisch aktualisiert werden. Die Änderungen eines Objekts können sich auf andere Objekte auswirken, ohne dass das geänderte Objekt die anderen genauer kennen muss. Das Subject trägt sich im Observer ein, um von ihm bei Änderungen benachrichtigt zu werden (zB registerInterest() ). Das Subject kann sich auch wieder austragen (unregisterInterest() ).

### **Was ist eine Klasse in OOP ?**

Eine Klasse ist ein „Prägestempel“ oder Prototyp, der die Variablen und die Methoden gemeinsam für alle Objekte einer bestimmten Art definiert.

### **Was ist Polymorphismus?**

Polymorphismus bedeutet direkt übersetzt etwa "Vielgestaltigkeit" und bezeichnet zunächst einmal die Fähigkeit von Objektvariablen, Objekte unterschiedlicher Klassen aufzunehmen. Das geschieht allerdings nicht unkontrolliert, sondern beschränkt sich für eine Objektvariable des Typs X auf alle Objekte der Klasse X oder einer daraus abgeleiteten Klasse. Objektorientierte Programmierung erlaubt eine natürliche Modellierung vieler Problemstellungen. Sie vermindert die Komplexität eines Programms durch Abstraktion, Kapselung, definierte Schnittstellen und Reduzierung von Querzugriffen. Sie stellt Hilfsmittel zur Darstellung von Beziehungen zwischen Klassen und Objekten dar, und sie erhöht die Effizienz des Entwicklers durch Förderung der Wiederverwendung von Programmcode.

### **Was ist ein Objekt in OOP?**

Objekte nehmen Platz im Speicher ein und haben eine zugeordnete Adresse wie ein Datensatz in Pascal oder eine Struktur in C. Dieser Speicher beinhaltet den Zustand des Objekts zu jedem gegebenen Zeitpunkt, und mit einem Objekt ist eine Menge von Prozeduren/Funktionen verbunden, welche die sinnvollen Operationen an diesem Objekt definieren.

## **Was ist der Unterschied zwischen White & Black-Box-Tests ?**

Beim White B. Test ist die Struktur des Programms bekannt und somit kann man alle Programmteile sorgfältig prüfen. Dies ist beim Black B. Testen nicht möglich, da man die innere Struktur nicht kennt & nur vom externen Verhalten des Programms Rückschlüsse auf die Fehlerfreiheit machen kann.

## **Arten des Testens ?**

- Testen einer Einheit
- Integrationstesten
- System / Akzeptanztesten
- Leistungstesten
- Stresstesten
- Sicherheitstesten
- Regressionstesten

## **White box testing ?**

Bei dem White Box Testverfahren, auch Struktur-Verfahren bezeichnet, kann das Programm, da die innere Struktur bekannt ist, mit Programmspezifischen Testdaten durchlaufen werden. Das Ergebnis wird mit dem erwarteten Ergebnis verglichen, um so Fehler zu lokalisieren. Es gibt weiters verschiedene Abdeckungsgrade ( ob z.B. alle Methoden jeder Klasse geprüft wurden..)

## **Black Box testing?**

Beim Black Box Testverfahren ist die Programmstruktur nicht bekannt. Die ganze Beurteilung des Testes beruht auf dem externen Verhalten des Programms. Das erwartete Ergebnis kann in diesem Fall nur aus den Spezifikationen ( für was ist das Programm geschrieben / zuständig,... ) des Programms eruiert werden. Das Ergebnis wird wieder mit dem erwarteten Ergebnis verglichen.

## **Was bedeutet Static ?**

Eine statische Variable wird auch Klassenvariable genannt. Variablen und Methoden mit dem Attribut static sind nicht an die Existenz eines konkreten Objekts gebunden, sondern existieren vom Laden der Klasse bis zum Beenden des Programms. Das static-Attribut beeinflusst bei Membervariablen ihre Lebensdauer und erlaubt bei Methoden den Aufruf, ohne daß der Aufrufer ein Objekt der Klasse besitzt, in der die Methode definiert wurde. Wird das Attribut static nicht verwendet, so sind Variablen innerhalb einer Klasse immer an eine konkrete Instanz gebunden. Ihre Lebensdauer beginnt mit dem Anlegen des Objekts und dem Aufruf eines Konstruktors und endet mit der Freigabe des Objekts durch den Garbage Collector.

## **Erkläre das Singleton Pattern.**

Der Zweck des S. Patterns ist, nur eine Instanz einer Klasse zu erlauben. Die Zustandsspeicherung erfolgt in einer Klassenvariable ( eine Variable mit dem Klassennamen ) um die Klasse nicht mit einer Instanz zu assoziieren. Der Konstruktor dieser Klasse wird als private deklariert, um zu verhindern, dass die Klasse mehrmals instanziiert wird. In einer statischen Methode ( meist getInstance() ) wird die einzige Instanz dieser Klasse kreiert. Die Statische Methode kann von jeder anderen Klasse über „klassenname.statischeMethode“ aufgerufen werden.