|  |
| --- |
| **ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**  **НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**  **«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**  Факультет компьютерных наук  Департамент программной инженерии  **ЗАДАЧА О ЧИТАТЕЛЯХ И ПИСАТЕЛЯХ**  Пояснительная записка  Исполнитель:  Студентка группы БПИ199  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Точилина П. В./  «13» декабря 2020 г. |

**Содержание**

[1.Текст задания 2](#_Toc58785152)

[1.1 Текст задания 2](#_Toc58785153)

[1.2 Пояснение к тексту задания 2](#_Toc58785154)

[2. Применяемые расчетные методы 3](#_Toc58785155)

[2.1 Теория решения задания 3](#_Toc58785156)

[2.2 Ограничения на ввод 3](#_Toc58785157)

[2.3 Дополнительные функционалы программы 3](#_Toc58785158)

[3. Тестирование программы 4](#_Toc58785159)

[3.1 Приветствие пользователя 4](#_Toc58785160)

[3.2 Корректные данные 4](#_Toc58785161)

[3.3 Некорректные данные 6](#_Toc58785162)

[4. Список источников. 7](#_Toc58785163)

1.Текст задания

1.1 Текст задания

Базу данных разделяют два типа процессов – читатели и писатели. Читатели выполняют транзакции, которые просматривают записи базы данных, транзакции писателей и просматривают и изменяют записи. Предполагается, что в начале БД находится в непротиворечивом состоянии (т.е. отношения между данными имеют смысл). Каждая отдельная транзакция переводит БД из одного непротиворечивого состояния в другое. Для предотвращения взаимного влияния транзакций процесс-писатель должен иметь исключительный доступ к БД. Если к БД не обращается ни один из процессов-писателей, то выполнять транзакции могут одновременно сколько угодно читателей. Создать многопоточное приложение с потоками-писателями и потоками-читателями. Реализовать решение, используя семафоры.

1.2 Пояснение к тексту задания

Для более удобного представления программы, условие реализовано на ситуации буквальных писателей и читателей некоторых книг.

2. Применяемые расчетные методы

2.1 Теория решения задания

Разделим потоки на три типа:

1. Основной поток. Создает заданное число потоков-читателей и потоков-писателей, после чего ожидает их присоединения и завершает программу.
2. Поток-читатель. Заходит в цикл с выходом при завершении написания всех книг потоками-писателями. Если до этого на данном участке не было других потоков-читателей, то поток блокирует доступ потокам-писателям к базе данных. Поток читает некоторую запись в базе данных. Если на данном участке не осталось потоков-читателей, то поток открывает доступ потокам-писателям к базе данных. Далее поток симулирует выполнение задания (спит).
3. Поток-писатель. Заходит в цикл с выходом при условии, что начато книг столько, сколько должно быть завершено (чтобы не написать лишних книг). Поток засыпает, симулируя написание книги после чего добавляет новую запись в базу данных.

Доступ к базе данных, выводу и некоторым переменным ограничивается семафорами.

2.2 Ограничения на ввод

Количество книг, потоков-писателей и потоков-читателей должно быть целым положительным числом.

2.3 Дополнительные функционалы программы

Скорость выполнения задания потоками-писателями и потоками-читателями генерируется рандомно в отрезке [3000, 5000].

3. Тестирование программы

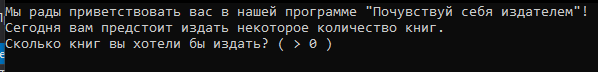
3.1 Приветствие пользователя

Рисунок . Приветствие

3.2 Корректные данные

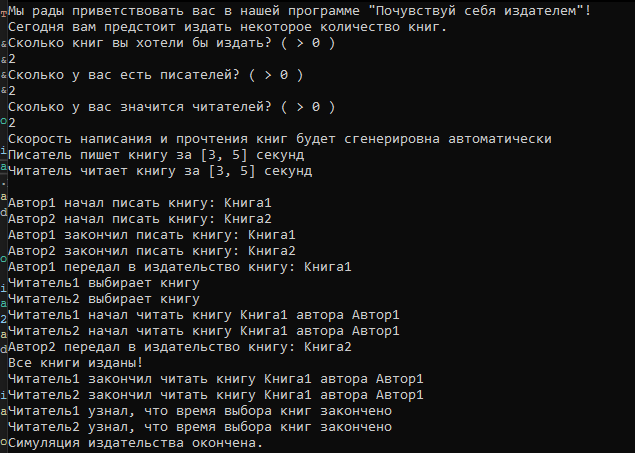
****Протестируем программу при вводе одинакового количества писателей и книг

Рисунок . Тест 1

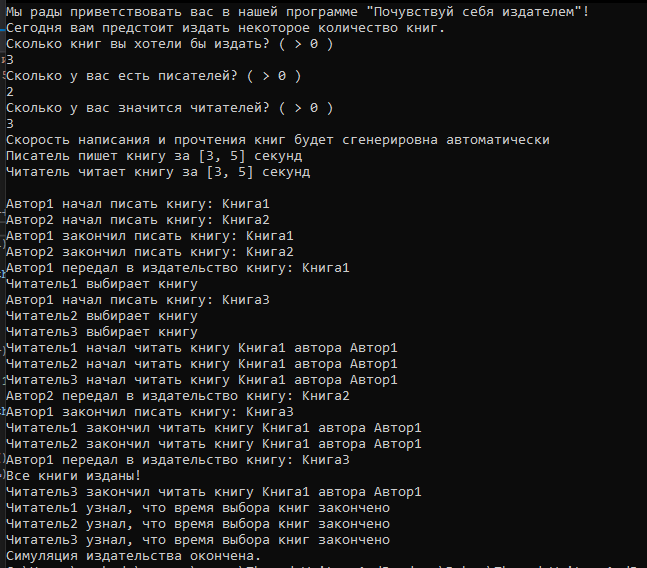
****Протестируем программу на стандартных данных

Рисунок . Тест 2

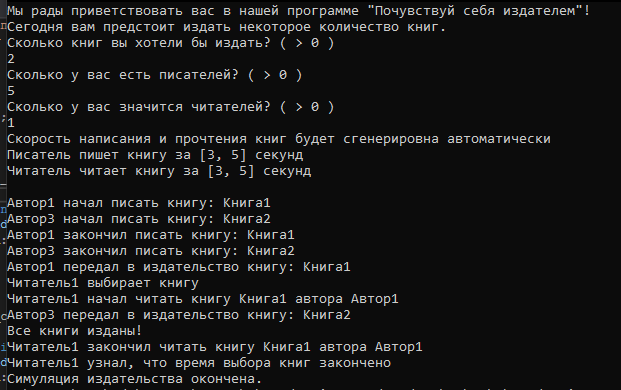
****Протестируем программу при вводе писателей больше чем книг

Рисунок . Тест 3

3.3 Некорректные данные

Проверим на вводе некорректных значений int.

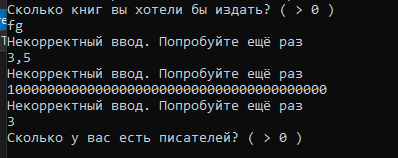


Рисунок . Тест 4

4. Список источников.

1. Требования к оформлению [Электронный ресурс].

Режим доступа: [http://softcraft.ru/edu/comparch/tasks/mp02/](https://vk.com/away.php?to=http%3A%2F%2Fsoftcraft.ru%2Fedu%2Fcomparch%2Ftasks%2Fmp02%2F&cc_key=), свободный.

1. Модель читатели-писатели [Электронный ресурс].

Режим доступа: <https://ru.wikipedia.org/wiki/Задача_о_читателях-писателях>, свободный.