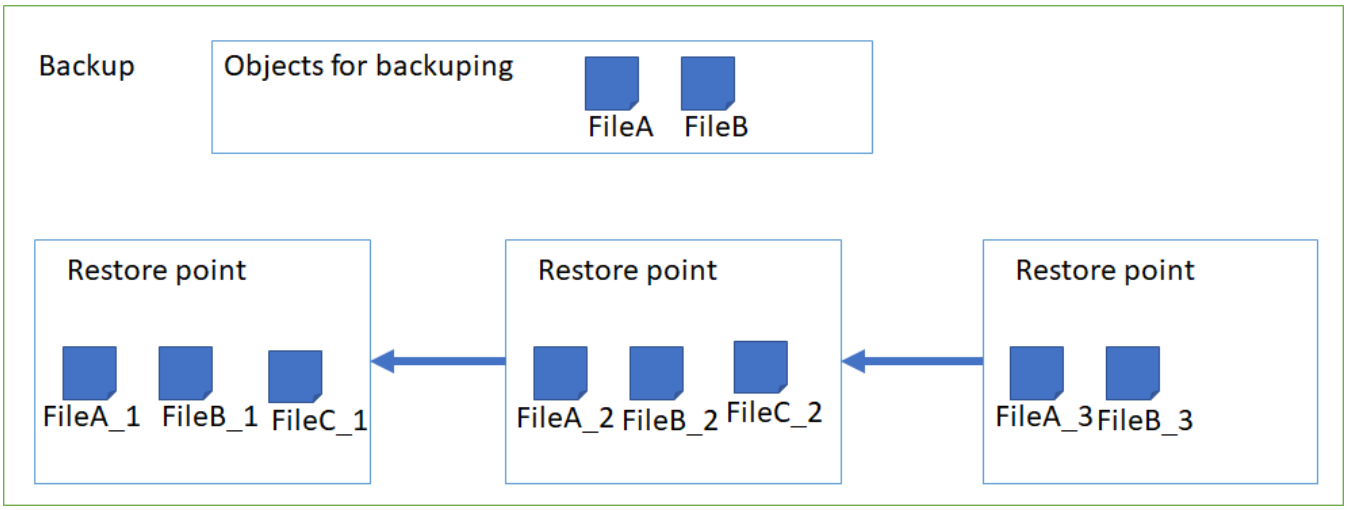
Лабораторная работа 4

Бекапы

Теормин

Бекап — в общем случае, это резервная копия каких-то данных, которая делается для того, чтобы в дальнейшем можно было восстановить эти данные, то есть откатиться до того момента, когда она была создана.

Точка восстановления — резервная копия объектов, созданная в определенный момент. Представлена датой создания и список резервных копий объектов, которые бекапились. Есть два типа точек восстановления - полноценные и инкрементальные. Полноценные точки содержат всю информацию про объекты, которые забекапились. Инкрементальные поинты - это разница (дельта) относительно предыдущей точки, т.е. мы храним только изменения.



Условие

В рамках лабораторной работы подразумевается разработка системы, которая управляет процессом создания бекапов. Для упрощения выполнения лабораторной, создавать физически резервные копии указанных файлов не требуется. Достаточно будет создать запись о том, что было сделано резервное копирование.

```
FileRestoreCopyInfo CreateRestore(string filePath)
{
    var fileInfo = new FileInfo(filePath);
    var fileRestoreCopyInfo = new FileRestoreCopyInfo(filePath, fileInfo.Size, DateTime.Now);
    //File.Copy(filePath, _pathWhereWeNeedToStoreOurBackup); <- Вот эту часть мы можем скипать
    return fileRestoreCopyInfo;
}
```

Информация о бекапе представлена в виде набора параметров: Id, CreationTime, BackupSize и список точек восстановления.

Алгоритмы создания и хранения

Для создания бекапа указываются объекты: список файлов или папок. Должна быть реализована возможность в последствии этот список редактировать - добавлять и удалять объекты из списка объектов которые будут обрабатываться в алгоритме.

Система должна поддерживать несколько алгоритмов создания точек восстановления для бекапа, а также возможность увеличивать их количество. Результатом работы алгоритма является создание новой точки восстановления для указанного бекапа. Точка восстановления хранит о себе информацию о том, какие объекты были в ней забекаплены. В алгоритме должна быть возможность указать требуется ли создать полноценную точку или только дельту с прошлого раза (т.е. инкремент).

Требуется реализовать как минимум два алгоритма хранения:

1. Алгоритм раздельного хранения — файлы копируются в специальную папку и хранятся там раздельно.
2. Алгоритм общего хранения — все указанные в бекапе объекты складываются в один архив.

Алгоритмы очистки точек

Помимо создания, нужно контролировать количество хранимых точек восстановления. Чтобы не допускать накопления большого количества старых и неактуальных точек, требуется реализовать механизмы их очистки — они должны контролировать, чтобы цепочка точек восстановления не выходила за допустимый лимит. В рамках лабораторной подразумеваются такие типы лимитов:

1. По количеству - ограничивает длину цепочки точек восстановления (храним последние N точек)

2. По дате - ограничивает насколько старые точки будут хранится (храним все точки, которые были сделаны не позднее указанной даты)

3. По размеру - ограничивает суммарный размер, занимаемый бекапом (храним все последние бекапы, суммарный размер которых не превышает лимит)

4. Гибрид - возможность комбинировать лимиты. Пользователь может указывать, как комбинировать:

    - нужно удалить точку, если вышла за хотя бы один установленный лимит
    - нужно удалить точку, если вышла за все установленные лимиты

    Например, пользователь выбирает гибрид алгоритмов "по количеству" и "по дате". Если по одному из алгоритмов необходимо оставить 3 точки, а по другому — 5, то выбирается количество точек в соответствии с параметром, указанном при создании "гибрида" (использовать максимальное или минимальное значение отобранных точек).

Алгоритм должен учитывать то, что инкрементальные точки не должны остаться без точки, от которой взята дельта. В случае, если пришлось оставить точек больше, чем планировалось, результат выполнения алгоритма должен вернуть соответствующее предупреждение.

Тесты

Для проверки работоспособности разработанной системы предлагается реализовать юз-кейсы (пользовательские сценарии использования программы):

1. Кейс №1
    1. Я создаю бекап, в который добавляю 2 файла.
    2. Я запускаю алгоритм создания точки для этого бекапа — создается точка восстановления.
    3. Я должен убедиться, что в этой точке лежит информация по двум файлам.
    4. Я создаю следующую точку восстановления для цепочки
    5. Я применяю алгоритм очистки цепочки по принципу ограничения максимального количества указав длину 1.
    6. Я убеждаюсь, что в ответ получу цепочку длиной 1.
2. Кейс №2
    1. Я создаю бекап, в который добавляю 2 файла размером по 100 мб.
    2. Я создаю точку восстановления для него.
    3. Я создаю следующую точку, убеждаюсь, что точки две и размер бекапа 200 мб.
    4. Я применяю алгоритм очистки с ограничением по размеру, указываю 150 мб для цепочки и убеждаюсь, что остается один бекап.
3. Кейсы с инкрементами на тестирование алгоритмов сохранения.
4. Кейсы на два способа комбинации в гибридном лимите.

Notes

1. Для проверки работоспособности алгоритма работы со временем вам нужно будет продумать механизм путешествия во времени.