

# Содержание

<b>1</b>	<b>Модуль DataManipulator</b>	<b>2</b>
1.1	Назначение модуля DataManipulator	2
1.2	Главное меню модуля DataManipulator.	2
1.3	Меню обработки данных OCDF-формата модуля DataManipulator.	2
1.3.1	Считывание OCDF-данных из файла.	2
1.3.2	Возможности по обработке OCDF-данных.	4
1.4	Обезка OCDF-данных.	5
1.4.1	Обрезка OCDF-данных по заданному проценту.	5
1.4.2	Обрезка OCDF-данных по заданному количеству.	6
1.4.3	Обрезка OCDF-данных по заданному моменту времени.	7
1.4.4	Возможные ошибки при обрезке OCDF-данных.	7
1.5	Парсинг OCDF-данных.	9
1.5.1	Процесс парсинга OCDF-данных.	9
1.5.2	Возможные ошибки при парсинге OCDF-данных.	9
1.6	Выравнивание диапазонов OCDF-данных.	10
1.6.1	Математические основы выравнивания диапазонов OCDF-данных.	11
1.6.2	Процесс выравнивания диапазонов OCDF-данных.	11
1.6.3	Возможные ошибки при выравнивании диапазонов OCDF-данных.	12
1.7	Сохранение OCDF-данных в файл формата csv.	12
1.8	Сохранение OCDF-данных в бинарный файл.	12
1.9	Выход из меню работы с OCDF-данными.	12

# 1 Модуль DataManipulator

## 1.1 Назначение модуля DataManipulator

Модуль DataManipulator предназначен для различного рода обработки данных, с целью получения наборов данных (датасетов) необходимых для обучения моделей нейронных сетей. DataManipulator помогает преодолеть этап анализа и обработки данных, а также этап конструирования признаков.

## 1.2 Главное меню модуля DataManipulator.

В главном меню модуля DataManipulator представлены основные направления для обработки данных. Предоставляется выбор, с каким форматом данных будет происходить работа: с OCDF- или TDF-форматом. В главном меню также можно выбрать пункт, позволяющий создать TDF-данные на основе OCDF-данных. Также имеется пункт, который выводит на экран краткую информацию о форматах данных и о самом модуле DataManipulator. Внешний вид меню можно увидеть на рисунке 1

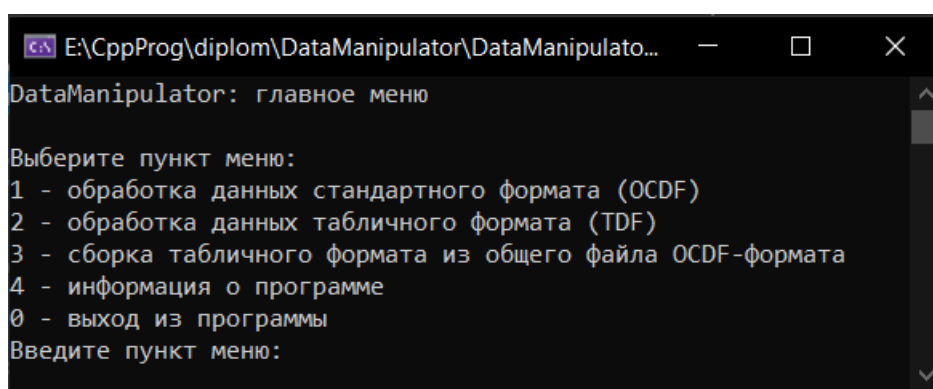


Рис. 1: главное меню модуля DataManipulator

## 1.3 Меню обработки данных OCDF-формата модуля DataManipulator.

### 1.3.1 Считывание OCDF-данных из файла.

Для того, чтобы попасть в меню по обработке данных OCDF-формата будет предложено для начала их загрузить из файла. Это показано на рисунке на рисунке 2.

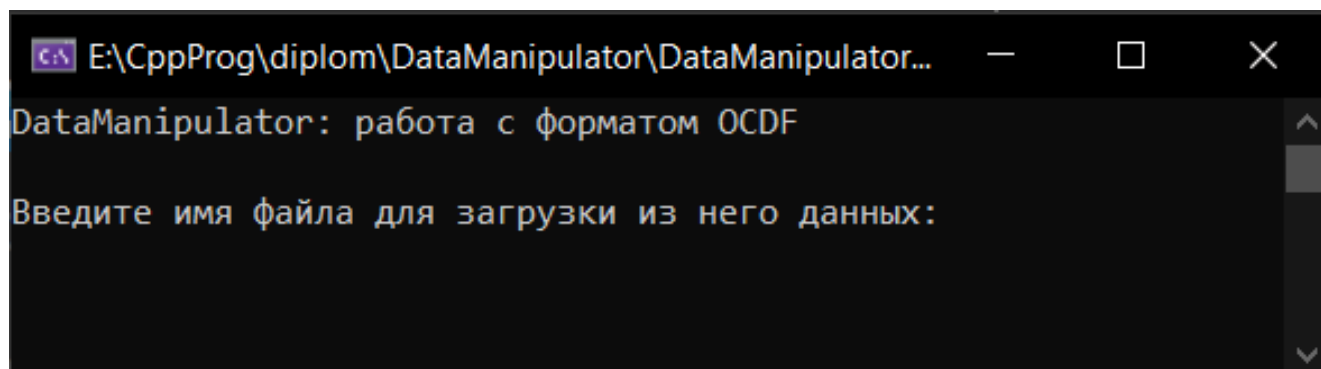
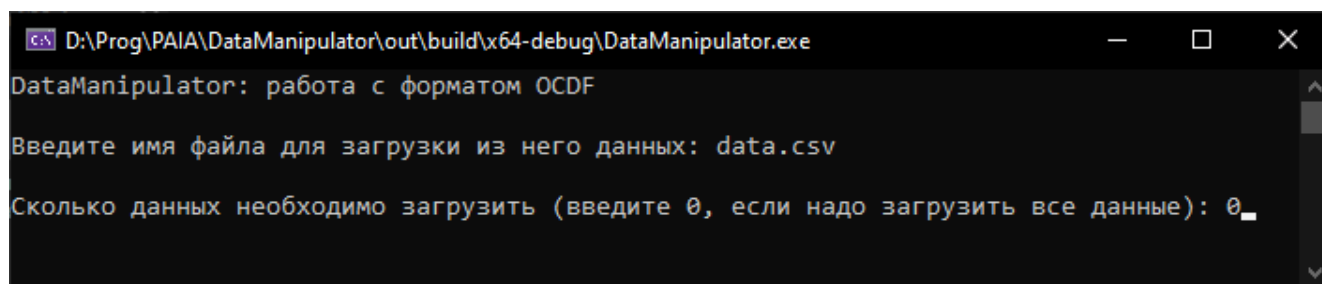


Рис. 2: указание файла для загрузки из него OCDF-данных.

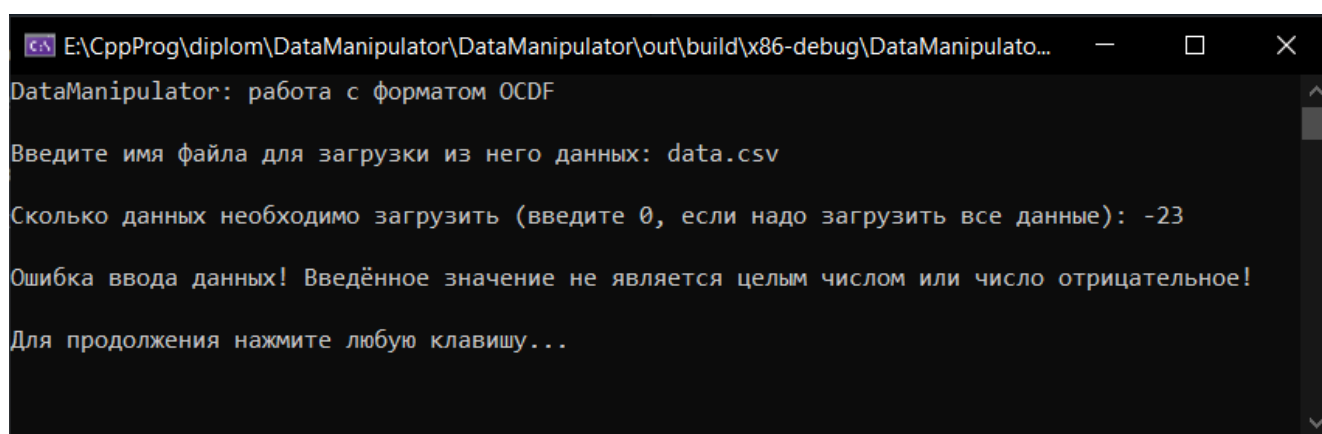
Вводим путь к файлу или перетаскиваем файл в консоль для получения его абсолютного пути и нажимаем Enter. После этого будет предложено ввести количество данных, которые необходимо считать из файла. Это показано на рисунке 3.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: работа с форматом OCDF
Введите имя файла для загрузки из него данных: data.csv
Сколько данных необходимо загрузить (введите 0, если надо загрузить все данные): 0
```

Рис. 3: указание необходимого количества данных для считывания.

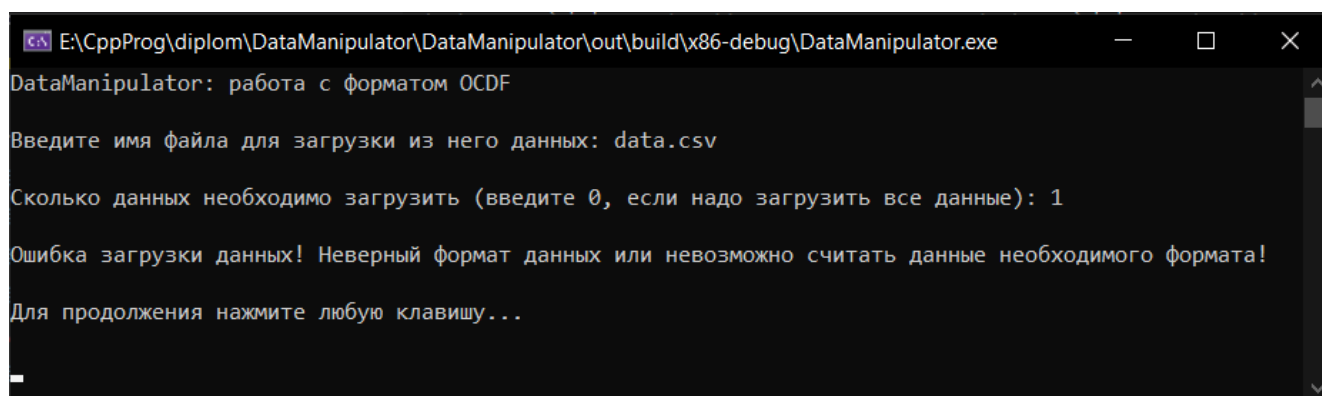
Есть несколько особенностей, при вводе количества данных. В случае, если будет введён 0 или число, превышающее количество данных в файле, то будут считаны все данные, которые есть в файле. Если будет введено отрицательное число или хотя бы один символ, отличный от цифр, то программа выдаст предупреждение о некорректном вводе данных, показанное на рисунке 4, и предложит ввести путь к файлу и необходимое количество данных для считывания снова.



```
E:\CppProg\diplom\DataManipulator\DataManipulator\out\build\x86-debug\DataManipulator.exe
DataManipulator: работа с форматом OCDF
Введите имя файла для загрузки из него данных: data.csv
Сколько данных необходимо загрузить (введите 0, если надо загрузить все данные): -23
Ошибка ввода данных! Введённое значение не является целым числом или число отрицательное!
Для продолжения нажмите любую клавишу...
```

Рис. 4: предупреждение об ошибке при вводе необходимого количества данных для считывания.

Могут возникнуть и другие предупреждения, связанные непосредственно с самим файлом. Если файл с данными не существует, или этот файл пуст, или данные в нём не представлены в необходимом формате, то возникнет предупреждение о невозможности чтения данных, представленное на рисунке 5.



```
E:\CppProg\diplom\DataManipulator\DataManipulator\out\build\x86-debug\DataManipulator.exe
DataManipulator: работа с форматом OCDF
Введите имя файла для загрузки из него данных: data.csv
Сколько данных необходимо загрузить (введите 0, если надо загрузить все данные): 1
Ошибка загрузки данных! Неверный формат данных или невозможно считать данные необходимого формата!
Для продолжения нажмите любую клавишу...
```

Рис. 5: предупреждение об ошибке чтения данных.

Как уже понятно, чтение данных происходит в текстовом режиме. Однако существует возможность считывать данные, которые представлены в бинарном виде. Для этого необ-

ходимо, чтобы файл имел расширение .bin, иначе чтение данных будет происходить в текстовом режиме. Пример файла с данными в бинарном виде можно увидеть на рисунке 6.

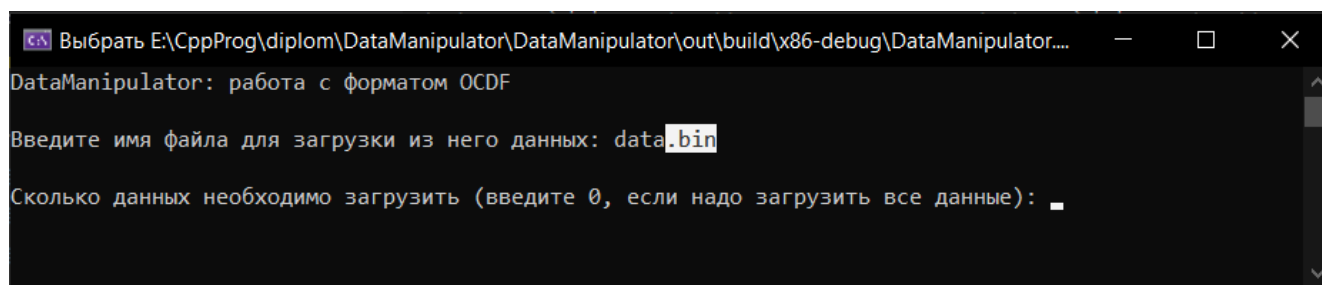


Рис. 6: ввод файла с данными в бинарном виде.

После указания файла с данными в бинарном виде будет предложено ввести количество данных, которое необходимо считать из файла. Все предупреждения, которые связаны с чтением данных и ввода значений уже разобраны выше.

### 1.3.2 Возможности по обработке OCDF-данных.

После считывания OCDF-данных из файла открывается меню для работы с данными, представленное на рисунке 7.

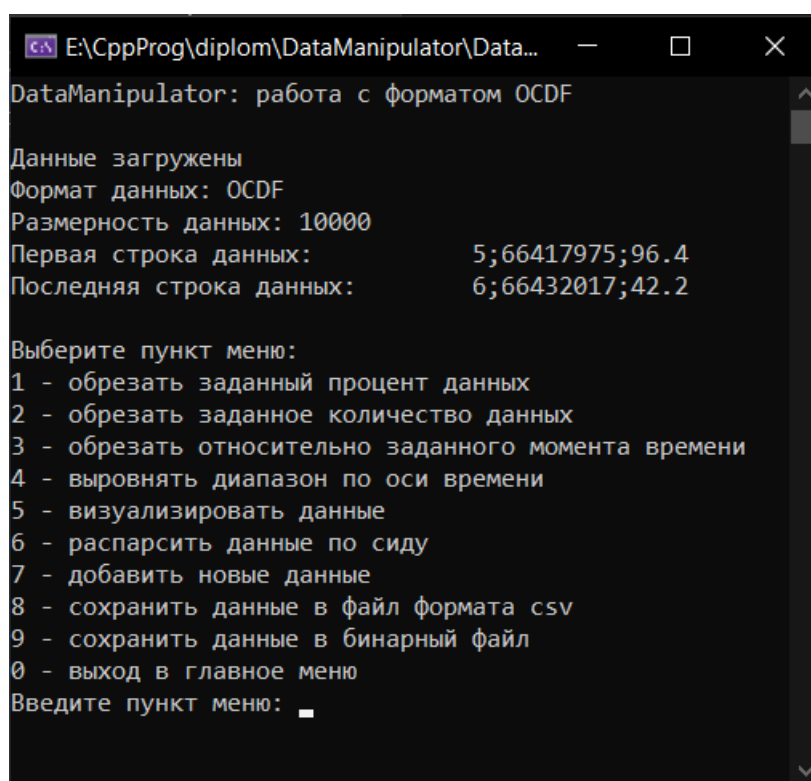


Рис. 7: ввод файла с данными в бинарном виде.

В появившемся меню будет указана информация о считанных данных: формат данных, их количество, а также первая и последняя строки последовательности данных. После идёт выбор действий, которые можно совершить над данными: обрезка данных, визуализация данных, приведение данных к равноинтервальному виду, парсинг данных, добавление данных и сохранение результатов обработки. Каждое из них подробно будет разобрано ниже.

## 1.4 Обрезка OCDF-данных.

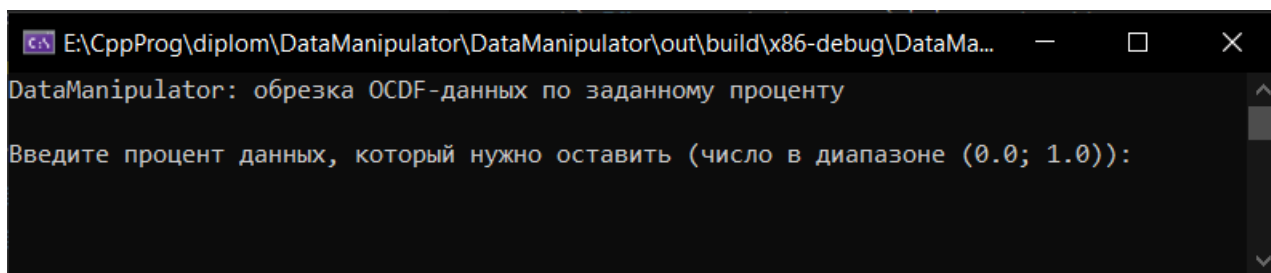
Существует 3 вида обрезки данных: обрезка заданного процента данных, обрезка заданного количества данных и обрезка по заданному моменту времени. Также у каждого вида обрезки данных существует 2 режима обрезки данных: обрезка данных слева направо или обрезка данных справа налево. Рассмотрим каждый вид обрезки данных на примерах данных, показанных на рисунке 8.

```
Данные загружены
Формат данных: OCDF
Размерность данных: 69870
Первая строка данных:      1;1256;0
Последняя строка данных:   1;10612923;100
```

Рис. 8: пример данных для дальнейшей их обрезки.

### 1.4.1 Обрезка OCDF-данных по заданному проценту.

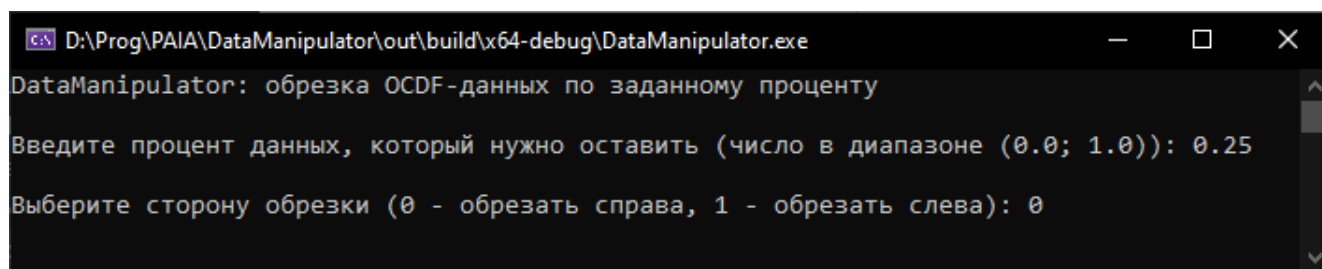
Обрезка по заданному проценту данных оставляет введённое пользователем процентное количество данных от общего их количества. Для ввода процента обрезки данных необходимо задать число в промежутке (0,00; 1,00), где 0,00 - 0%, а 1,00 - 100%, как это показано на рисунке 9.



```
E:\CppProg\diplom\DataManipulator\DataManipulator\out\build\x86-debug\DataMa...
DataManipulator: обрезка OCDF-данных по заданному проценту
Введите процент данных, который нужно оставить (число в диапазоне (0.0; 1.0)):
```

Рис. 9: ввод процента для обрезки данных.

После ввода процента для обрезки данных, программа предлагает выбрать режим обрезки, введя значения 0 или 1, как показано на рисунке 10.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: обрезка OCDF-данных по заданному проценту
Введите процент данных, который нужно оставить (число в диапазоне (0.0; 1.0)): 0.25
Выберите сторону обрезки (0 - обрезать справа, 1 - обрезать слева): 0
```

Рис. 10: ввод режима для обрезки по заданному проценту.

Если ввести 0, то заданный процент данных будет оставлен слева, т.е. в начале. Если ввести 1, то заданный процент данных будет оставлен справа, т.е. в конце. Вводим режим обрезки данных 0 и смотрим результаты на рисунке 11.

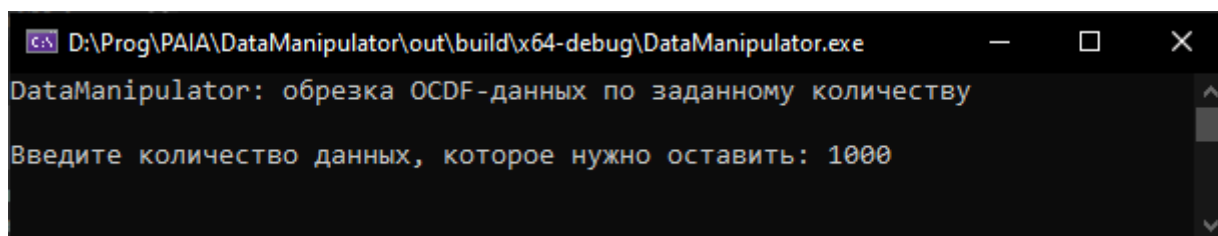
```
Данные загружены
Формат данных: OCDF
Размерность данных: 17467
Первая строка данных:      1;1256;0
Последняя строка данных:   1;1981280;79.7
```

Рис. 11: результаты обрезки данных по заданному проценту.

Была произведена обрезка 75% данных от общего количества, поскольку было введён 25% данных, которые необходимо оставить, в режиме справо налево, т.е. с конца.

#### 1.4.2 Обрезка OCDF-данных по заданному количеству.

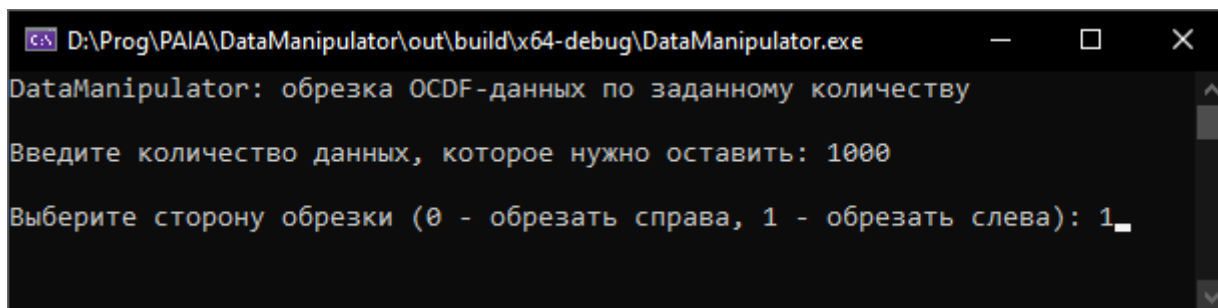
Обрезка по заданному количеству данных оставляет введённое пользователем количество данных. Для ввода количества обрезки данных необходимо задать целое положительное число отличное от нуля, как это показано на рисунке 12.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: обрезка OCDF-данных по заданному количеству
Введите количество данных, которое нужно оставить: 1000
```

Рис. 12: ввод значения количества для обрезки данных.

После ввода количества для обрезки данных, программа предлагает выбрать режим обрезки, введя значения 0 или 1, как показано на рисунке 13.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: обрезка OCDF-данных по заданному количеству
Введите количество данных, которое нужно оставить: 1000
Выберите сторону обрезки (0 - обрезать справа, 1 - обрезать слева): 1_
```

Рис. 13: ввод режима для обрезки по заданному количеству.

Если ввести 0, то заданное количество данных будет оставлено слева, т.е. в начале. Если ввести 1, то заданное количество данных будет оставлено справа, т.е. в конце. Вводим режим обрезки данных 1 и смотрим результаты на рисунке 14.

```
Данные загружены
Формат данных: OCDF
Размерность данных: 1000
Первая строка данных:      1;1709916;98.8
Последняя строка данных:   1;1981280;79.7
```

Рис. 14: результаты обрезки данных по заданному количеству.

Была произведена обрезка 16467 данных, поскольку было введено 1000 данных, которые необходимо оставить, в режиме справо налево, т.е. в конце.

### 1.4.3 Обрезка OCDF-данных по заданному моменту времени.

Обрезка по заданному моменту времени оставляет промежуток данных до или после указанного момента. Для ввода момента времени для обрезки данных необходимо задать целое положительное число отличное от нуля, как это показано на рисунке 15.

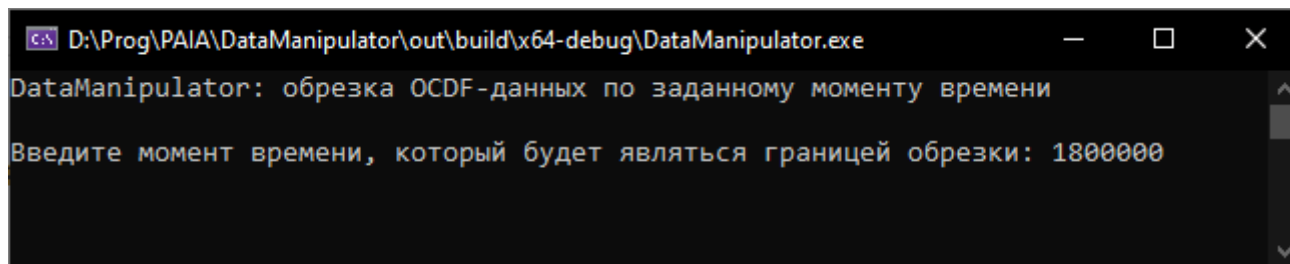


Рис. 15: ввод значения момента времени для обрезки данных.

После ввода момента времени для обрезки данных, программа предлагает выбрать режим обрезки, введя значения 0 или 1, как показано на рисунке 16.

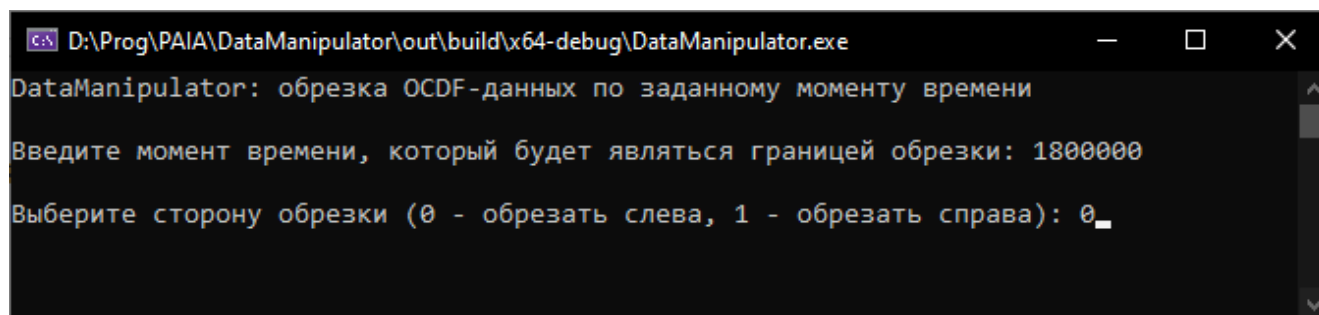


Рис. 16: ввод режима для обрезки по заданному моменту времени.

Если ввести 0, то данные будут оставлены слева относительно заданного момента времени, т.е. в начале. Если ввести 1, то данные будут оставлены справа относительно заданного момента времени, т.е. в конце. Вводим режим обрезки данных 0 и смотрим результаты на рисунке 17.

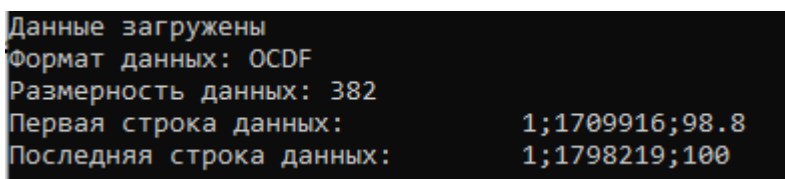


Рис. 17: результаты обрезки данных по заданному моменту времени.

### 1.4.4 Возможные ошибки при обрезке OCDF-данных.

При обрезке данных могут возникать различные ситуации, которые способны вызывать всяческие ошибки или исключительные случаи. Самым основным, что может вызвать ошибку, это некорректный ввод данных. Так, например, при обрезке данных по заданному проценту не допустимо вводить числа не принадлежащие диапазону (0,00; 1,00), хоть и ввод 1,00 доступен, однако при этом данные никак не изменятся. Ввод отрицательных или не целых чисел при обрезке по заданному моменту времени и по заданному количеству данных также вызывает ошибки некорректного ввода. Пример такой ошибки продемонстрирован на рисунке 18.



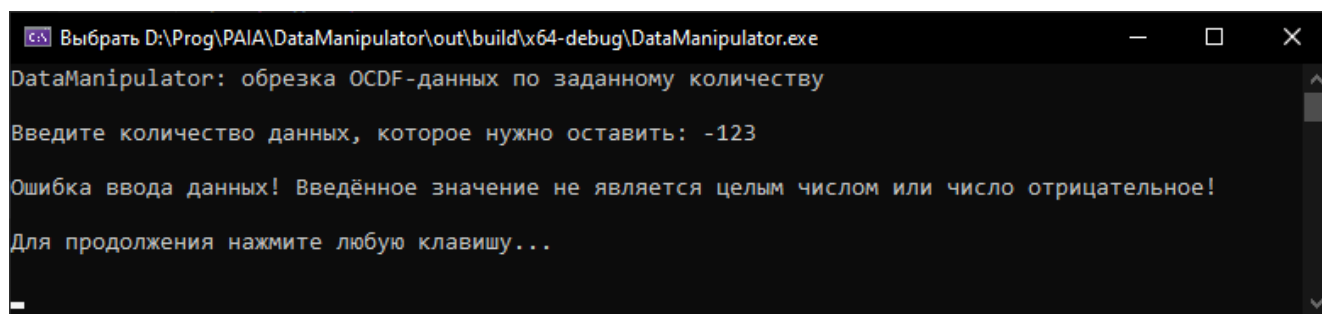


Рис. 18: пример ввода недопустимых чисел при обрезке OCDF-данных.

Также может быть вызвана ошибка некорректного ввода, при использовании символов, отличных от цифр, характерное для всех видов обрезки. Пример такой ошибки продемонстрирован на рисунке 19.

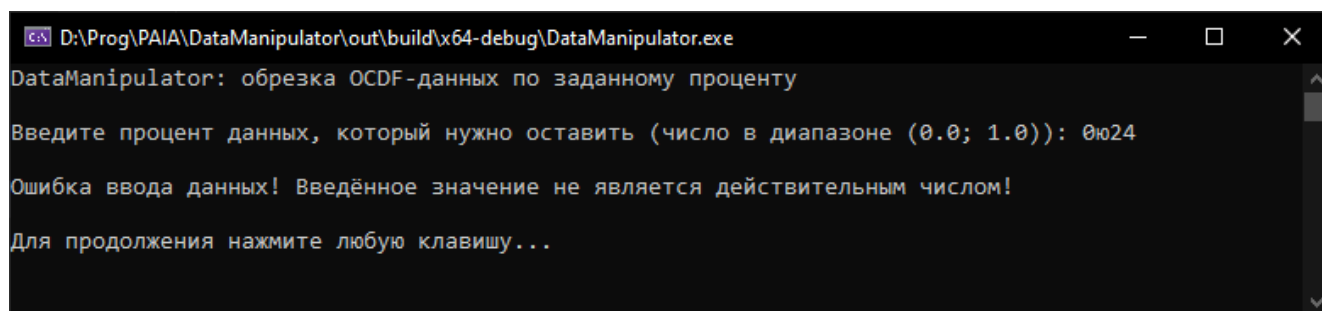


Рис. 19: пример ввода недопустимых символов при обрезке OCDF-данных.

При обрезке данных по заданному количеству есть вероятность того, что пользователь введёт число больше общего количества данных. В таком случае данные не изменятся и ошибка не появится.

При обрезке данных по заданному моменту времени есть вероятность того, что пользователь введёт момент времени, который находится раньше начального момента времени или позже конечного момента времени в самих данных. В таком случае, если будет выбран соответствующий режим обрезки, данные либо никак не изменятся, либо появится ошибка о возврате пустых данных, что продемонстрирована на рисунке 20.

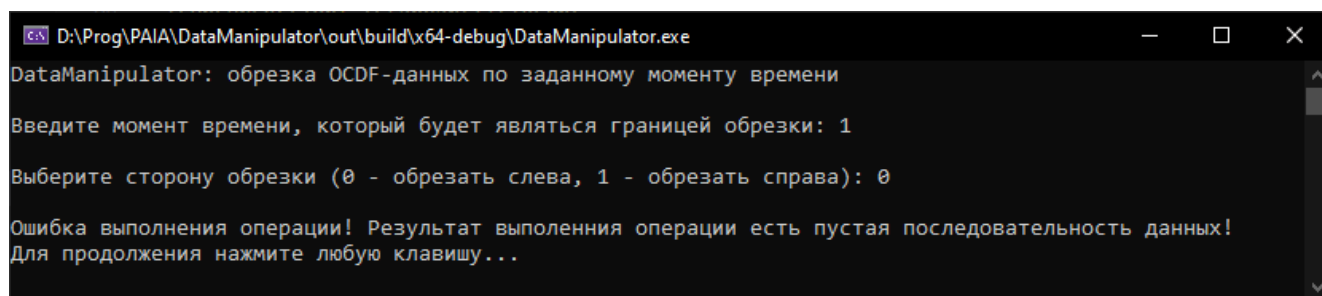


Рис. 20: пример ошибки при возврате пустых OCDF-данных.

Такая ошибка может возникать также при вводе 0,00 для обрезки данных по заданному проценту. В случае такой ошибки данные никак не изменятся и управление перейдёт к меню обработки OCDF-данных.

Последняя возможная ошибка относится к вводу недопустимых символов при выборе режимов обрезки. В таком случае возникает ошибка, которая перезапускает операцию заново. Пример такой ошибки продемонстрирован на рисунке 21.



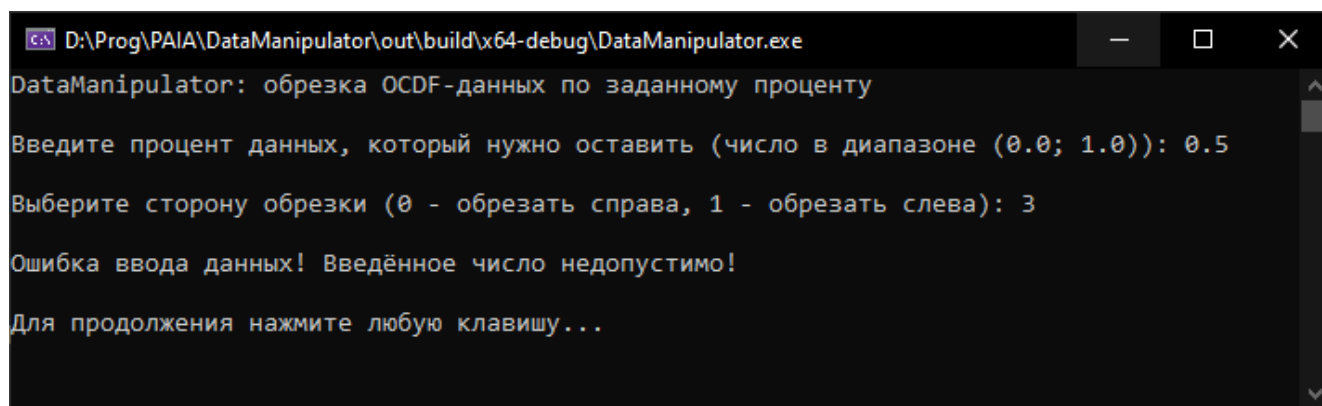


Рис. 21: пример ввода некорректных данных при выборе режима обрезки OCDF-данных.

## 1.5 Парсинг OCDF-данных.

Основной функцией парсинга данных является разделение данных относительно указанного сида для данных, где имеются значения двух и более сидов, с целью получения данных, содержащих информацию, относящуюся исключительно к указанному сиду.

### 1.5.1 Процесс парсинга OCDF-данных.

Процесс парсинга начинается с того, что программа запрашивает: данные какого сида необходимо получить? Это можно увидеть на рисунке 22.

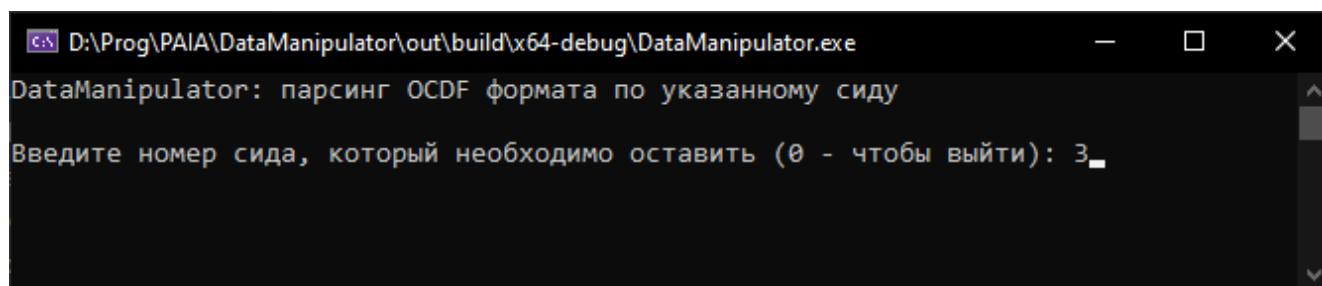


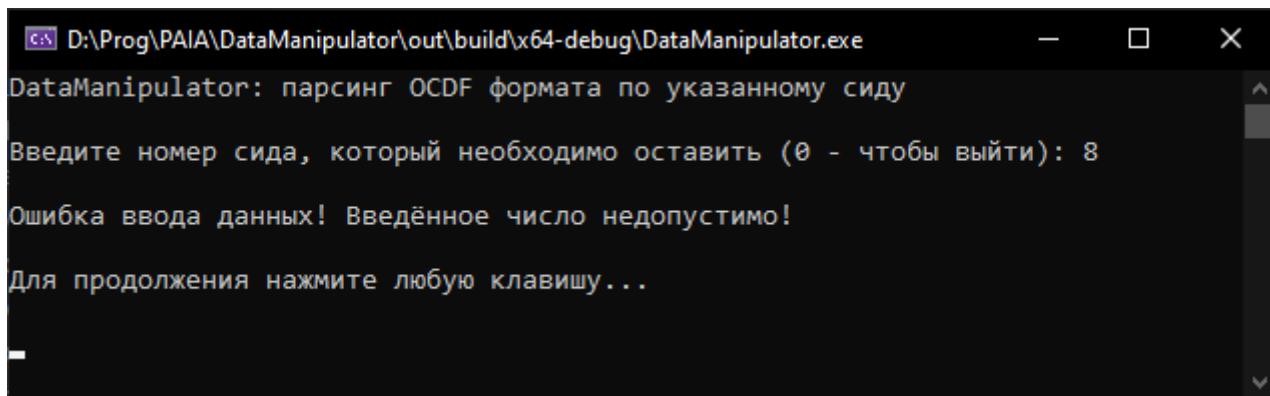
Рис. 22: ввод номера сида для парсинга OCDF-данных.

После ввода номера сида, который необходимо получить, выполняется сам процесс парсинга, результатом которого являются данные, содержащие информацию, касающиеся указанного сида. Данная операция весьма проста.

### 1.5.2 Возможные ошибки при парсинге OCDF-данных.

Несмотря на простоту операции парсинга, всё же есть ситуации, при которых возникают ошибки или исключительные случаи.

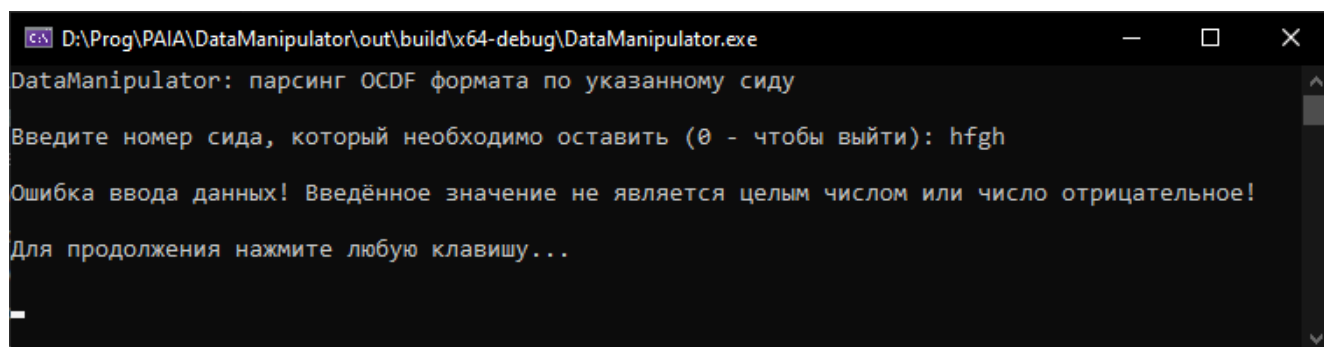
Ошибка может возникать, если ввести недопустимый сид. На данный момент программа считает допустимыми следующие сиды: 1, 2, 3, 4, 5, 6. Иные введённые числа вызывают ошибку, продемонстрированную на рисунке 23.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: парсинг OCDF формата по указанному сиду
Введите номер сида, который необходимо оставить (0 - чтобы выйти): 8
Ошибка ввода данных! Введённое число недопустимо!
Для продолжения нажмите любую клавишу...
```

Рис. 23: ошибка ввода несуществующего сида при парсинге OCDF-данных.

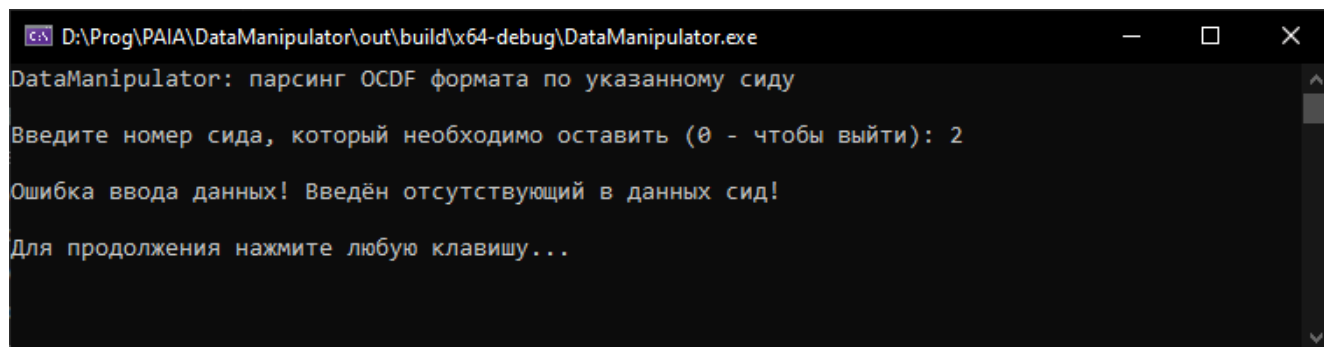
Ошибка также может возникать, если ввести недопустимые символы, т.е. символы отличные от цифр. Данная ошибка продемонстрирована на рисунке 24.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: парсинг OCDF формата по указанному сиду
Введите номер сида, который необходимо оставить (0 - чтобы выйти): hfgH
Ошибка ввода данных! Введённое значение не является целым числом или число отрицательное!
Для продолжения нажмите любую клавишу...
```

Рис. 24: ошибка ввода недопустимого символа при парсинге OCDF-данных.

Также ошибка может возникать в случае, если был введён сид, которого нет в данных, подвергаемых парсингу. Данная ошибка продемонстрирована на рисунке 25.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: парсинг OCDF формата по указанному сиду
Введите номер сида, который необходимо оставить (0 - чтобы выйти): 2
Ошибка ввода данных! Введён отсутствующий в данных сид!
Для продолжения нажмите любую клавишу...
```

Рис. 25: ошибка ввода отсутствующего сида в OCDF-данных.

Результатом выше перечисленных ошибок будет перезапуск операции парсинга.

## 1.6 Выравнивание диапазонов OCDF-данных.

Выравнивание диапазонов OCDF-данных по оси времени предполагает формирование равноинтервальных данных. Такие данные необходимы прежде всего для корректной работы модели, чтобы она могла представлять поведение графика в определённых моментах времени, иначе точность модели упадёт или возникнут трудности в определении моментов времени у предсказанных величин.

### 1.6.1 Математические основы выравнивания диапазонов OCDF-данных.

Равноинтервальные данные предполагают то, что расстояние или интервалы между соседними точками по оси абсцисс или по оси времени будут взаимно равноудалёнными друг от друга. В таком случае, момент времени каждой точки можно выразить с помощью арифметической прогрессии по формуле 1.

$$t_i = t_0 + i \cdot r \quad (1)$$

где  $t_0$  – начальное значение времени;

$r$  – необходимое расстояние между точками, представляющее собой некоторую константу, которую может задать пользователь;

$i$  – номер точки, итерации, элемента прогрессии;  $t_i$  –  $i$ -ый элемент прогрессии;

$t_i$  –  $i$ -ый элемент прогрессии.

Теперь необходимо определить значение ординаты для каждого нового рассчитанного момента времени. Для этого используем уравнение из аналитической геометрии, а именно уравнение прямой, проходящей через две точки, которое можно увидеть на формуле 2.

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} \quad (2)$$

где точка  $(x, y)$  – это точка, которую необходимо найти между точками  $(x_1, y_1)$  и  $(x_2, y_2)$ , взятых из реальных данных.

Про соотношение этих точек известно, что  $x_1 \leq x \leq x_2$ .

Для определения ординаты  $i$ -того элемента прогрессии необходимо взять две ближайших точки из реальных данных относительно оси абсцисс к искомой точке. Тогда для определения ординаты имеем формулу 3.

$$y = \left( \frac{(x - x_1) \cdot (y_2 - y_1)}{x_2 - x_1} \right) + y_1 \quad (3)$$

Таким образом можно получить аппроксимированные данные, где взаимное удаление между двумя соседними точками будет везде во всей цепочке данных постоянным.

### 1.6.2 Процесс выравнивания диапазонов OCDF-данных.

Для получения равноинтервальных OCDF-данных необходимо выбрать соответствующий пункт меню, после чего нас встретит сообщение, запрашивающее задать: каким будет интервал между точками? Это продемонстрировано на рисунке 26.

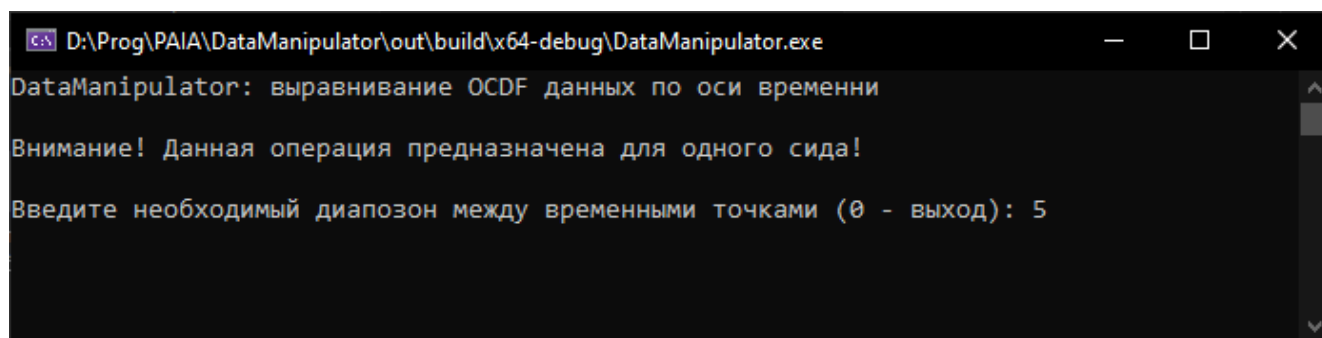


Рис. 26: ввод интервала для аппроксимации OCDF-данных.

Также на рисунке присутствует важное предупреждение, говорящее о том, что данная операция предназначена для данных, содержащих информацию всего по одному сиду.

После ввода начнётся процесс аппроксимации, в результате которого можно будет получить данные, расстояние между соседними точками которых равно введённому пользователем числу.

### 1.6.3 Возможные ошибки при выравнивании диапазонов OCDF-данных.

Также к знакомой нам уже ошибке относится и ввод отрицательных, не целых чисел и ввод отличных от цифр символов, продемонстрированный на рисунке

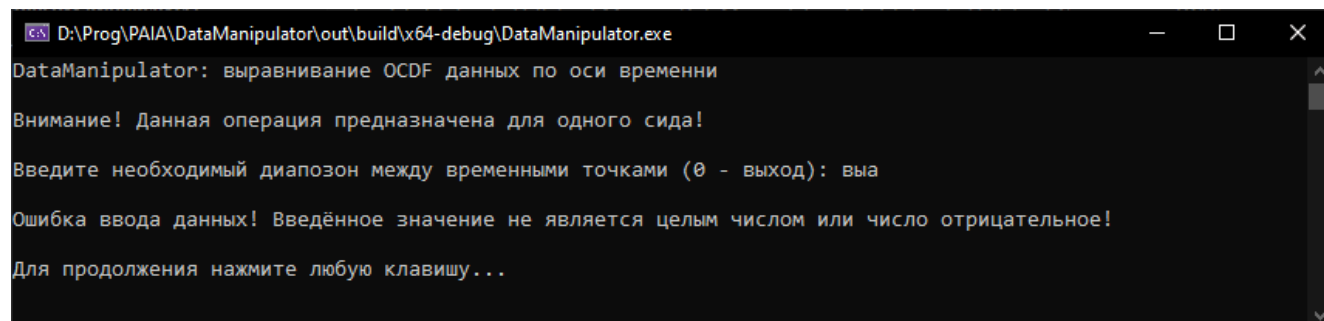


Рис. 27: ввод некорректных значений при аппроксимации OCDF-данных.

По сути, при вводе будущего интервала нет никаких ограничений, однако при вообще больших чисел результат может быть неудовлетворительным. Этот момент остаётся на совести пользователя.

### 1.7 Сохранение OCDF-данных в файл формата csv.

### 1.8 Сохранение OCDF-данных в бинарный файл.

### 1.9 Выход из меню работы с OCDF-данными.