

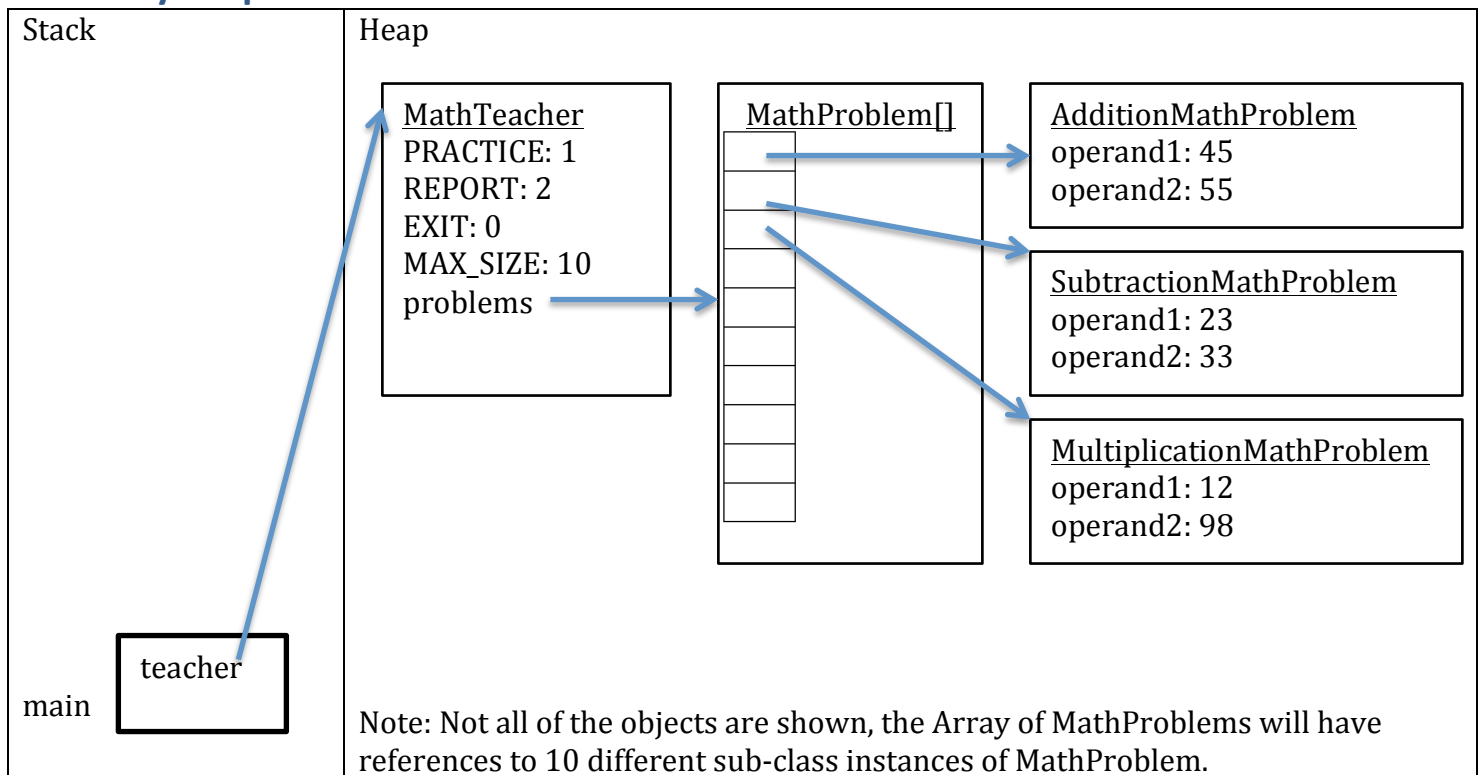
CST8132: Object Oriented Programming

Assignment #2: Math Practice Sample Solution

Problem Statement:

- In this assignment a program was created to allow users to practice basic arithmetic of addition, subtraction, and multiplication and to receive feedback.
- Inheritance and polymorphism were used to implement the program in that there was an abstract superclass, MathProblem, containing operator fields as well as a mix of concrete and abstract methods.
- MathProblem was subclassed by AdditionMathProblem, SubtractionMathProblem, and MultiplicationMathProblem that each override the inherited abstract methods to provide appropriate implementations performing addition, subtraction and multiplication respectively.
- The core of the program is an Array of references to MathProblem type, each reference refers to an instance of one of AdditionMathProblem, SubtractionMathProblem, or MultiplicationMathProblem permitting polymorphic method calls to be made at run time.

Memory Map:



Test Plan

In Addition to the completed program, also include a test plan that tests your program, here is a sample of the expected format. Create one table per class, with one or more tests per method.

Functionality being tested	What are we testing?	Pre-Condition Any special context or program state to take note of?	Test Method How will we test?	Post-Condition Expected Results	Post-Condition Actual Results
RandomUtil .nextInt(int)	Does it return an integer between 0 to 9 when 10 is the argument value?	Program has loaded, a call to next was made with 10 as argument.	Step through with debugger and examine returned value.	Value from 0 to 9 returned	Matches
MathTeacher .printReport()	What if user prints report before taking a test?	The array of MathProblem references is null (not null elements but null itself)	Watch program output for correct message "Please take test first"	Should get correct message on screen for user, no program crash	Matches

Full sample test plan is not provided, however the expectation was that the following classes and methods would have been in tables:

MathProblem:

- Constructor – does it set values between 1 to 100 for operands 1 and 2
- getOperand1() – does it return the correct value?
- setOperand1(int) – does it set a value correctly?
- getOperand2() – does it return the correct value?
- setOperand2(int) – does it set a value correctly?
- getAnswer() – a note that it will be tested in subclass test tables
- getUserAnswer()
- setUserAnswer(int)
- getQuestionText() – a note that it will be tested in subclass test tables
- getQuestionTextWithFeedback() – a note that it will be tested in subclass test tables

AdditionMathProblem

- getAnswer() – addition performed?
- getQuestionText() – addition?
- getQuestionTextWithFeedback() – addition?

SubtractionMathProblem

- getAnswer() – subtraction performed?
- getQuestionText() – subtraction?
- getQuestionTextWithFeedback() – subtraction?

MultiplicationMathProblem

- getAnswer() – multiplication performed?
- getQuestionText() – multiplication?
- getQuestionTextWithFeedback() – multiplication?

RandomUtil

- nextInt(max: int) – returns int from 0 up to value specified (exclusive)

MathTeacher

- runMenu() – correct menu is displayed, correct methods are called for user input
- initializeProblems() – loads array with 10 references to sub classes of MathProblem
- printReport() – handles situation of no practice tests yet?, provides correct outputs?
- practiceMath() – shows user 10 questions, inputs answers into appropriate problem objects?