

1. Why do we have an abstract PageParser class, and why do we have a pure virtual parse function inside the PageParser class?

PageParser is just a parent class for more specific parsers (ie. MDParse and TXTParse) so it is best to have it remain abstract. Accordingly, while every derived class of PageParser will have a parse() function, each child class will create it differently. For example, the MDParse needs to deal with search terms and links, while the TXTParse only handles search terms. This is why the parse() function is pure virtual in PageParser.

2. Why does the Handler class have a pure virtual process function?

Similarly to PageParser, the Handler class will have several child/derived classes. Although all derived classes will have the process() function, each derived class (ie QuitHandler, PrintHandler, AND, etc) will have its own implementation of it, since each handles a different command.

3. Consider the class hierarch/organization and list the the sequence of class function calls (i.e. who calls who) that will result from an AND query (e.g. AND term1 term2), starting from the SearchUI::run() until the results are computed and displayed.

1. SearchUI::run()
2. Handler::handle()
3. (Handler) AND::canHandle()
4. (Handler) AND::process()
5. SearchEng::search()
6. (WebPageSetCombiner) AND::combine()

