Project realization report

VoteChain.

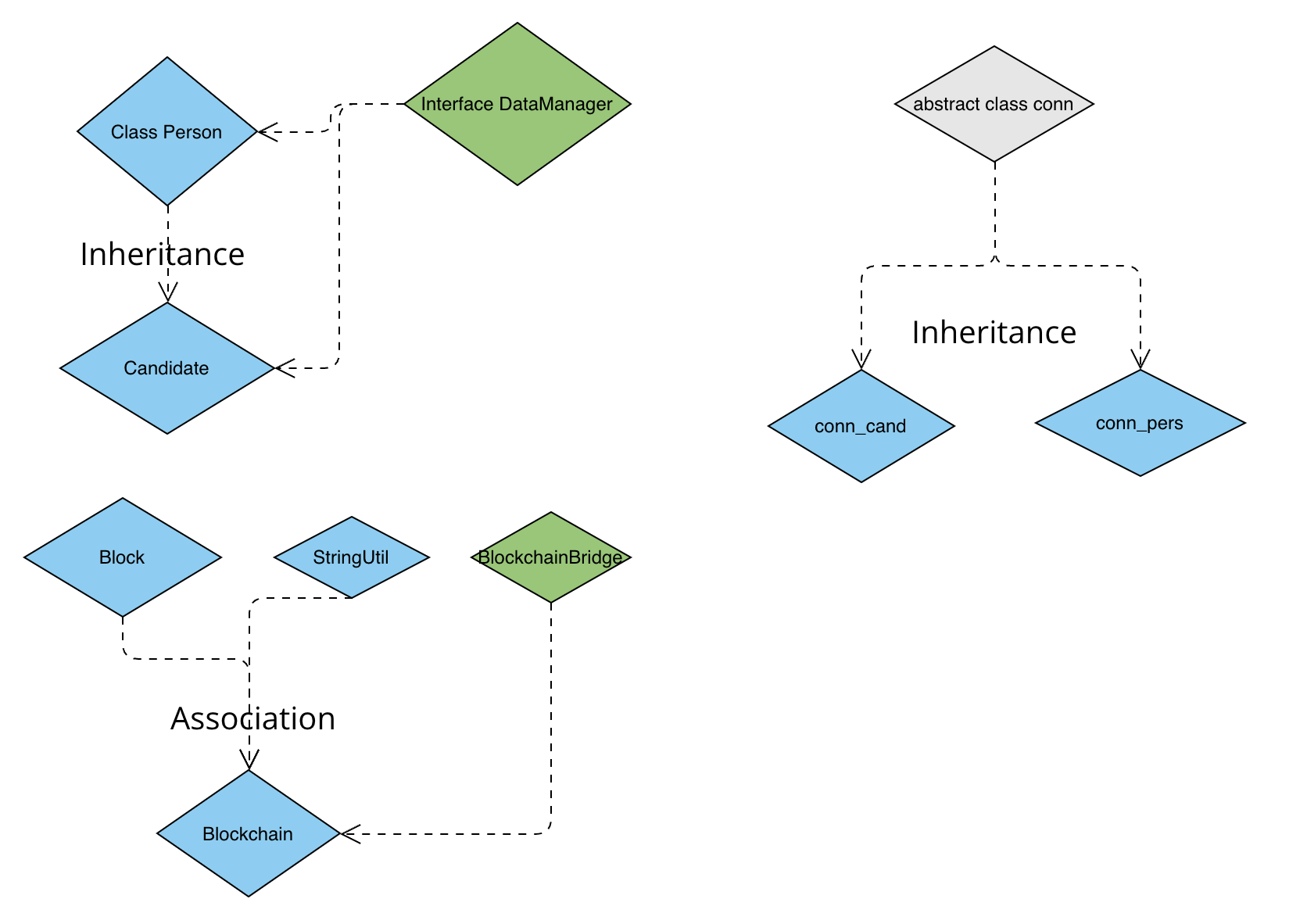
The goal of the project was to create a platform for conducting government elections using blockchain to ensure fairness, security and transparency of the system.

Login to the system is carried out through a database where all citizens should be stored; the passport number is used as a login. The election ends when the entire population has voted.

After sending a vote for a candidate, each user is given a hash of his vote, using which he can view all information about his vote in a special window, including the hash of the previous vote, and verify the integrity of the system. In addition, it will be possible to search all the votes at once using a hash and make sure that the election results are calculated correctly.

The program works correctly, performs all intended functions and has a pleasant and user-friendly interface. Therefore, I am satisfied with my implementation of the project.

**Main class diagram:**



The Person class creates a basic person prototype that stores the main data and is responsible for the voting method.

The Candidate class inherits from person, the main difference is that a candidate cannot vote.

The Blockchain class stores created instances of the Block class, where transaction and voting data is stored.

The StringUtil class handles encryption and hash creation.

The classes conn, conn\_cand, conn\_pers work with databases.

**My assessment of the fulfillment of the required criteria:**

**Main criteria:**

1. The program must work and correspond to the project assignment approved by the teacher and to principal teacher's requirements that occurred during the project realization. **completed**
2. The submitted source code must include all necessary files and it must be possible to compile it in the Eclipse environment installed in the laboratory where the exercises are held. **completed**
3. The program must embrace a meaningful inheritance between own classes and overriding of own methods. **completed**
4. Encapsulation has to be employed in the program. **completed**
5. The program must embrace sufficient comments to understand the code. **completed**
6. The documentation must correspond to the program and it must contain a class diagram. **completed**
7. A student has to be capable of answering all the teacher's questions regarding the project during the final presentation. **completed**

In my opinion, all the required criteria have been fulfilled.

**Successfully fulfilled additional criteria:**

1) Using the Prototype and Bridge design patterns.

2) Using own exception: AlreadyVotedException.

3) Providing a graphical user interface.

4) Using generics: Array List, List…

5) Explicit use of RTTI: instanceof in Person, to fill the list of persons

6) Using lambda expressions: in blockchain class, in chain validation method.

7)Using default method implementation in interfaces: in DataManager interface, IsEmpty method.

**Main versions of the program:**

27.03.2024 First version, beginning of creating of the interface

03.04.2024 Main working version, basic functionality and design done

06.04.2024 Create README.md

07.04.2024 Added search by hash

17.04.2024 Add CSS, design improvements

18.04.2024 Added button for hash copy

04.05.2024 small improvements, final version