

Лабораторная работа №6

Иванова Виктория 6203-010302D

## Задание 1

Разработала метод интегрирования по методу трапеций в классе Functions. Данный метод принимает в качестве параметров ссылку на объект функции, левую и правую границы интегрирования, а также шаг дискретизации. Особенностью реализации является проверка на выход интервала интегрирования за границы области определения функции с генерацией исключения в случае нарушения. Алгоритм метода трапеций заключается в разбиении области интегрирования на участки заданной длины с последующим вычислением площади под кривой как суммы площадей трапеций. Тестирование метода проводилось на примере функции экспоненты на отрезке [0,1]. Определила также, какой шаг дискретизации нужен, чтобы рассчитанное значение отличалось от теоретического в 7 знаке после запятой, 0.0001.

## Задание 2

Создала пакет threads, в котором размещены классы, связанные с потоками. Создала класс Task в пакете threads, инкапсулирующий параметры задания: ссылку на интегрируемую функцию, границы области интегрирования, шаг дискретизации и количество заданий. Реализовала в мэйне метод nonThread(), выполняющий последовательную обработку заданий в цикле. На каждой итерации генерируется логарифмическая функция со случайным основанием от 1 до 10, устанавливаются случайные границы интегрирования (левая от 0 до 100, правая от 100 до 200) и случайный шаг дискретизации от 0 до 1. После формирования параметров выводится сообщение "Source" с указанием значений границ и шага, затем вычисляется интеграл и выводится сообщение "Result" с результатом вычислений.

## Задание 3

В пакете threads создала два класса. Создала классы SimpleGenerator и SimpleIntegrator, реализующие интерфейс Runnable. Для решения проблем NullPointerException и несогласованности данных применены блоки синхронизации на объекте task и добавлена проверка на null в цикле ожидания данных. В классах реализована логика ожидания освобождения задания через метод wait(), ожидания появления данных для обработки с использованием notify() для координации работы потоков.

## Задание 4

В пакете threads создала два следующих класса. Создала классы Generator и Integrator, расширяющие класс Thread. Первоначальная реализация с одним семафором не обеспечивала строгого чередования операций, поэтому архитектура переработана с введением двух семафоров для раздельного управления доступом генератора и интегратора. Это позволило достичь четкой последовательности операций. В методе complicatedThreads() реализовала прерывание рабочих потоков после 50 миллисекунд выполнения.

Lab 4-2023 sources

```

77     public static void main(String[] args) throws IOException {
78
79         System.out.println("nonThread();");
80         nonThread();
81
82         //System.out.println("simpleThreads();");
83         //simpleThreads();
84
85         //System.out.println("complicatedThreads();");
86         //complicatedThreads();
87
88     }
89
90     0 ×
91
92     :
93
94
95
96
97
98
99
100

```

"C:\Program Files\Java\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.3\lib\idea\_rt.jar" -Dfile.encoding=UTF-8

интеграл экспоненты

Теоретическое значение 1.718281828459045

Результат с вычислениями через метод трапеции с шагом 0.0001 1.7182818298909435

Разница 1.4318983776462346E-9

Точность 1e-7

nonThread();

Source 78.3259928678357 115.8882152723404 0.805890319562504

Result 78.3259928678357 115.8882152723404 0.805890319562504 114.12705770100642

Source 62.16923107501694 136.5507210615512 0.9343045384164742

Result 62.16923107501694 136.5507210615512 0.9343045384164742 166.2106664315096

Source 36.95882274955673 134.6945611268885 0.9251026235636894

Result 36.95882274955673 134.6945611268885 0.9251026235636894 213.7946506088382

Source 84.5581324264119 180.97355073806418 0.7148739250214899

Result 84.5581324264119 180.97355073806418 0.7148739250214899 289.886849683264

Source 48.97885325654608 145.3894332931066 0.5869861690654138

Result 48.97885325654608 145.3894332931066 0.5869861690654138 191.34705365086367

Source 29.182695972496873 132.49039545953522 0.8997658290340319

Run 0 ×

:

```

93
94
95
96
97
98
99
100

```

Result 15.67463007382326 166.805153502611 0.9940185910176431 358.19052876632213

Source 70.71207096294579 197.3774564357987 0.6937797273150014

Result 70.71207096294579 197.3774564357987 0.6937797273150014 378.12400201970155

Source 43.29837963789676 130.14825831790995 0.5731171236298398

Result 43.29837963789676 130.14825831790995 0.5731171236298398 203.1721427711274

Source 76.64824731894977 134.73453968105258 0.4759481179034284

Result 76.64824731894977 134.73453968105258 0.4759481179034284 158.6603163290944

Source 69.90950685943368 141.58571629516874 0.4053108844822201

Result 69.90950685943368 141.58571629516874 0.4053108844822201 163.37327633816923

Source 65.80693591335479 131.50612730982755 0.9850616234492237

Result 65.80693591335479 131.50612730982755 0.9850616234492237 196.45585737931725

Source 79.04470391841456 146.87256868559797 0.09443181501514253

Result 79.04470391841456 146.87256868559797 0.09443181501514253 181.41143064752632

Source 89.44505128353002 125.01375871972633 0.11883555263980283

Result 89.44505128353002 125.01375871972633 0.11883555263980283 99.3496886649028

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** The left sidebar shows a tree view with nodes like `functions`, `.idea`, `basic`, and sub-nodes `Cos`, `Exp`, `Log`, and `Sin`.
- Code Editor:** The main window displays Java code. Lines 92-95 print "nonThread()", "simpleThreads()", and "simpleThreads()". Lines 96-99 print "complicatedThreads()" and "complicatedThreads()". A yellow lightbulb icon is shown next to line 96.
- Output Window:** Below the editor, the terminal output shows:
  - "C:\Program Files\Java\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition интеграл экспоненты"
  - Теоретическое значение 1.718281828459045
  - Результат с вычислениями через метод трапеции с шагом 0.0001 1.7182818298909435
  - Разница 1.4318983776462346E-9
  - Точность 1e-7
- Text Output:** Below the terminal, there is a block of text output from the `simpleThreads()` method, which includes:
  - Source values: 93.29126207145448, 150.50147084077963, 0.027003153428532434
  - Result values: 93.29126207145448, 150.50147084077963, 0.027003153428532434, 138.7846584877329
  - Source values: 75.24621461252237, 114.15899817775217, 0.04746023562952972
  - Result values: 75.24621461252237, 114.15899817775217, 0.04746023562952972, 127.08182272943223
  - Source values: 74.2642762021783, 115.19703767416377, 0.8333292660211357
  - Result values: 74.2642762021783, 115.19703767416377, 0.8333292660211357, 91.51593529216645
  - Source values: 12.069297631890585, 143.28629402884775, 0.48250089449803046
  - Result values: 12.069297631890585, 143.28629402884775, 0.48250089449803046, 303.3959988061266
  - Source values: 70.37063120717238, 179.7559926722994, 0.5797527940891325
  - Result values: 70.37063120717238, 179.7559926722994, 0.5797527940891325, 268.33984721339067
  - Source values: 31.59944827479294, 155.12940555237282, 0.8895222220403737

```
functions 92
          //System.out.println("nonThread();");
          //nonThread();
          System.out.println("simpleThreads();");
          simpleThreads();
          //System.out.println("complicatedThreads();");
          //complicatedThreads();

0 ×

Result 96.84541213213784 110.16098486032809 0.6942558519569598 191.23316304325448
Source 1.5733996005536377 194.8070756943457 0.07595888565436015
Result 1.5733996005536377 194.8070756943457 0.07595888565436015 539.3358012798486
Source 16.90197668608494 141.20378040922228 0.15819649135374825
Result 16.90197668608494 141.20378040922228 0.15819649135374825 447.7419590037659
Source 54.71821606909434 196.6873967811128 0.1160296551195017
Result 54.71821606909434 196.6873967811128 0.1160296551195017 517.8998333787189
Source 69.09998896856335 102.33365421145132 0.3175362545298608
Result 69.09998896856335 102.33365421145132 0.3175362545298608 95.52517758344155
Source 13.821664764458152 164.41407758803138 0.3234063611230944
Result 13.821664764458152 164.41407758803138 0.3234063611230944 325.01862418319087
Source 78.27406679285723 199.82706138493182 0.3124187061288488
Result 78.27406679285723 199.82706138493182 0.3124187061288488 291.9813246411513
Source 65.9130603693635 120.32602542196521 0.44943080098046373
Result 65.9130603693635 120.32602542196521 0.44943080098046373 143.588713437545

Process finished with exit code 0
```

The screenshot shows the IntelliJ IDEA interface. On the left, the project structure is visible with a tree view containing 'functions', '.idea', 'basic', and several source files: 'Cos.java', 'Exp.java', 'Log.java', and 'Sin.java'. The code editor on the right contains Java code for thread management and printing results. Below the editor is a terminal window showing the execution of a Java application with a Java agent. The terminal output displays the integral of an exponential function, the theoretical value, the result from the trapezoidal rule, the difference, the tolerance, and the output of the 'complicatedThreads()' method, which prints multiple sets of numerical values.

```
"C:\Program Files\Java\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2017.3.3\lib\idea_rt.jar" -Dfile.encoding=UTF-8
интеграл экспоненты
Теоретическое значение 1.718281828459045
Результат с вычислениями через метод трапеции с шагом 0.0001 1.7182818298909435
Разница 1.4318983776462346E-9
Точность 1e-7
complicatedThreads()
Source 22.430390678795796 111.58295208154723 0.3241661848101106
Result 22.430390678795796 111.58295208154723 0.3241661848101106 719.2468314642829
Source 54.33841769696731 158.63925991123546 0.8086653747696793
Result 54.33841769696731 158.63925991123546 0.8086653747696793 236.69838301099736
Source 53.67795180348421 171.82905314849216 0.7278330558168359
Result 53.67795180348421 171.82905314849216 0.7278330558168359 412.41991890120903
Source 45.9459444099228 198.59200350277393 0.7861993854675838
Result 45.9459444099228 198.59200350277393 0.7861993854675838 332.65613359290836
Source 26.267453802014128 128.66315992943288 0.4283331782420159
Result 26.267453802014128 128.66315992943288 0.4283331782420159 736.9305947128906
```

The screenshot shows a file tree on the left and a terminal window on the right.

**File Tree:**

- > .idea
- functions
- > .idea
- basic
  - (C) Cos
  - (C) Exp
  - (C) Log
  - (C) Sin

**Terminal Output:**

```
Result 30.863190160126663 167.39822556299265 0.44069608809113836 308.3623173701697
Source 43.17877400713942 110.43221309530162 0.8751453889941831
Result 43.17877400713942 110.43221309530162 0.8751453889941831 684.9324186850968
Source 48.02121708647782 151.57084002994347 0.39984381639612065
Result 48.02121708647782 151.57084002994347 0.39984381639612065 498.70183497242016
Source 36.61907702015402 173.13078669472873 0.3555632266710309
Result 36.61907702015402 173.13078669472873 0.3555632266710309 1339.2871229128393
Source 11.877164422777808 183.06054292203441 0.9891534719248024
Result 11.877164422777808 183.06054292203441 0.9891534719248024 655.7193618495377
Source 12.76625403649765 132.23851005930436 0.13555010273611312
Result 12.76625403649765 132.23851005930436 0.13555010273611312 231.39047355498815
Source 50.3469365125551 136.8050836448929 0.05507962713695802
Result 50.3469365125551 136.8050836448929 0.05507962713695802 282.2292225034598
Source 65.78225055352166 100.57568889454531 0.42853371048191935
Result 65.78225055352166 100.57568889454531 0.42853371048191935 144.29171425463525
```