

Info B Testat Blatt 3

	P	G1	G2	G3	G4	G5	G6	G7	G8
Aufgabe 1:EBNF und Syntaxdiagramm	30								
Operation	6								
Operator	1								
Bruch	1								
Nenner	2								
Zähler	3								
Zahl	2								
Zahl ohne Null	2								
Kein Nullteiler	1								
Vorzeichen (vorzeichenbehafteter Nenner ist zusätzlich auch in Ordnung)	1								
immer korrekt mit Leerzeichen (keine Leerzeichen am Anfang oder am Ende)	2								
Terminalzeichen in Anführungszeichen gesetzt	1								
Syntaxdiagramm (Punktatzug, wenn Syntaxdiagramm nicht EBNF wieder gibt. OBACHT: online EBNF oft anders als im Skript definiert)	8								
Aufgabe 2:Bruchrechner	25								
add	1								
subtract	1								
korrekter regulärer Ausdruck + Erklärung (führende Nullen i.O., minus im Nenner auch oke)	10								
Methode parseFraction mit matches, split und parseInt	2								
Rechner	0								
funktioniert	4								
Korrekte Anzahl Eingabeelemente (immer drei)	1								
Korrektheit der Operatoren	1								
Korrektheit der Operanden	1								
Operanden geprüft, bzw. default in switch-case	1								
keine Exceptions werden geworfen, nur Fehlermeldungen ausgegeben	1								
javadoc	2								
Aufgabe 3:equals und hashCode	28								
Konstruktor und Vererbung? impliziter Aufruf des super Konstruktors od. explizit aufrufen; werden nicht vererbt	3								
Testklasse, alle nötigen Fälle abgedeckt, 1 Punkt für nennen und erklären von fehlenden Fällen	0								
equals null	2								
Reflexivität	2								
Symmetrie	2								
Transitivität	2								
Konsistenz	2								
Probleme bei Vererbung: gleiche PersonWerte bei Person p1 und Student s1 aber unterschiedliche Ergebnisse	2								
hashCode überprüft	2								
für beide Klassen alles geprüft	2								
Fehler erkannt: getClass anstatt instanceof	2								
Lösung (a) ist sinnvoll, falls Vererbungshierarchie nicht verletzt, Lösung (b) auch oke (wenn nicht erkannt, gibt es noch 2 Punkte, wenn man Unterschied zwischen instanceof und getClass erklären kann	4								
Beziehung zwischen equals und hashCode erklären	3								
Aufgabe 4:Dynamisches Binden	17								
Dynamisches und statisches Binden	5								
Pro Stelle richtig 1 Punkt	0								
Stelle 1: motorboat (statisch gebunden)	1								
Stelle 2: boat (Cast auf Boat, statisch gebunden)	1								
Stelle 3: motorboat (dyn. Bindung der Methode, innerhalb statische Bindung)	1								
Stelle 4: motorboat (in getBoatClass() von Boat dyn. Bindung, also getType() von MotorBoat wird aufgerufen)	1								
Stelle 5: canoe (statisch gebunden)	1								
Stelle 6: boat (dynamisch gebunden, innerhalb der Methode statisch an this)	1								
Stelle 7: boat (siehe Stelle 4, allerdings hat Canoe keine getType() Methode, also getType() von Boat)	1								

Stelle 8: boat (Referenz vom Typ Boat, Auswirkungen wie ein Cast, statische Bindung)	1								
Stelle 9: boat (dynamische Bindung)	1								
Stelle 10: canoe (statische Bindung)	1								
Cast von motorBoat auf canoe möglich?	2								
Summe	100								

G1: Di, 10:30: Hannah Lewerentz, Nenita Josephine Schick

G2: Di, 11:00: Freya Berstermann, Irina Vortkamp

G3: Di, 11:30: Felix Nardmann, Benjamin Tolksdorf

G4: Di, 14:00: Daniel Statkus, Rene Brinkege

G5: Di, 14:30: Daniel Pohl, Bianca Krömer

G6: Di, 16:00: Darren Kopatz, Alexander Miller

G7: Di, 16:30: Lisa Rexilius, Jannis De Riz

G8: Mi, 12:00: Ertan Keser, Lukas Horst