

# А. Задания к лабораторным работам

Алексей Мартынов

Василий Касилов

26 августа 2024 г.

Версия 1.0

## 1. Стандартные потоки

Реализовать подсчет площади кругов заданных радиусов методом Монте-Карло. Подсчет площади должен выполняться в пределах одного процесса с использованием заданного числа потоков выполнения. Работа должна быть выполнена в виде 1 исполняемого файла, принимающего параметры следующим образом:

```
$ ./lab tries [seed]
```

`ties` — количество испытаний в методе Монте-Карло. Опциональный параметр `seed` задает инициализирующее значения для генератора псевдослучайных чисел, используемого в расчете. Если параметр не задан, то должно использоваться значение 0.

Со стандартного ввода программа должна принимать пары целых положительных чисел: радиус круга, площадь которого нужно рассчитать методом Монте-Карло и количество потоков выполнения, используемых в расчете площади для этого круга. Расчет площади выполняется отдельно для каждой пары сразу после ввода. Признаком завершения ввода является EOF.

Для каждой введенной пары программа должна выводить на отдельных строках пары вещественных чисел через пробел с точностью до трех знаков: время расчета в миллисекундах и площадь круга соответствующего радиуса. Например:

```
$ cat data
1 6
2 12
$ ./lab 100000000 < data
327.027 3.145
184.423 12.582
```

Программа должна завершаться с ненулевым кодом возврата и сообщением в потоке ошибок, если:

1. количество аргументов командной строки не соответствует требованиям,
2. количество испытаний не положительные,
3. инициализирующее значение генератора отрицательное,
4. введенный радиус или число потоков не положительные.

Во всех остальных случаях программа должна завершаться с кодом возврата 0.

## 2. Процессы I (POSIX)

Реализовать интерактивную программу, принимающую команды от пользователя и выполняющую расчет площади покрытия методом Монте-Карло пересекающихся кругов заданных радиусов. Взаимодействие с пользователем и расчет площади должны выполняться в двух разных процессах. Для создания процессов и их взаимодействия должны быть использованы средства, описанные в стандарте POSIX. Работа должна быть выполнена в виде 1 исполняемого файла, принимающего опциональный параметр командой строки - значение для инициализации генератора псевдослучайных чисел (считать 0, если не задано).

Задача основного процесса - обрабатывать команды, вводимые со стандартного ввода, отправлять запросы на расчет дочернему процессу и сохранять полученные результаты расчетов. Все расчеты должны выполняться в одном дочернем процессе.

Программа должна поддерживать следующие операции:

- Создание круга заданного радиуса с указанными координатами.
- Создание набора из созданных фигур.
- Просмотр созданных фигур и наборов.
- Вычисление ограничивающего прямоугольника заданного набора кругов или отдельного круга.
- Выполнение запроса на расчет площади покрытия указанного набора.
- Просмотр результатов завершенных расчетов.
- Ожидание завершения указанного расчета.

Пользовательский интерфейс должен поддерживать следующие команды:

- **circle** <name> <r> <x> <y>

Создание круга с именем <name> радиуса <r> с координатами <x> и <y>. Команда считается ошибочной, если круг с данным именем уже существует или параметры некорректны.

- **set** <name> <set-size> <circle-1> ... <circle-n>

Создание набора с именем <name> из <set-size> кругов с указанными именами <circle-1> ... <circle-n>. Команда считается ошибочной, если набор с заданным именем уже существует, некорректно задан размер набора или хотя бы один круг, входящий в набор, не с помощью команды **circle**.

- **show** <circle-name>

Вывод параметров круга с именем **name** на отдельной строке в формате <radius> (<x> <y>). Например для ввода:

```
circle c1 5 3 4
show c1
```

Ожидается вывод:

```
5 (3 4)
```

Команда считается ошибочной, если круга с заданным именем не существует.

- **showset** <set-name>

Вывод каждого элемента набора с именем <set-name> на отдельной строке. Формат вывода соответствует выводу команды **show**. Команда считается ошибочной, если набора с заданным именем не существует.

- **frame** <circle-name>

Вывод ограничивающего прямоугольника для круга с именем <circle-name>. Например для ввода:

```
circle c2 3 0 0
frame c2
```

Ожидается вывод:

```
(-3 -3) (3 3)
```

Команда считается ошибочной, если круга с заданным именем не существует.

- **frameset** <set-name>

Вывод ограничивающего прямоугольника для набора с именем <set-name>. Формат вывода аналогичен команде **frame**. Команда считается ошибочной, если набора с заданным именем не существует.

- **area** <calc-name> <set-name> <threads> <tries>

Запуск расчета площади покрытия набором <set-name> с заданным числом потоков <threads> при количестве испытаний в методе Монте-Карло <tries>. <calc-name> является именем расчета, по которому можно получить результаты расчета. Команда считается ошибочной, если расчет с заданным именем уже запускался или набор для расчета не существует.

- **status** <calc-name>

Вывод статуса расчета с именем <calc-name>. Для завершенного расчета должна выводиться площадь покрытия с точностью до трёх знаков. Если расчет не завершен, должно быть выведено сообщение <IN PROGRESS>. Команда считается ошибочной, если заданный расчет не запускался.

- **wait** <calc-name>

Ожидание завершения расчета с именем <calc-name>. Должны быть выведены результаты расчета - площадь покрытия набора. Команда считается ошибочной, если заданный расчет не запускался.

Каждая команда располагается на отдельной строке. Радиус кругов, координаты, количество испытаний и число потоков в методе Монте-Карло задаются целыми числами. Имена кругов, наборов и расчетов задаются строкой без пробелов, начинающейся с букв английского алфавита. Ввод команд должен завершаться EOF. Программа должна завершаться нулевым кодом возврата.

3. Процессы II (Windows)
4. Структуры данных с блокировками
5. Структуры данных свободные от блокировок
6. Асинхронное программирование I (Future и Promise)