

Задание.

Базовое задание

Есть файл input.txt, где построчно записаны данные. В каждой строке записано несколько элементов. Их типы описываются с помощью TypeList. Типы и порядок элементов во всех строках одинаковые.

Необходимо разработать шаблонный класс Reader, который будет получать на вход TypeList с описанием строки в файле, открывать этот файл и позволять с помощью метода void* readNextLine() читать по очереди все строки в файле. Возвращаемое значение функции – указатель на область памяти, где последовательно записаны считанные элементы строки.

Задание на макс. 8

Теперь элементы в строках могут быть сжаты неким алгоритмом. Как параметры класса Reader даются три TypeList. Первый – все так же говорит типы элементов в строке (после декомпрессии). Второй – говорит, к какому типу относится сжатый элемент(например, CompressedType). У этого типа должна быть определена функция decompress(). Если же в этом TypeList указан фиксированный тип NoneType, то значит, что специализированного типа для сжатого элемента нет – и нужно обращаться к третьему TypeList. В третьем TypeList для всех элементов, у которых во втором TypeList был указан конкретный тип, содержится тип NoDecompressor. Для тех же, у кого был указан тип NoneType, должен содержаться тип класса с функцией decompress() – и тогда ее нужно применить к считанному из файла элементу для декомпрессии.

Задание на 9-10.

Во втором TypeList может быть указан класс с функцией decompress(), либо NoneType. Вместо третьего TypeList Reader принимает список указателей на функцию(либо nullptr). Декомпрессоры из второго TypeList и списка функций применяются последовательно. Если во втором TypeList NoneType для этого элемента – этот шаг декомпрессии опускается. Если в списке указателей на функцию nullptr – этот шаг декомпрессии опускается.