

Предметная область

Backup Object – объект, который отслеживается системой для создания резервных копий. Объекты могут быть как файлами, так и папками.

Restore Point – снимок набора отслеживаемых объектов. Должна как минимум хранить дату создания и коллекцию **Backup Object**, отслеживаемых на момент создания **Restore Point**.

Backup – в контексте данной системы, **Backup**, это коллекция **Restore Point**, то есть, полная история резервного копирования в рамках одной **Backup Task**.

Backup Task – сущность хранящая конфигурацию резервного копирования (текущий список **Backup Object**, способов хранения, способов сжатия) и информацию о созданных **Restore Point**. Так же **Backup Task** должна инкапсулировать логику своего выполнения, т.е. создания новых **Restore Point**.

Repository – абстракция над хранением и записью данных, будь то данные объектов, соответствующих **Backup Object**, либо данные о каком-либо **Storage**. Простейшая реализация репозитория в рамках данной лабораторной - абстракция над чтением/записью в некоторую директорию на локальной файловой системе.

Storage – файл, в котором хранится копия данных, соответствующих какому-либо **Backup Object**, созданная в конкретной **Restore Point**.

Пример логики работы

Выполняем следующие действия:

1. Имеем в **Repository** три объекта File A, File B, Folder C.
2. Создаём **Backup Task**, добавляем в неё три **Backup Object**, соответствующие объектам находящимся в репозитории.
3. Запускаем выполнение **Backup Task**, получаем **Restore Point**, он записывается в репозиторий, в соответствующей директории появляются **Storage** File A(1), File B(1), Folder C(1).
4. Запускаем выполнение ещё раз, получаем **Storage** с версиями (2).
5. Удаляем из **Backup Task** File B, запускаем выполнение ещё раз, получаем третий **Restore Point**, ему будут соответствовать два **Storage** - File A(3), Folder C(3).

Создание резервных копий

Под созданием резервной копии данных, подразумевается создание копии данных в другом месте. Система должна поддерживать расширяемость в вопросе выбора **Storage Algorithm**, используемых для хранения резервных копий (должна иметь возможность добавить новый алгоритм безболезненно, помним про OCP).

В данной лабораторной требуется реализовать два **Storage Algorithm**:

1. **Split Storage** – алгоритм раздельного хранения, для каждого **Backup Object** в **Restore Point** создаётся отдельный **Storage** - архив, в котором лежат данные объекта.
2. **Single Storage** – алгоритм общего хранения, для всех **Backup Object** в **Restore Point** создаётся один общий **Storage** - архив, в котором лежат данные каждого объекта.

Storage Algorithm не должен нести ответственность за реализацию архивации.

Хранение копий

В лабораторной работе подразумевается, что резервные копии будут создаваться локально на файловой системе. Но логика выполнения должна абстрагироваться от этого, должна быть введена абстракция - репозиторий (см. принцип DIP из SOLID).

В тестах стоит реализовать хранение в памяти (как вариант - **InMemoryRepository** с использованием паттерна Composite), так как при запуске тестов на настоящей файловой системе будет генерироваться много мусорных данных, а так же системы CI не дружат с запросами к файловой системе во время автоматического выполнения тестов.

Ожидаемая структура:

- Корневая директория
 - Директории различных **Backup Task**
 - Директории различных **Restore Point**
 - Файлы **Storage**

Создание Restore Point

Backup Task отвечает за создание новых точек восстановления, выступает фасадом, инкапсулируя логику выполнения этой операции.

При создании **Backup Task** должна быть возможность указать её название, **Repository** для хранения **Backup** (его данных), **Storage Algorithm**.

Backup Task должна поддерживать операции добавления и удаления отслеживаемых ей **Backup Object**.

Результатом выполнения **Backup Task** является создание **Restore Point** и соответствующих ей **Storage** в выбранном **Repository**.

Тест кейсы

1. Тест 1
 1. Создаём **Backup Task**, использующую **Split Storage**
 2. Добавляем в **Backup Task** два **Backup Object**
 3. Запускаем выполнение **Backup Task**
 4. Удаляем из **Backup Task** один **Backup Object**
 5. Запускаем выполнение **Backup Task**
 6. Проверяем то, что было создано две **Restore Point** и три **Storage**

2. Тест 2 (лучше оформить в виде консольного приложения, так как нормально проверить можно только на настоящей файловой системе)
 1. Создаём **Backup Task**, использующую **FileSystemRepository** и **Single Storage**
 2. Добавляем в **Backup Task** два **Backup Object**
 3. Запускаем выполнение **Backup Task**
 4. Проверяем то, что директории и файлы были созданы