Transcriptz

Tasks

Tasks		Stories	Related Documents/Notes
☒1	Settle on an approach for handling missing values	S6	Use `Double` objects rather than `double` values and use null to represent missing values
☒2	Design a utility class for working with missing elements	S6	The specifications for the `Numerics` class
☒3	Design unit tests for the class in task 2	S6	The description of the unit tests for the `Numerics` class
☐4	Implement the unit tests in task 3	S6	
☐5	Implement the class in task 2	S6	
☐6	Test (and debug if necessary) the class in task 8	S6	
☒7	Design an approach for representing non-missing numbers	S6	UML diagram and specifications for the `LabeledDouble` class
☒8	Design an approach for representing missing numbers	S6	UML diagram and specifications for the `LabeledDouble` class
☐9	Design unit tests for the class in tasks 7-8	S6	
☐10	Implement the unit tests for the class in task 7-8	S6	
☐11	Test and debug the class in task 7-8	S6	
☒12	Determine how to handle size issues		The specifications for the `SizeException` class
☒13	Design a class that can be used to select courses with passing grades	S1-S5	The specifications for the `ThresholdFilter` class
☒14	Determine whether the class in task 13 should implement an interface	S1-S5	The `Filter` interface in the design document
☒15	Design unit tests for the class in task 13	S1-S5	The description of the unit tests for the `ThresholdFilter` class

☐16	Implement the unit tests for the class in task 13	S1-S5	
☐17	Implement the class in task 13	S1-S5	
☐18	Test (and debug, if necessary) the class in task 13	S1-S5	
☒19	Design a class that can be used to calculate weighted totals	S2	The specifications for the `WeightedTotalCalculator` class
☒20	Design a class that can be used to calculate weighted totals	S5	The specifications for the `WeightedAverageCalculator` class
☒21	Determine whether the classes in tasks 19 and 20 should implement an interface	S3-S4	The `Calculator` interface in the design document
☒22	Design a class that can be used to represent letter grades	S1	The specifications for the `LetterGrade` enum
☒23	Design unit tests for the class in task 24		
☐24	Implement the unit tests for the class in task 24		
☐25	Implement the class in task 24		
☐26	Test (and debug, if necessary) the class in task 24		
☒27	Create two integration/system tests in which the students pass all of their courses.	S1-S6	Tests: Lisa, Maggie
☒28	Create an integration/system test in which the student passes some courses and fails other courses.	S1-S6	Tests: Bart, Homer
☒29	Create one integration/system test with missing grades.	S7	Tests: Marge
☒30	Implement the main class.	S7	`TranscriptzH1.java`
☐31	Create Eclipse "Run Configurations" for each system test	S1-S7	
☐32	Test the system	S1-S7	
☐33	Debug the system (if necessary)	S1-S7	