Object Processing

Matchmaking service for professionals!

Goals:

- search and match people by profession
- search and match people by country of residence
- search and match people by name

Raw Data

The data are highly structured!!!!

- Name, Country of Residence, Cell Number, Profession, Email
- each new line is a different person
- every person has the five attributes (^^^)
- every item is separated using a comma

```
Samantha Rhodes,Maldives,(912)136-3882,"Research officer, trade union",terryjames@example.net
Matthew Johnson,Iran,001-366-114-0721x393,Agricultural consultant,molly98@example.com
Summer Stewart,Jamaica,2494404249,Economist,ovazquez@example.com
William Anderson,United States Virgin Islands,+1-888-635-0096x9565,Museum/gallery exhibitions officer,randyhartman@example.net
Jeremy Bates,Central African Republic,+1-449-207-9863x997,Chartered public finance accountant,livingstonamanda@example.net
Jorge Wright,Peru,(970)111-7796,"Surveyor, building control",harringtonmichael@example.com
Dustin Jackson,Gambia,755-090-9702x49724,Immunologist,ijennings@example.org
Joan Paul,San Marino,(218)682-4690x416,"Surveyor, planning and development",mcdonaldrenee@example.org
```

Designing a system

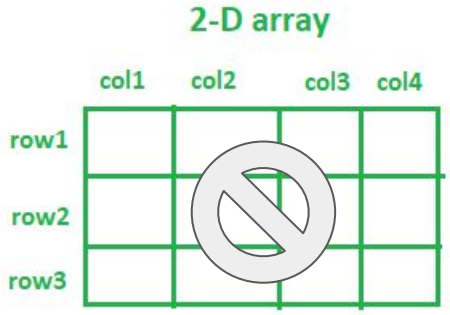
- Need an algorithm that does matching
  - need to know what the actual **search term** is
  - need to know what **attribute** to examine

- Example
  - Profession: accountant
  - Profession: violinist
  - Profession: egraber@allegheny.edu …. which attribute category to examine?

Designing a system

- Need an algorithm that does matching
  - need to know what the actual search term is
  - need to know what attribute to examine
    - **USER will specify the search term and attribute in a CLI**

- Data must be in a format that can searched
  - concept of a Matrix?



Designing a system

- Need an algorithm that does matching
  - need to know what the actual search term is
  - need to know what attribute to examine
    - **USER will specify the search term and attribute in a CLI**

- Data must be in a format that can searched easily
  - ~~**concept of a Matrix?**~~
  - Want to keep all the data belonging to one person together
    - recall concept of encapsulation!
  - make a custom abstract data type to store the data for each person
    - **class Person()**
    - attributes in the class can store values for each data attribute
    - …where in a class would something like ^^ happen?

Recall from Monday

How/where does the class **store** data?

```
class Train():
  """Abstract data type representing a train."""
  def __init__(self):
    self._cars = 0

  def add(self, num_new_cars: int):
    """Add cars to the train."""
    self._cars += num_new_cars

  def numcars(self):
    return self._cars

  def __repr__(self):
    return f"Train with {self._cars} cars"
```

Poll

https://forms.gle/g3ynuM8x9TPV42nKA

Designing a system

- Need an algorithm that does matching
  - **USER will specify the search term and attribute in a CLI**
- Data must be in a format that can searched easily
  - **class Person()**
  - constructor creates and fills attributes that store the values of the data attributes for a given person

```
class Person:
    """Define a Person class."""

    def __init__(self, name: str, residence: str, cell: str, profession: str, email: str) -> None:
        """Define the constructor for a person."""
        self.name = name
        self.residence = residence
        self.cell = cell
        self.profession = profession
        self.email = email
```

Compare…

Notice any conceptual differences?

What does this mean for practical use?

```
class Train():
  """Abstract data type representing a train."""
  def __init__(self):
    self._cars = 0

  def add(self, num_new_cars: int):
    """Add cars to the train."""
    self._cars += num_new_cars

  def numcars(self):
    return self._cars

  def __repr__(self):
    return f"Train with {self._cars} cars"
```

```
class Person:
    """Define a Person class."""

    def __init__(self, name: str, residence: str, cell: str, profession: str, email: str) -> None:
        """Define the constructor for a person."""
        self.name = name
        self.residence = residence
        self.cell = cell
        self.profession = profession
        self.email = email
```

Designing a system

- Need an algorithm that does matching
  - **USER will specify the search term and attribute in a CLI**
- Data must be in a format that can searched easily
  - **class Person()**
  - constructor creates and fills attributes that store the values of the data attributes for a given person
- Need to be able to be able to access all the data in order to pass it into the constructor
  - File I/O
  - not part of the class because it is needed to instantiate objects!

File I/O

Read the documentation on csvread

- https://docs.python.org/3/library/csv.html

Look at previous example in Integer Squaring

Designing a system

- Need an algorithm that does matching
  - **USER will specify the search term and attribute in a CLI**
- Data must be in a format that can searched easily
  - **class Person()**
  - constructor creates and fills attributes that store the values of the data attributes for a given person
- Need to be able to be able to access all the data in order to pass it into the constructor
  - File I/O
  - not part of the class because it is needed to instantiate objects!
- What methods would be useful for our original goal (helping friend with matchmaking service)?

```
def create_list(self) -> List[str]:
    """Create a list of strings representing the person."""
    details = []
    details.append(self.name)
    details.append(self.residence)
    details.append(self.cell)
    details.append(self.profession)
    details.append(self.email)
    return details
```

Designing a system

- Need an algorithm that does matching
  - **USER will specify the search term and attribute in a CLI**
- Data must be in a format that can searched easily
  - **class Person()**
  - constructor creates and fills attributes that store the values of the data attributes for a given person
- Need to be able to be able to access all the data in order to pass it into the constructor
  - File I/O
  - not part of the class because it is needed to instantiate objects!
- What methods would be useful for our original goal (helping friend with matchmaking service)?
  - making a simple list of one person's attributes
  - inside the class because it is handling object attributes

More on Matching

- Need an algorithm that does matching
  - **USER will specify the search term and attribute in a CLI**
- should matching occur inside or outside of the class?
  - remember, there are 50,000 people in the data file
  - Matching function outside of class
- Steps needed before matching
  - access the data using python
  - all data → person objects
  - all person objects → container
  - take **search term**, **attribute** (category), and **container** to find all matched persons

```
def find_matching_people(attribute: str, search_term: str, list_of_persons: List[Person]) -> List[Person]:
    list_of_matching_persons = []
    for current_person in list_of_persons:
      if search_term in getattr(current_person, attribute):
            list_of_matching_persons.append(current_person)
    return list_of_matching_persons
```

Once matches are found, then what?

- display the list of matching persons
  - do slight modification on ^^ for prettiness and human readability

```
def create_display_text(list_of_persons: List[Person]) -> str:
    """Convert list of persons into a string with pretty formatting."""
    display_text = ""
    for current_person in list_of_persons:
      display_text += "- " + str(current_person) + "\n"
    return display_text
```

Colab