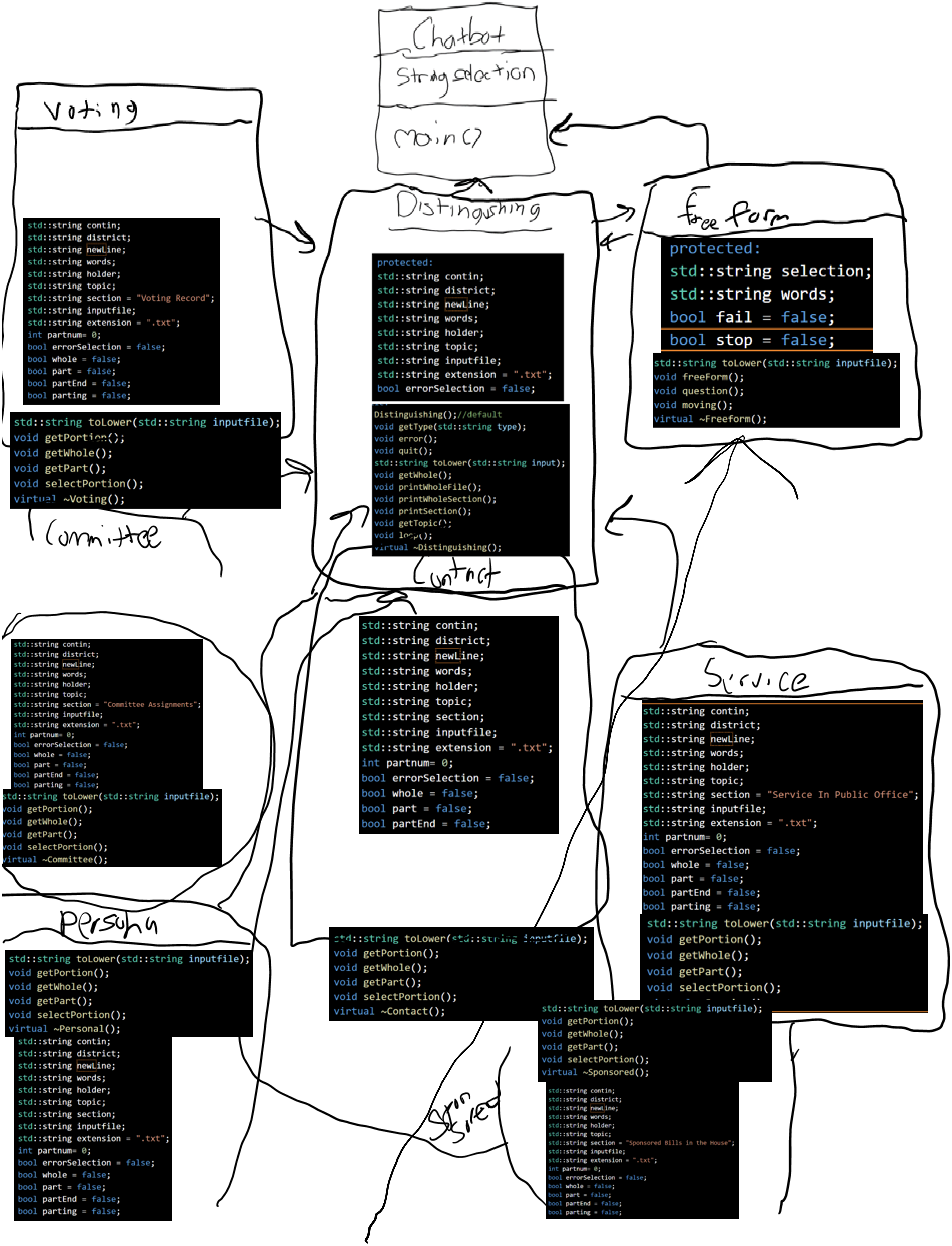


Chatbot

String selection

main()

Voting

```
std::string contin;
std::string district;
std::string newLine;
std::string words;
std::string holder;
std::string topic;
std::string section = "Voting Record";
std::string inputfile;
std::string extension = ".txt";
int partnum= 0;
bool errorSelection = false;
bool whole = false;
bool part = false;
bool partEnd = false;
bool parting = false;
```

```
std::string toLower(std::string inputfile);
void getPortion();
void getWhole();
void getPart();
void selectPortion();
virtual ~Voting();
```

Distinguishing

```
protected:
std::string contin;
std::string district;
std::string newLine;
std::string words;
std::string holder;
std::string topic;
std::string inputfile;
std::string extension = ".txt";
bool errorSelection = false;
```

```
Distinguishing();//default
void getType(std::string type);
void error();
void quit();
std::string toLower(std::string input);
void getWhole();
void printWholeFile();
void printWholeSection();
void printSection();
void getTopic();
void loop();
virtual ~Distinguishing();
```

free form

```
protected:
std::string selection;
std::string words;
bool fail = false;
bool stop = false;
```

```
std::string toLower(std::string inputfile);
void freeForm();
void question();
void moving();
virtual ~Freeform();
```

Committee

```
std::string contin;
std::string district;
std::string newLine;
std::string words;
std::string holder;
std::string topic;
std::string section = "Committee Assignments";
std::string inputfile;
std::string extension = ".txt";
int partnum= 0;
bool errorSelection = false;
bool whole = false;
bool part = false;
bool partEnd = false;
bool parting = false;
std::string toLower(std::string inputfile);
void getPortion();
void getWhole();
void getPart();
void selectPortion();
virtual ~Committee();
```

Contact

```
std::string contin;
std::string district;
std::string newLine;
std::string words;
std::string holder;
std::string topic;
std::string section;
std::string inputfile;
std::string extension = ".txt";
int partnum= 0;
bool errorSelection = false;
bool whole = false;
bool part = false;
bool partEnd = false;
```

```
std::string toLower(std::string inputfile);
void getPortion();
void getWhole();
void getPart();
void selectPortion();
virtual ~Contact();
```

Service

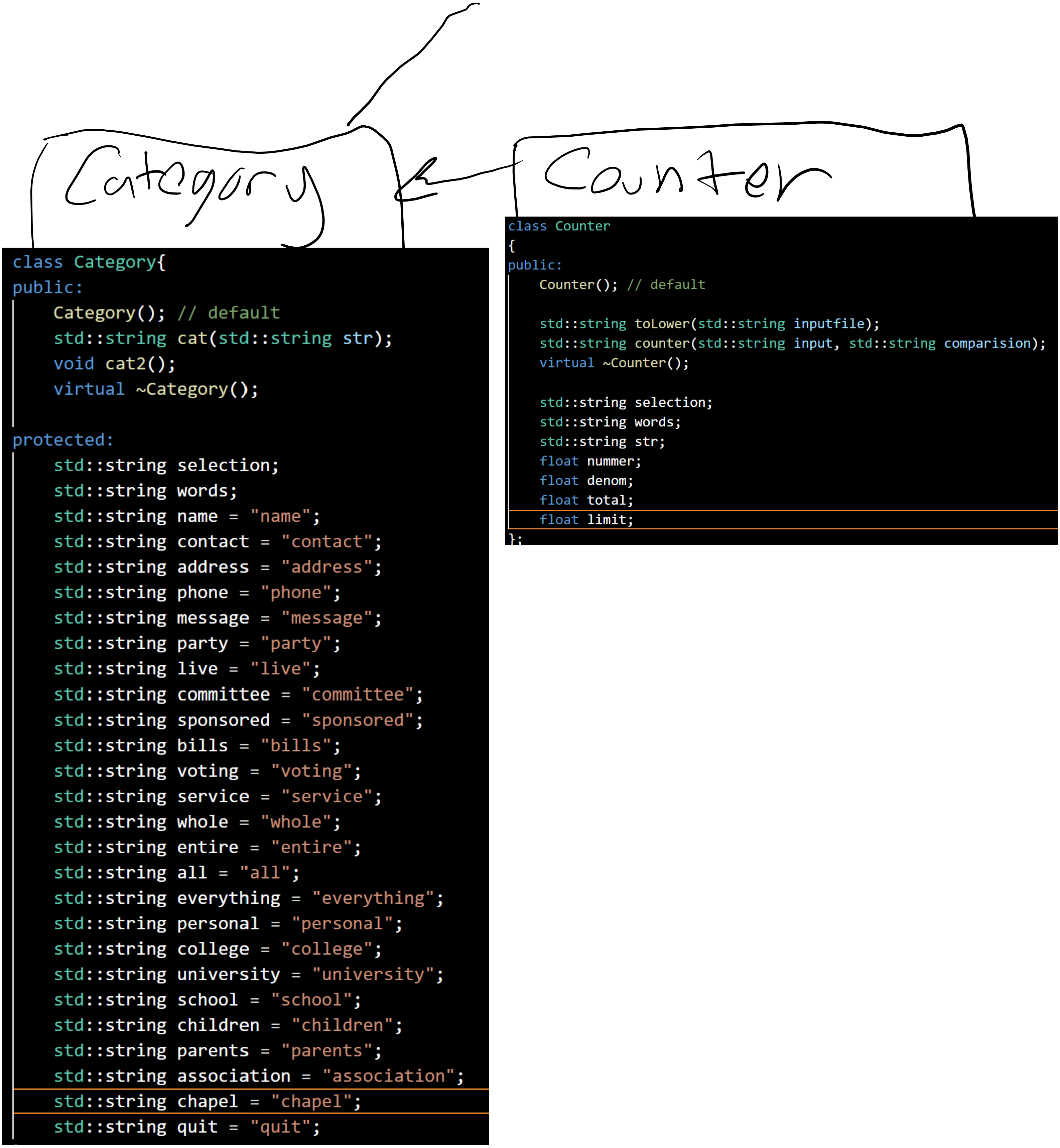
```
std::string contin;
std::string district;
std::string newLine;
std::string words;
std::string holder;
std::string topic;
std::string section = "Service In Public Office";
std::string inputfile;
std::string extension = ".txt";
int partnum= 0;
bool errorSelection = false;
bool whole = false;
bool part = false;
bool partEnd = false;
bool parting = false;
std::string toLower(std::string inputfile);
void getPortion();
void getWhole();
void getPart();
void selectPortion();
```

Persona

```
std::string toLower(std::string inputfile);
void getPortion();
void getWhole();
void getPart();
void selectPortion();
virtual ~Personal();
std::string contin;
std::string district;
std::string newLine;
std::string words;
std::string holder;
std::string topic;
std::string section;
std::string inputfile;
std::string extension = ".txt";
int partnum= 0;
bool errorSelection = false;
bool whole = false;
bool part = false;
bool partEnd = false;
bool parting = false;
```

Sponsred

```
std::string toLower(std::string inputfile);
void getPortion();
void getWhole();
void getPart();
void selectPortion();
virtual ~Sponsored();
std::string contin;
std::string district;
std::string newLine;
std::string words;
std::string holder;
std::string topic;
std::string section = "Sponsored Bills in the House";
std::string inputfile;
std::string extension = ".txt";
int partnum= 0;
bool errorSelection = false;
bool whole = false;
bool part = false;
bool partEnd = false;
bool parting = false;
```



```cpp
class Category{
public:
    Category(); // default
    std::string cat(std::string str);
    void cat2();
    virtual ~Category();

protected:
    std::string selection;
    std::string words;
    std::string name = "name";
    std::string contact = "contact";
    std::string address = "address";
    std::string phone = "phone";
    std::string message = "message";
    std::string party = "party";
    std::string live = "live";
    std::string committee = "committee";
    std::string sponsored = "sponsored";
    std::string bills = "bills";
    std::string voting = "voting";
    std::string service = "service";
    std::string whole = "whole";
    std::string entire = "entire";
    std::string all = "all";
    std::string everything = "everything";
    std::string personal = "personal";
    std::string college = "college";
    std::string university = "university";
    std::string school = "school";
    std::string children = "children";
    std::string parents = "parents";
    std::string association = "association";
    std::string chapel = "chapel";
    std::string quit = "quit";
```

```cpp
class Counter
{
public:
    Counter(); // default

    std::string toLower(std::string inputfile);
    std::string counter(std::string input, std::string comparision);
    virtual ~Counter();

    std::string selection;
    std::string words;
    std::string str;
    float nummer;
    float denom;
    float total;
    float limit;
};
```