# Page Class

The Page object represents a Wikipedia page. Below is a detailed description of the attributes and methods.

Attributes

* String url
  + stores the url of this wiki page
* String name
  + is the string of the link clicked on to get to **this** Page
* String name
  + Stores the actual title of the wiki page
* Document doc
  + Stores the document corresponding to the url
* HashMap<String, String> linkMap
  + Key: link name
  + Value: url associated with link name
* HashSet<Page> outgoingPages
  + Set containing all page objects created from **this** Page
* HashMap<String, Page> neighborPages
  + Key: url
  + Value: Page object created from the url
* HashSet<Page> outgoingPages
  + Set containing all outgoing urls on **this** Page
* Page parent
  + Points to the Parent page that contains a link to **this** Page

Methods

* **public** Page(String linkText, String url)
  + Creates a Page object based on the link name and the url
* **public** Page(String name)
  + Creates a Page object based on the name passed in. We set the name and create the corresponding url.
* **public** **static** String createUrl(String name)
  + Creates the corresponding wiki url for a wiki page name
  + ex: "Cologne" becomes https://en.wikipedia.org/wiki/Cologne

# Racer Class

The Racer class creates a Racer object. In this class we actual perform the BFS and print the results of the “wiki race”. Below is a detailed description of the attributes and methods.

Attributes

* Page start
  + The start page object
* HashSet<String> discoveredUrls
  + Set containing all of the wikipages (by url) that have been discovered
* String startUrl
  + The url for the starting wiki page
* String endUrl
  + The url for the ending wiki page
* String startName
  + The name of the starting wiki page
* String startName
  + The name of the ending wiki page

Methods

* **public** Racer(String start, String end)
  + Creates a racer object based on the names of the starting and ending wiki pages
* **public** **void** bfs()
  + Performs a modified version of the BFS algorithm we learned in class.
  + Here we create a graph as we go, since we don’t have a graph of all of Wikipedia. The “graph” that we make isn’t a real graph. When we discover a new wiki page, we update a parent pointer so that we can traverse the path once BFS is over.
  + Here is an overview of this modified version of BFS
    - We create a queue of Pages
    - We add the start Page to the queue
    - While the queue isn’t empty and we haven’t found a Page with the endUrl, we pop a page from the queue (currPage). If its url matches the endUrl we end BFS. Else, we set the document for the current page and set its neighbors.
* **public** **boolean** setDoc(Page p)
  + Creates a Document object for a Page object
  + If the document loaded successfully, we return true. Else we return false.
* **public** **void** setTitle(Page p)
  + Sets the title parameter for a Page object
* **public** **boolean** getNeighbors(Page p)
  + If the inputted Page p was null we return false
  + Else we return true after the following:
  + Here we get all of the neighbors for a Page object. We do this by selecting all of the wiki links on the current wiki page. We then create a Page object for each link. For each Page object created, we set its parent to be the current Page (ie. p). We then set the parameters outgoingPages, neighborPages, outgoingUrls accordingly.
* **public** **void** printRace(Page ending)
  + Here we print out the results of the race via the parent pointers.
  + We also print out the size of the shortest path
* **public** **void** printSize()
  + Here we print out the total number of wiki pages visited from BFS

# Tester Class

Here we test the BFS method. I have pasted the test results here because it takes a while to run BFS (These four tests took my computer 10 minutes to run! ). As you can see below, the tests pass.

As an aside, performing BFS on Wikipedia can be heap exhaustive. If there is no path or the heap has simply run out of memory the program will throw the following error:

Exception in thread "main" java.lang.OutOfMemoryError: Java heap space

We had to manually increase the heap size of the java project to actually run some of the tests. We set the heap to have 2 GB of space.

Testing Output

starting race from Joe Stephenson to Bloomington, Minnesota

::::::::::::results::::::::::::::

page title: Joe Stephenson

page url: https://en.wikipedia.org/wiki/Joe\_Stephenson

> click on link: New York Giants

page title: History of the New York Giants (baseball)

page url: https://en.wikipedia.org/wiki/New\_York\_Giants\_(NL)

> click on link: Bloomington, Minnesota

page title: Bloomington, Minnesota

page url: https://en.wikipedia.org/wiki/Bloomington,\_Minnesota

shortest path size: 2

total num of wiki pages visited: 33

-----------------------------------------------------------------------

-----------------------------------------------------------------------

starting race from Billie Eilish to Michael Jackson

::::::::::::results::::::::::::::

page title: Billie Eilish

page url: https://en.wikipedia.org/wiki/Billie\_Eilish

> click on link: Michael Jackson

page title: Michael Jackson

page url: https://en.wikipedia.org/wiki/Michael\_Jackson

shortest path size: 1

total num of wiki pages visited: 1

-----------------------------------------------------------------------

-----------------------------------------------------------------------

starting race from Earthquake to Cologne

::::::::::::results::::::::::::::

page title: Earthquake

page url: https://en.wikipedia.org/wiki/Earthquake

> click on link: Firestorm

page title: Firestorm

page url: https://en.wikipedia.org/wiki/Firestorm

> click on link: Cologne

page title: Cologne

page url: https://en.wikipedia.org/wiki/Cologne

shortest path size: 2

total num of wiki pages visited: 154

-----------------------------------------------------------------------

-----------------------------------------------------------------------

starting race from Jeremy Bentham to Hydrochloric acid

::::::::::::results::::::::::::::

page title: Jeremy Bentham

page url: https://en.wikipedia.org/wiki/Jeremy\_Bentham

> click on link: Pain in invertebrates

page title: Pain in invertebrates

page url: https://en.wikipedia.org/wiki/Pain\_in\_invertebrates

> click on link: hydrochloric acid

page title: Hydrochloric acid

page url: https://en.wikipedia.org/wiki/Hydrochloric\_acid

shortest path size: 2

total num of wiki pages visited: 421

-----------------------------------------------------------------------

-----------------------------------------------------------------------