



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №2
Технології розроблення програмного забезпечення
*Діаграма варіантів використання. Сценарії варіантів
використання. Діаграми UML. Діаграми класів.
Концептуальна модель системи.*
Варіант №14

Виконав:
студенти групи ІА-14,
Міщук Максим Дмитрович

Перевірив:

Драган М.С.

Тема роботи: Діаграма варіантів використання. Сценарії варіантів використання. Діаграми UML. Діаграми класів. Концептуальна модель системи.

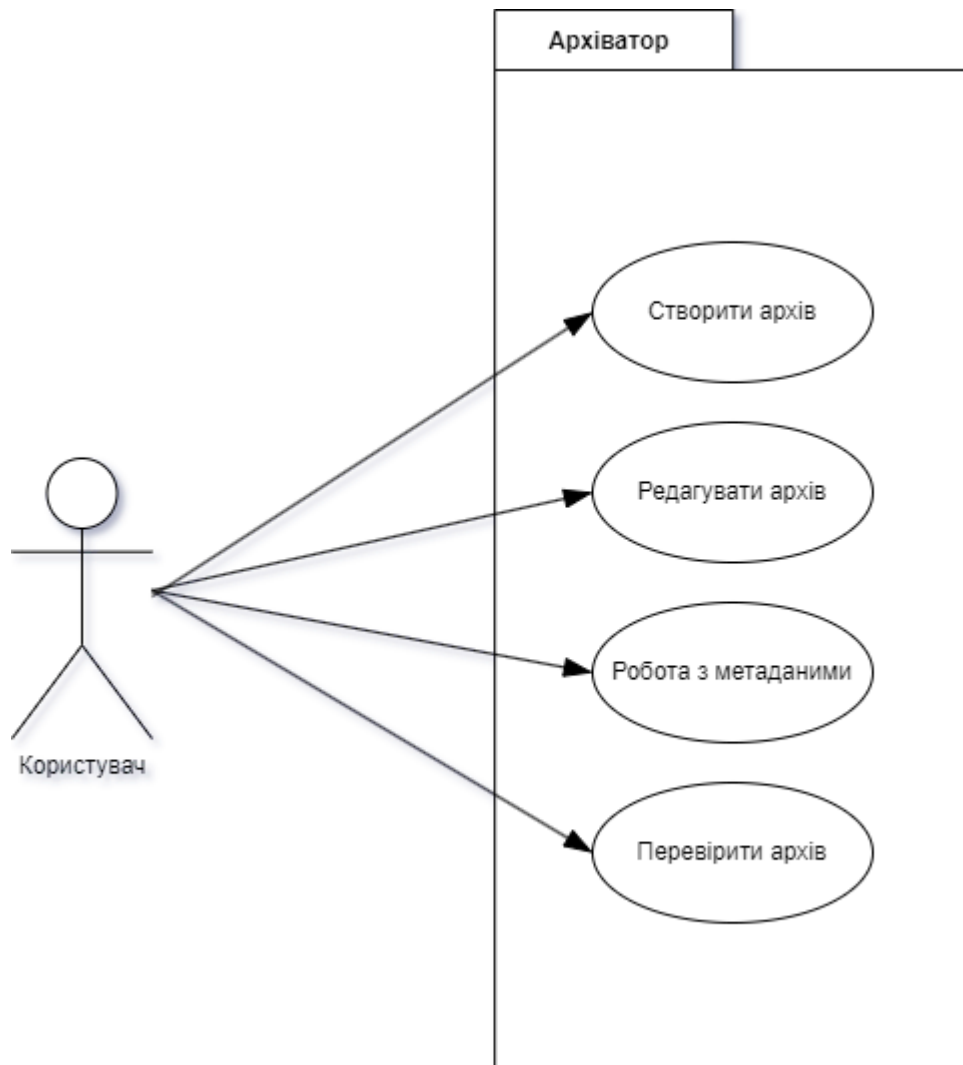
Вхідні дані:

..14 Архіватор (strategy, adapter, factory method, facade, visitor, p2p)

Архіватор повинен являти собою візуальний додаток з можливістю створення і редагування архівів різного типу (.tar.gz, .zip, .rar, .ace) - додавання/ видалення файлів / папок, редагування метаданих (по можливості), перевірка checksum архівів, тестування архівів на наявність пошкоджень, розбиття архівів на частини.

Хід роботи:

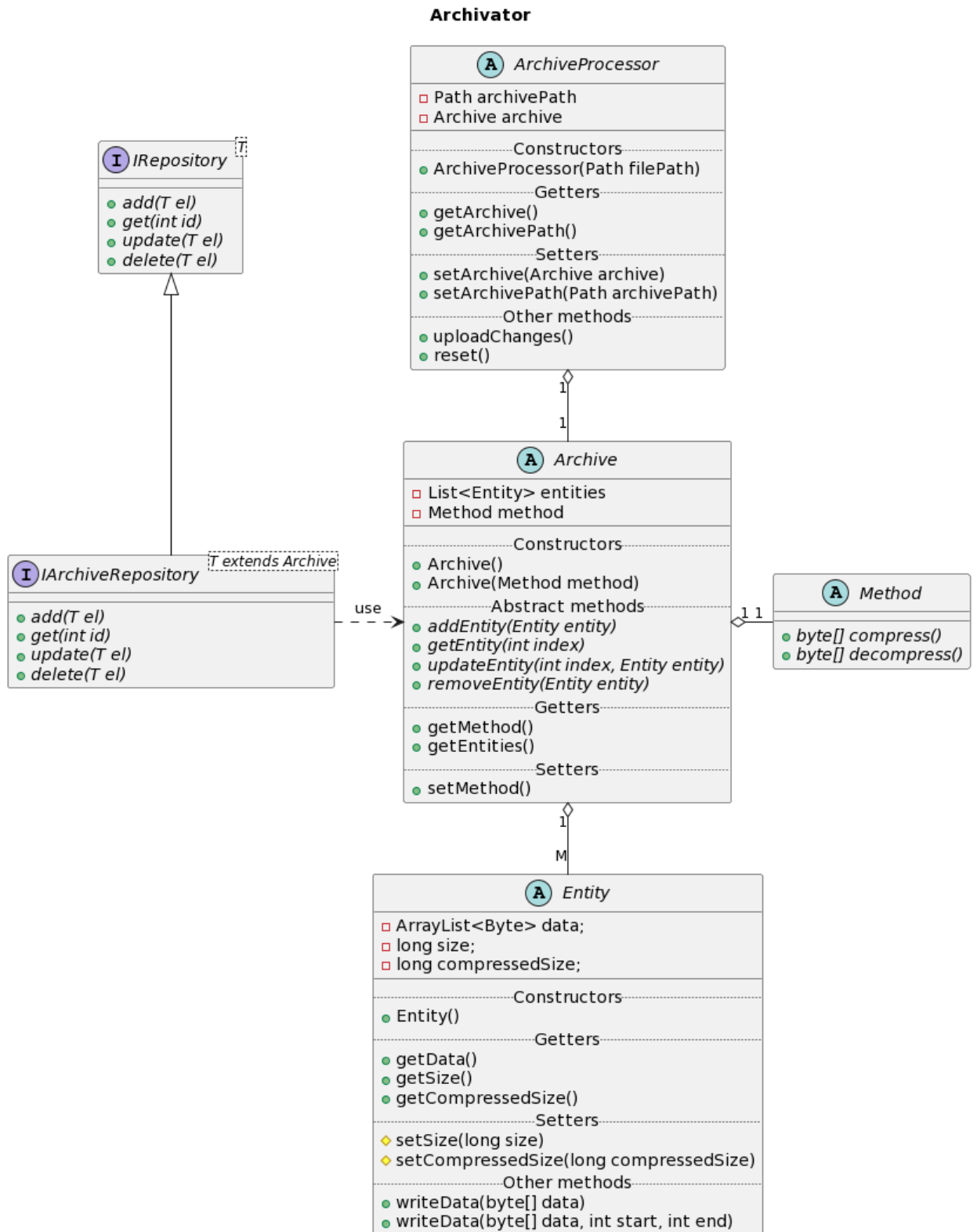
1. Зобразити діаграму прецедентів:



2. До отриманих прецедентів створити похідні прецеденти:



3. Зобразити діаграму класів:



4. Розробити основні класи та структури систем баз даних:

```
package labs.lab2;

import java.util.ArrayList;

7 usages
public abstract class Entity {
    2 usages
    private ArrayList<Byte> data;
    2 usages
    private long size;
    2 usages
    private long compressedSize;

    public Byte[] getData() {
        Byte[] arr = new Byte[data.size()];
        return data.toArray(arr);
    }

    public long getSize() { return size; }

    public long getCompressedSize() { return compressedSize; }

    protected void setSize(long size) { this.size = size; }

    protected void setCompressedSize(long compressedSize) { this.compressedSize = compressedSize; }

    public void writeData(byte[] data) { writeData(data, start: 0, end: data.length - 1); }

    1 usage
    public void writeData(byte[] data, int start, int end) { throw new UnsupportedOperationException(); }
}
```

```

package labs.lab2;

import java.util.List;

4 usages
public abstract class Archive {
    1 usage
    public static Method defaultMethod;
    1 usage
    private List<Entity> entities;
    3 usages
    private Method method;

    public Archive() {
        this(defaultMethod);
    }

    1 usage
    public Archive(Method method) {
        this.method = method;
    }

    public List<Entity> getEntities() {
        return entities;
    }

    public Method getMethod() {
        return method;
    }

    public void setMethod(Method method) {
        this.method = method;
    }

    public abstract boolean addEntity(Entity entity);

    public abstract Entity getEntity(int index);

    public abstract Entity updateEntity(int index, Entity entity);

    public abstract boolean removeEntity(Entity entity);
}

```

```
package labs.lab2;

public abstract class Method {
    abstract byte[] compress(byte[] data);
    abstract byte[] decompress(byte[] data);
}
```

```
package labs.lab2;

import java.nio.file.Path;

public class ArchiveProcessor {
    3 usages
    private Path archivePath;
    2 usages
    private Archive archive;

    public ArchiveProcessor(Path filePath) {
        this.archivePath = filePath;
    }

    public Path getArchivePath() {
        return archivePath;
    }

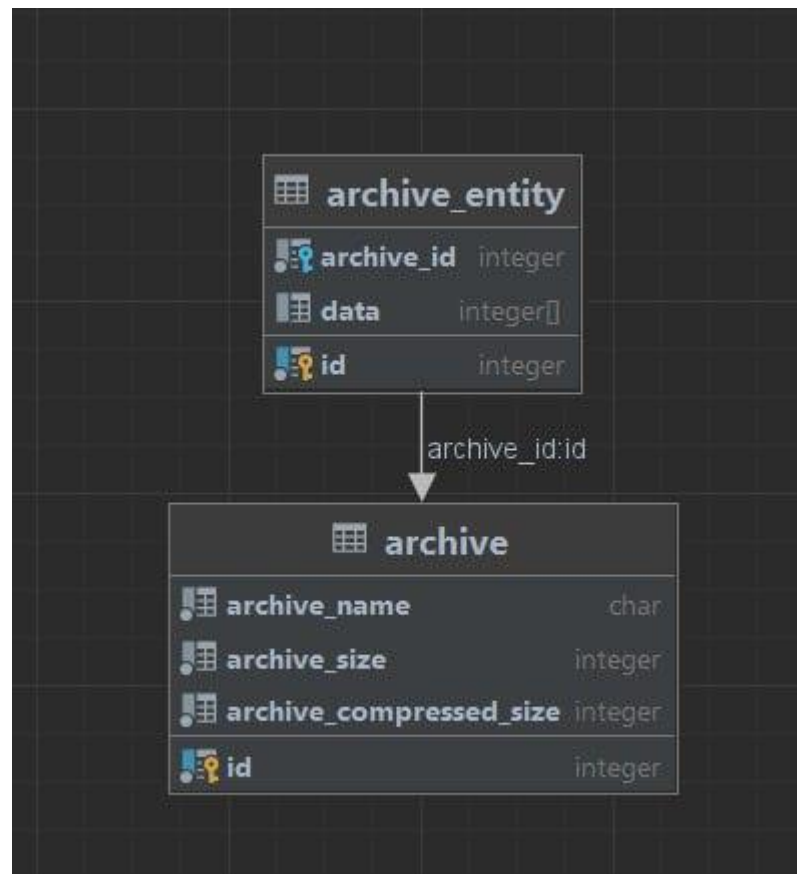
    public void setArchivePath(Path archivePath) {
        this.archivePath = archivePath;
    }

    public Archive getArchive() {
        return archive;
    }

    public void setArchive(Archive archive) {
        this.archive = archive;
    }

    public boolean uploadChanges() { throw new UnsupportedOperationException(); }

    public boolean reset() {
        throw new UnsupportedOperationException();
    }
}
```



5. Для класів даних реалізувати шаблон Репозиторію:

```
package labs.lab2;

public interface IRepository<T> {
    T add(T el);
    T get(int id);
    T update(T el);
    T delete(T el);
}
```

```
public interface IArchiveRepository<T extends Archive> extends IRepository<T>
```