Министерство науки и высшего образования Российской Федерации федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» Институт информатики и кибернетики Кафедра технической кибернетики

Лабораторная работа №4

По дисциплине «Обьектно-ориентированное программирование»

Студент: Кораблев Д.С.

Группа: 6203-010302D

Самара, 2025

1: Дополнение классов табулированных функций

В классы ArrayTabulatedFunction и LinkedListTabulatedFunction добавлены новые конструкторы, принимающие готовый массив объектов FunctionPoint. Конструкторы выполняют проверки:

- Если точек меньше двух или они не упорядочены по возрастанию абсциссы (X), выбрасывается IllegalArgumentException.
- Обеспечена корректная инкапсуляция: внутренние структуры данных инициализируются с защитным копированием переданного массива.

2: Рефакторинг иерархии интерфейсов

1. Создан базовый интерфейс Function, описывающий общие свойства математической функции одной переменной:
   - double getLeftDomainBorder() – левая граница области определения.
   - double getRightDomainBorder() – правая граница области определения.
   - double getFunctionValue(double x) – значение функции в точке X.
2. Интерфейс TabulatedFunction теперь расширяет (наследует) интерфейс Function. Соответствующие методы (getLeftDomainBorder, getRightDomainBorder, getFunctionValue) были исключены из объявления TabulatedFunction, так как они теперь унаследованы от Function.

3: Реализация аналитических (базовых) функций

Создан пакет functions.basic, содержащий реализации элементарных функций:

- Абстрактный класс TrigonometricFunction предоставляет общую реализацию методов получения границ области определения для тригонометрических функций.
- От него унаследованы конкретные классы: Sin, Cos, Tan. Их метод getFunctionValue использует стандартные методы Math.sin(), Math.cos(), Math.tan().
- Также созданы независимые классы для логарифма (Log) и экспоненты (Exp), реализующие интерфейс Function.

4: Создание функций-комбинаторов (мета-функции)

Создан пакет functions.meta, содержащий классы для создания сложных функций из более простых:

- Sum, Mult – реализуют сумму и произведение двух функций.
- Power – возводит функцию в заданную степень.
- Shift, Scale – выполняют сдвиг и масштабирование аргумента и/или значения функции.
- Composition – реализует композицию двух функций (f(g(x))).

Каждый класс конструктора принимает исходные функции (и параметры) и вычисляет новые границы области определения и значения.

5: Утилитарный класс для операций над функциями

Создан служебный класс Functions с приватным конструктором (невозможно создать его экземпляр). Содержит статические фабричные методы для удобного создания функций-комбинаторов:

- shift(), scale(), power(), sum(), mult(), composition().

Каждый метод принимает исходные функции и параметры, и возвращает соответствующий объект класса из пакета functions.meta.

6: Утилитарный класс для работы с табулированными функциями

Создан служебный класс TabulatedFunctions (также с приватным конструктором). Реализован ключевой метод:

- TabulatedFunction tabulate(Function f, double leftX, double rightX, int pointsCount) – создает табулированное представление аналитической функции f на отрезке [leftX, rightX] с заданным числом точек. При выходе отрезка за область определения f выбрасывает IllegalArgumentException.

7: Реализация ввода-вывода для табулированных функций

В класс TabulatedFunctions добавлены методы для сериализации/десериализации объектов TabulatedFunction:

- Для байтовых потоков (бинарный формат):
  - void outputTabulatedFunction(... OutputStream out)
  - TabulatedFunction inputTabulatedFunction(... InputStream in)
- Для символьных потоков (текстовый формат):
  - void writeTabulatedFunction(... Writer out)
  - TabulatedFunction readTabulatedFunction(... Reader in)
- Обработка исключений: Методы объявляют throws IOException. Потоки не закрываются внутри методов — ответственность за их закрытие лежит на вызывающем коде.

8: Сравнение форматов ввода-вывода

- Байтовые потоки:
  - Плюсы: Точность представления чисел, компактность, скорость.
  - Минусы: Данные не читаемы человеком.
- Символьные потоки:
  - Плюсы: Читаемый (текстовый) формат, удобен для отладки и обмена.

■ Минусы: Больший объем данных, возможна потеря точности при преобразовании чисел в строки.

9: Реализация интерфейсов Serializable и Externalizable

- Serializable (стандартная Java-сериализация):

■ Плюсы: Простота реализации (достаточно объявить интерфейс), автоматическое управление.

■ Минусы: Меньшая производительность, меньший контроль над процессом, потенциальные уязвимости безопасности.

- Externalizable (ручное управление сериализацией):

■ Плюсы: Полный контроль над форматом данных, высокая производительность, возможность улучшенной защиты данных.

■ Минусы: Сложность реализации (требуется ручное написание методов writeExternal/readExternal).

10: Конечный результат

```
Синус и косинус:
Sin =  0,000000 TabSin =  0,000000       Cos = 1,000000 TabCos = 1,000000
Sin =  0,099833 TabSin =  0,097982       Cos = 0,995004 TabCos = 0,982723
Sin =  0,198669 TabSin =  0,195963       Cos = 0,980067 TabCos = 0,965446
Sin =  0,295520 TabSin =  0,293945       Cos = 0,955336 TabCos = 0,948170
Sin =  0,389418 TabSin =  0,385907       Cos = 0,921061 TabCos = 0,914355
Sin =  0,479426 TabSin =  0,472070       Cos = 0,877583 TabCos = 0,864608
Sin =  0,564642 TabSin =  0,558234       Cos = 0,825336 TabCos = 0,814862
Sin =  0,644218 TabSin =  0,643982       Cos = 0,764842 TabCos = 0,764620
Sin =  0,717356 TabSin =  0,707935       Cos = 0,696707 TabCos = 0,688404
Sin =  0,783327 TabSin =  0,771888       Cos = 0,621610 TabCos = 0,612188
Sin =  0,841471 TabSin =  0,835841       Cos = 0,540302 TabCos = 0,535972
Sin =  0,891207 TabSin =  0,883993       Cos = 0,453596 TabCos = 0,450633
Sin =  0,932039 TabSin =  0,918022       Cos = 0,362358 TabCos = 0,357141
Sin =  0,963558 TabSin =  0,952051       Cos = 0,267499 TabCos = 0,263648
Sin =  0,985450 TabSin =  0,984808       Cos = 0,169967 TabCos = 0,169931
Sin =  0,997495 TabSin =  0,984808       Cos = 0,070737 TabCos = 0,070437
Sin =  0,999574 TabSin =  0,984808       Cos = -0,029200 TabCos = -0,029056
Sin =  0,991665 TabSin =  0,984808       Cos = -0,128844 TabCos = -0,128549
Sin =  0,973848 TabSin =  0,966204       Cos = -0,227202 TabCos = -0,224761
Sin =  0,946300 TabSin =  0,932175       Cos = -0,323290 TabCos = -0,318254
Sin =  0,909297 TabSin =  0,898147       Cos = -0,416147 TabCos = -0,411747
Sin =  0,863209 TabSin =  0,862441       Cos = -0,504846 TabCos = -0,504272
Sin =  0,808496 TabSin =  0,798488       Cos = -0,588501 TabCos = -0,580488
Sin =  0,745705 TabSin =  0,734535       Cos = -0,666276 TabCos = -0,656704
Sin =  0,675463 TabSin =  0,670582       Cos = -0,737394 TabCos = -0,732920
Sin =  0,598472 TabSin =  0,594072       Cos = -0,801144 TabCos = -0,794171
Sin =  0,515501 TabSin =  0,507908       Cos = -0,856889 TabCos = -0,843917
Sin =  0,427380 TabSin =  0,421745       Cos = -0,904072 TabCos = -0,893664
Sin =  0,334988 TabSin =  0,334698       Cos = -0,942222 TabCos = -0,940984
Sin =  0,239249 TabSin =  0,236716       Cos = -0,970958 TabCos = -0,958261
Sin =  0,141120 TabSin =  0,138735       Cos = -0,989992 TabCos = -0,975537
Sin =  0,041581 TabSin =  0,040753       Cos = -0,999135 TabCos = -0,992814
```

```
Сумма квадратов синуса и косинуса:
Arg =  0,000000 Value = 1,000000
Arg =  0,100000 Value = 0,975345
Arg =  0,200000 Value = 0,970488
Arg =  0,300000 Value = 0,985429
Arg =  0,400000 Value = 0,984968
Arg =  0,500000 Value = 0,970398
Arg =  0,600000 Value = 0,975624
Arg =  0,700000 Value = 0,999358
Arg =  0,800000 Value = 0,975073
Arg =  0,900000 Value = 0,970586
Arg =  1,000000 Value = 0,985897
Arg =  1,100000 Value = 0,984515
Arg =  1,200000 Value = 0,970314
Arg =  1,300000 Value = 0,975910
Arg =  1,400000 Value = 0,998723
Arg =  1,500000 Value = 0,974808
Arg =  1,600000 Value = 0,970691
Arg =  1,700000 Value = 0,986371
Arg =  1,800000 Value = 0,984068
Arg =  1,900000 Value = 0,970237
Arg =  2,000000 Value = 0,976203
Arg =  2,100000 Value = 0,998094
Arg =  2,200000 Value = 0,974549
Arg =  2,300000 Value = 0,970802
Arg =  2,400000 Value = 0,986852
Arg =  2,500000 Value = 0,983628
Arg =  2,600000 Value = 0,970167
Arg =  2,700000 Value = 0,976503
Arg =  2,800000 Value = 0,997473
Arg =  2,900000 Value = 0,974298
Arg =  3,000000 Value = 0,970920
Arg =  3,100000 Value = 0,987341
```

```
Экспонента(символьный поток):
Exp =  1,000000          ReadTabExp =  1,000000
Exp =  2,718282          ReadTabExp =  2,718282
Exp =  7,389056          ReadTabExp =  7,389056
Exp =  20,085537         ReadTabExp =  20,085537
Exp =  54,598150         ReadTabExp =  54,598150
Exp =  148,413159        ReadTabExp =  148,413159
Exp =  403,428793        ReadTabExp =  403,428793
Exp =  1096,633158       ReadTabExp =  1096,633158
Exp =  2980,957987       ReadTabExp =  2980,957987
Exp =  8103,083928       ReadTabExp =  8103,083928
Exp =  22026,465795      ReadTabExp =  22026,465795


Логарифм(байтовый поток):
x=1,0: Ln = 0,000000     ReadTabLn = 0,000000
x=1,9: Ln = 0,641854     ReadTabLn = 0,641854
x=2,8: Ln = 1,029619     ReadTabLn = 1,029619
x=3,7: Ln = 1,308333     ReadTabLn = 1,308333
x=4,6: Ln = 1,526056     ReadTabLn = 1,526056
x=5,5: Ln = 1,704748     ReadTabLn = 1,704748
x=6,4: Ln = 1,856298     ReadTabLn = 1,856298
x=7,3: Ln = 1,987874     ReadTabLn = 1,987874
x=8,2: Ln = 2,104134     ReadTabLn = 2,104134
x=9,1: Ln = 2,208274     ReadTabLn = 2,208274
x=10,0: Ln = 2,302585    ReadTabLn = 2,302585


Логарифм от экспоненты:
LogExp =  1,000000       SerLogExp =  1,000000
LogExp =  2,000000       SerLogExp =  2,000000
LogExp =  3,000000       SerLogExp =  3,000000
LogExp =  4,000000       SerLogExp =  4,000000
LogExp =  5,000000       SerLogExp =  5,000000
LogExp =  6,000000       SerLogExp =  6,000000
LogExp =  7,000000       SerLogExp =  7,000000
LogExp =  8,000000       SerLogExp =  8,000000
LogExp =  9,000000       SerLogExp =  9,000000
LogExp =  10,000000      SerLogExp =  10,000000
```

Рисунок 1 - Финальный результат