Architectural Design

Team Two:

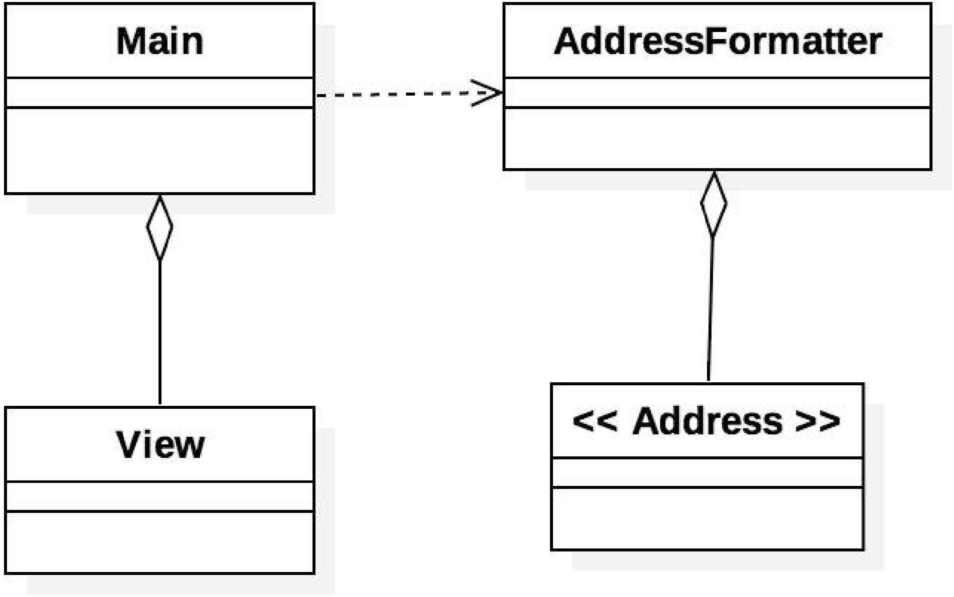
Jessica DiMariano
Sean McDonald
Xi Chen
Jason Lueckenhoff
Aaron Gershman

Class Structure

- main-subroutine
  - allows methods to be separated into specialized functions
  - allows multiple team members to experiment with a single method implementation, without interfering with other methods
- initiating program is done in “Main”
- formatting addresses will be done in "AddressFormatter"
  - aggregates “addresses” from data input
  - where most of our program logic will be done
- “Views” can be written separately from the program logic

Class Diagram

- Main
- View
    - simple navigation screens
- AddressFormatter
    - aggregates multiple addresses
    - where the program logic resides, making changes to the data
- Address
    - individual address entities



Decoupling Benefits of Design

- the "main-subroutine" architectural pattern is simple and modular
- new features could easily added
  - adding “AI engines” or other optional discussed enhancements can be added in the future without interfering with existing structures
- new *experimental* features can be tested by multiple members simultaneously
- separating the addresses into specialized functions helps maintain information hiding principles
  - methods which format ZIP will *only* receive information on ZIP, etc.

Implementation Language

- Python 3.4+
    - readable code without verbosity of Java
    - supports multi-paradigm styles
        - object-oriented, functional, etc.
    - does not require a special IDE
        - PyCharm, Sublime Text … Notepad.
    - very portable language (usable on nearly all machines)
    - many built-in methods for manipulating strings, lists, tuples, etc.

Implementation Language Cont.

- Python 3.4+
  - many graphics packages for user interface design
    - TkInter (ships with python, lightweight, multi-paradigm, etc.)
  - Python 2.x only supported until 2020
  - healthy open-source community
    - packages such as “usaddress” can be used for tagging different parts of the addresses

Bonus: usaddress + part of speech tagging

code

```
import usaddress

gersh_address = ('303 Hillcrest Drive,
Kirksville MO 63501-2292')


out = usaddress.parse(gersh_address)
for x in out:
      print (x)
```

output

```
(u'303', 'AddressNumber')
(u'Hillcrest', 'StreetName')
(u'Drive,', 'StreetNamePostType')
(u'Kirksville', 'PlaceName')
(u'MO', 'StateName')
(u'63501-2292', 'ZipCode')
```