

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Кафедра вычислительной техники

Тестирование программного обеспечения

Лабораторная работа № 1

*Выполнил
Сальников Юрий Антонович
Группа Р3311*

Санкт-Петербург
2017

Вариант

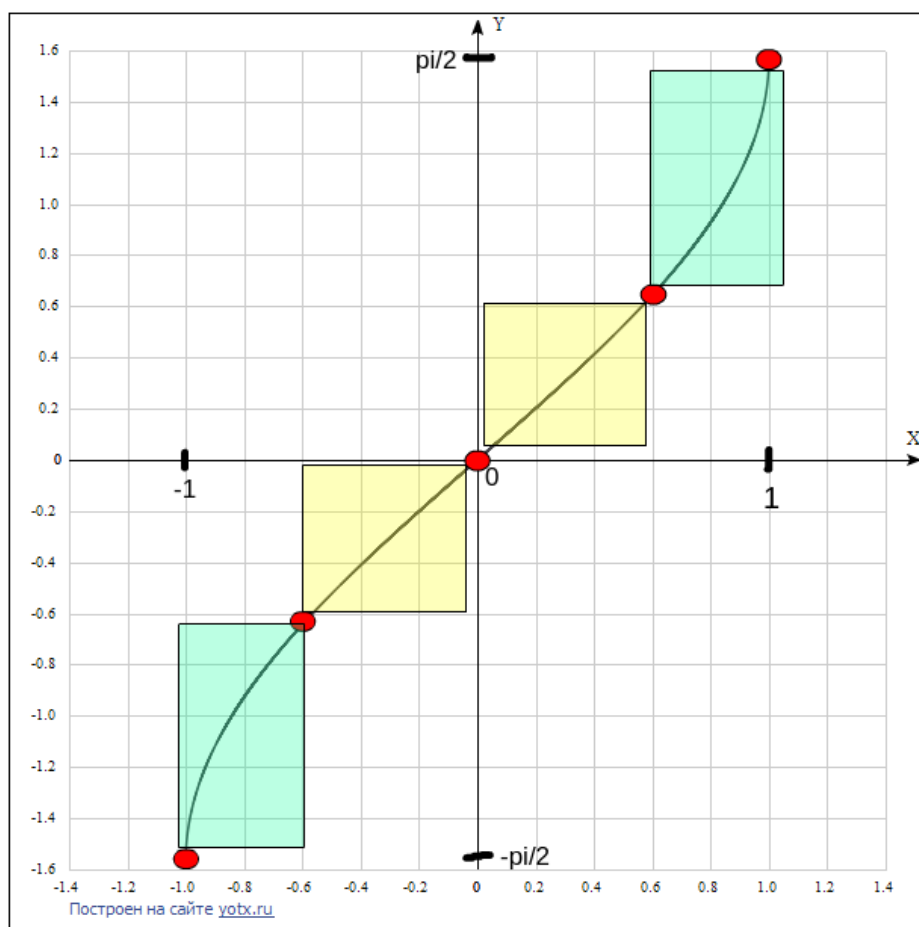
Введите вариант:

9876543

1. Функция $\arcsin(x)$
2. Программный модуль для работы с Фибоначчиевой кучей (Logical Representation, <http://www.cs.usfca.edu/~galles/visualization/FibonacciHeap.html>)
3. Описание предметной области:

“Марвин наблюдал за ней с холодным презрением. Его логические схемы в отвращении застрекотали и начали манипулировать идеей применить к двери физическое насилие. Затем они щелкнули и сказали: "А стоит ли обращать внимание? Ради чего?". Тогда схемы решили позабавиться, произведя сравнительный анализ молекулярных компонентов двери и клеток человеческого мозга. После этого они на бис измерили уровень водорода в близлежащем кубическом парсеке космоса, а потом им все надоело, и они отключились. С судорогой отчаяния робот повернулся.”

Задание 1



Функция определена при $-1 < x < 1$, поэтому -1 и 1 — граничные значения. Также в качестве граничных значений рассматривается $x = 0$, так как при переходе через него функция меняет знак, и $x = \pm 0,6$, так как приблизительно в этих точках меняется характер роста функции.

Таким образом, после расстановки граничных значений получаем 4 класса эквивалентности (выделены на рисунке). Можно выделить еще один класс эквивалентности — область, на которой функция не определена, т. е. при $x < -1$ и при $x > 1$.

Для тестирования понадобится проверить по три случая на каждое граничное значение и по два — на каждый класс эквивалентности: таким образом будет обеспечено достаточное тестовое покрытие.

Задание 2

Тестируемый эталонный набор переходов был получен в результате следующих действий:

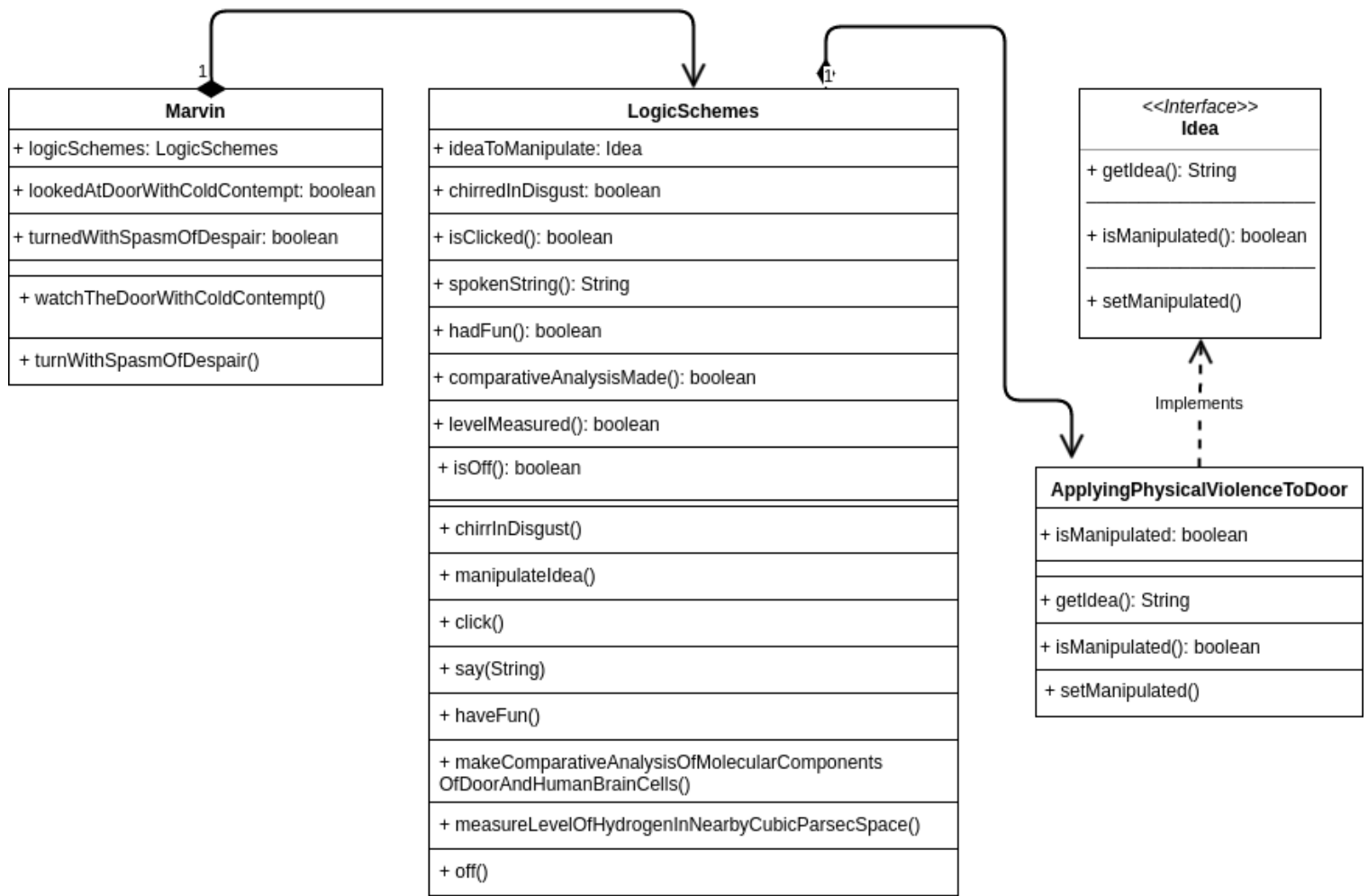
- добавление 1
- добавление -5
- добавление 4
- извлечение минимума
- извлечение минимума

Результаты логирования для вышеуказанного набора действий выглядят так:
012013401356NKLM89AABFKLMGJDDE1256NOPL13KLM89ADE12

Данные переходы образуют достаточное тестовое покрытие, так как охватывают практически все рассматриваемые состояния программы. Однако, в связи с большим количеством состояний, построить граф переходов не представляется возможным.

Задание 3

UML-диаграмма классов:



Вывод

В ходе выполнения данной лабораторной работы было проведено модульное тестирование различных программных компонентов. Таким образом, были изучены основные методы модульного тестирования, такие, как тестирование «черного» ящика (анализ эквивалентности с поиском граничных значений) и тестирование «белого» ящика (тестирование логированием пошагового исполнения алгоритма для эталонного набора).