

Университет ИТМО, факультет программной инженерии и компьютерной техники  
Двухнедельная отчётная работа по «Информатике»: аннотация к статье

Дата прошедшей лекции	Номер прошедшей лекции	Название статьи/главы книги/видеолекции	Дата публикации (не старше 2022 года)	Размер статьи (от 400 слов)	Дата сдачи
10.09.2025	1	Data Representation	21.03.2025	430 слов	24.09.2025
24.09.2025	2	LZMA - современный подход к сжатию данных без потерь	14.02.2024	500 слов	08.10.2025

Выполнил(а) Васидов Мухаммадсаид, № группы P3132, оценка \_\_\_\_\_  
Фамилия И.О. студента не заполнять

**Прямая полная ссылка на источник или сокращённая ссылка (bit.ly, tr.im и т.п.)**

<https://swsys.ru/files/2024-2/170-177.pdf>

**Теги, ключевые слова или словосочетания (минимум три слова)**

- LZMA, алгоритмы сжатия, компрессия без потерь, коэффициент сжатия, Python

**Перечень фактов, упомянутых в статье (минимум четыре пункта)**

1. Алгоритм LZMA базируется на LZ77
2. Использует адаптивное арифметическое кодирование
3. Эффективен для больших объемов данных
4. Высокий коэффициент сжатия среди других методов – 8,742
5. Время сжатия LZMA превышает другие методы – 6 минут 38 секунд
6. Доступна через библиотеки Python

**Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта)**

1. Сохраняет информацию без потерь, полностью восстанавливает оригинальные данные.
2. Экономит максимальную дискового пространства.
3. Эффективен для структурированных и избыточных данных, такие как большие текстовые и табличные массивы.
4. Высокая надежность при долгосрочном хранении важных данных.
5. Код в открытом доступе.

**Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта)**

1. Низкая скорость сжатия по сравнению с Deflate, Snappy и Bzip2.
2. Использует сложные модели и больше памяти.
3. Неудобен для приложений, который требует быстрый доступ к данным.
4. Использует только одно ядро процессора ограничивая многопоточность сжатия.
5. Для небольшие файлы неэффективно.

**Ваши замечания, пожелания преподавателю или анекдот о программистах<sup>1</sup>**

— Программист настолько эффективно сжал информацию, что контрольная сумма осталась, а сами данные... не уместились!  
— А декодировать кто будет?  
— Надежда — на помехоустойчивое кодирование.

<sup>1</sup> Наличие этой графы не влияет на оценку