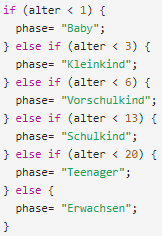
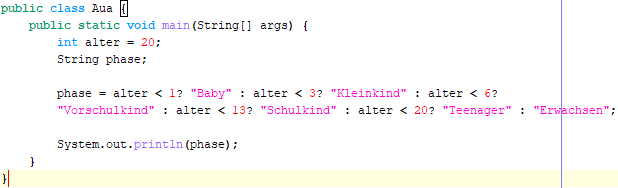
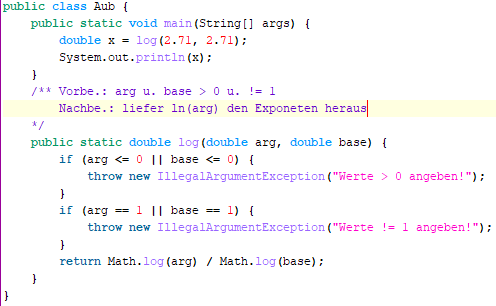
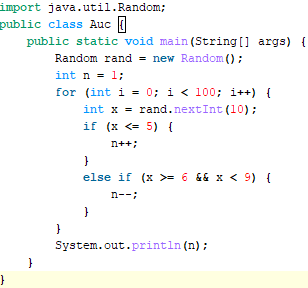
**PR1 – Formular für Lesenotizen**

**WS2020/21**

|  |  |  |  |
| --- | --- | --- | --- |
| Nachname  Abdel Kader | Vorname  Schehat | Matrikelnummer  1630110 | Abgabedatum:  12.11.2020 |

**Bedingte Ausführung (L4.4 – 4.6) & Programmlogik und indefinite Schleifen (L.5.1 – 5-3)**

**Lernzielfragen:**

1. Wandeln Sie den folgenden Programmtext in eine einzige Zuweisung der Form phase = ... ; um:
2. Formulieren Sie eine sinnvolle Vorbedingung für die folgende Methode. Ergänzen Sie in der Methode eine Überprüfung der Vorbedingung, welche zu einer geeigneten Exception führt, falls die Vorbedingung nicht erfüllt ist.
3. Schreiben Sie ein Programm, das eine sich zufällig entwickelnden Zahlenfolge simuliert:

Startwert: 1 mit Wahrscheinlichkeit von 60% wird 1 addiert, mit einer Wahrscheinlichkeit von 30% wird 1 subtrahiert, mit einer Wahrscheinlichkeit von 10% bleibt der Wert konstant. 100 Durchläufe

1. Schreiben Sie ein Programm, das den Benutzer nach einem neuen Passwort fragt, und zwar so lange, bis die Eingabe mindestens 8 Zeichen lang ist. Am Ende soll die Anzahl der Versuche ausgegeben werden.

* unnötig

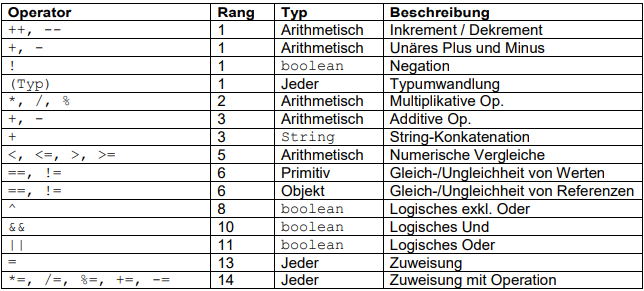
1. Was stimmt mit den folgenden Ausdrücken nicht?

* a != 5 || 6 => 6 ist kein boolescher Ausdruck
* 5 <= i <= 8 => Zahlenintervalle werden nicht so in Java angegeben

1. Welchen Wert haben die folgenden Ausdrücke (nehmen Sie an, dass die Variable lottogewinn vom Typ boolean existiert)?

* 17 < 15 && lottogewinn = > false
* 17 > 15 || lottogewinn = > true
* 13 != 12 ^ 12 != 11 = > false
* !(Math.PI < 4) = > false

**Notizen:**

* Präzedenzen

**Return**

* Alle if/else Pfade müssen einen return Befehl enthalten
* Compiler kann nicht Bedingungen vergleichen z.B.: if (a <= b) u. else if (a > b) return Statements besitzen kommt es trotzdem zu einer Fehlermeldung. Immer ende Methode return
* For-schleife mit if/else muss auch return berücksichtigt werden wenn Schleife/if kein Mal läuft

**Random**

* Import java.util.Scanner
* Methoden: nextInt(<max. Zahl>), nextDouble(<max Zahl>)
* Erzeugung Intervall min. – max.: nextInt(max – min + 1) + min

**Ternäre Operator**

* Bedingte Auswertung – Syntax: <type ><varname> = <boolean expr>? <expr1> : <expr2>;
* Wenn <boolean expr> wahr, dann <expr1> zugewiesen sonst <expr2>

**Indefinite Schleife - while-Schleife**

Sentinel-Schleife: Schleife, die bis zu Sentinel-Wert läuft. Nutzt häufig das Zaunpfahlstil

* Syntax: while (<condition>) {

<statement>;

}

**Sentinel-Werte**

* Ein spezieller (Eingabe-)Wert, der das Ende einer Folge von Daten(-eingaben) signalisiert