Gliederung der Folien

1. Einführung

1.1. Gliederung

1.2. Literatur

1.3. Einstieg in Java

- 1.3.1. Geschichte von Java
- 1.3.2. „Hello World“
- 1.3.3. Übersetzen eines Java-Programms

1.4. Die Entwicklungsumgebung Eclipse

2. Grundelemente der Programmierung

2.1. Variablen

- 2.1.1. Elementare Datentypen
- 2.1.2. Typumwandlung (Elem. Datentypen)
- 2.1.3. Darstellung von Literalen

2.2. Operatoren

- 2.2.1. Übersicht
- 2.2.2. Prioritätentabelle

2.3. Kontrollstrukturen

- 2.3.1. Sequenz
- 2.3.2. Verzweigung
  - 2.3.2.1. if -Anweisung
  - 2.3.2.2. Logische Ausdrücke
  - 2.3.2.3. switch-Anweisung
- 2.3.3. Schleifen
  - 2.3.3.1. for-Schleife
  - 2.3.3.2. while- und do-while-Schleife
  - 2.3.3.3. Schleifen allgemein

2.4. Arrays

- 2.4.1. Arrays allgemein
- 2.4.2. Wertetyp und Referenztyp
- 2.4.3. Beispiel

2.5. Funktionen

- 2.5.1. Beispiel Funktionen

2.5.2. Beispiel Rekursive Funktionen
2.5.3. Beispiel Funktionsstack
2.5.4. Funktionsstack allgemein
2.5.5. Rekursive Funktionen allgemein

2.6. Exceptions
2.6.1. Allgemein
2.6.2. Beispiel
2.6.3. Exceptions weiterreichen (`throws`)

3. Objektorientierung

3.1. Klassen und Objekte
3.1.1. Von Elementaren zu Abstrakten Typen
3.1.2. Beispiel
3.1.3. Anwendung
3.1.4. Begriffe
3.1.4.1. Klasse
3.1.4.2. Objekt
3.1.4.3. Methode
3.1.4.4. Konstruktor
3.1.4.5. Defaultkonstruktor
3.1.5. Überladen von Methoden
3.1.5.1. Allgemein
3.1.5.2. Beispiel
3.1.6. this
3.1.7. Objekte im Speicher
3.1.8. Garbage Collection
3.1.9. static
3.1.9.1. Allgemein
3.1.9.2. Beispiele
3.1.10. final
3.1.11. Zeichenketten
3.1.11.1. Die Klasse `String`
3.1.11.2. Wichtige Methoden
3.1.11.3. Beispiel
3.1.11.4. Die Klasse `StringBuffer`
3.1.12. Über nichtinitialisierte Variablen und dem Wert `null`
3.1.12.1. Nichtinitialisierte lokale Variablen
3.1.12.2. „Nichtinitialisierte“ Attribute
3.1.12.3. Der Wert `null`
3.1.12.4. Allgemein

3.2. Datenkapselung
3.2.1. Allgemein
3.2.2. Beispiel

3.2.3. Kommunikation zwischen Objekten
3.2.4. Sichtbarkeit
3.2.4.1. Konfigurationsmöglichkeiten
3.2.4.2. Sichtbarkeit bezieht sich auf Klassen
3.2.5. Packages
3.2.5.1. Allgemein
3.2.5.2. Syntax
3.2.5.3. Packagehierarchie
3.2.5.4. Namensräume
3.2.5.5. Sichtbarkeit von Klassen
3.2.5.6. Das Package `java.lang`
3.2.6. Kohäsion und Kopplung
3.2.6.1. Allgemein
3.2.6.2. Beispiel

3.3. Vererbung
3.3.1. Beispiel
3.3.1.1. Aufgabenstellung
3.3.1.2. Die Klasse `Kreis`
3.3.1.3. Exkurs: Escape-Sequenzen
3.3.1.4. Gegenüberstellung `Rechteck` und `Kreis`
3.3.1.5. Problem: Redundanz
3.3.1.6. Lösung: Vererbung
3.3.1.7. UML-Klassendiagramm
3.3.1.8. Die Oberklasse `Flaeche`
3.3.1.9. Die Unterklassen `Rechteck` und `Kreis`
3.3.2. super
3.3.3. Reihenfolge der Konstruktoraufrufe
3.3.4. toString()
3.3.5. Die Klasse `Object`
3.3.6. Vererbung allgemein
3.3.7. Typumwandlung
3.3.7.1. Vorschau
3.3.7.2. Typumwandlung allgemein
3.3.7.3. Typumwandlung elementarer Typen (Wiederholung)
3.3.7.4. Typumwandlung abstrakter Typen
3.3.7.5. Vergleich elementar und abstrakt

3.4. Polymorphie
3.4.1. Beispiel
3.4.2. Erläuterung
3.4.3. Späte Bindung
3.4.4. Begrifferklärung
3.4.5. Polymorphie in Java
3.4.6. Überschreiben von Methoden
3.4.7. Virtuelle Methodentabellen (VMT)
3.4.8. Nachteile der Polymorphie
3.4.9. Allgemein

3.5. Weitere Objektorientierte Konzepte in Java
3.5.1. Abstrakte Methoden und abstrakte Klassen
3.5.2. Innere Klassen (u.a.)
3.5.2.1. Aufgabenstellung Zeichenprogramm
3.5.2.2. Die Klasse `Zeichenprogramm`
3.5.2.3. Anpassen der `paint()`-Methoden
3.5.2.4. Das Ergebnis
3.5.2.5. Änderung auf anonyme Klassen
3.5.3. Interfaces
3.5.3.1. Allgemein
3.5.3.2. Beispiel
3.5.3.3. Warum Interfaces?

3.6. Von der Prozeduralen zur Objektorientierten Programmierung
3.6.1. Programmierparadigmen
3.6.2. Programmierung im Kleinen: Strukturierte Programmierung
3.6.3. Programmierung im Großen: Objektorientierte Programmierung
3.6.4. Vergleich zwischen Prozeduraler und Objektorientierter Programmierung

3.7. Zusammenfassung OOP

4. Erweiterte Konzepte in Java

4.1. Generics
4.1.1. Beispiel `Safe`
4.1.1.1. Implementierung ohne Generics
4.1.1.2. Problem der Anwendung
4.1.1.3. Implementierung mit Generics
4.1.2. Allgemein
4.1.3. Die Klasse `Gruppe`

4.2. Exkurs: Collections in Java
4.2.1. Grundtypen
4.2.2. Die Klasse ArrayList
4.2.3. Zuordnungen
4.2.3.1. Analogie Telefonbuch
4.2.3.2. Binäre Suche
4.2.3.3. Hashtabelle
4.2.3.4. Anwendung
4.2.3.5. Funktionsweise von `HashMap`
4.2.3.6. Wichtige Methoden von `HashMap`

4.3. Erweiterte for-Schleife
4.3.1. Erstes Beispiel
4.3.2. Der zweite Versuch

4.4. Variable Anzahl von Methodenargumenten

4.4.1. Beispiel
4.4.2. Allgemein

4.5. Die neue Version der Klasse `Gruppe`

5. Design Patterns

5.1. Beispiel

5.1.1. Eine „Zeichnung“
5.1.2. Die Ausgabe
5.1.3. Das Klassendiagramm

5.2. Beschreibung

5.2.1. Herkunft
5.2.2. Definitionen

5.3. Composite-Pattern

5.4. Weitere Patterns