

YourGPSApp

parserdemo::  
ParserDemoApp

+ main(String[]): void

creates and  
passes to Parser  
constructor

creates and passes  
to Parser constructor

creates and calls

parser::Parser

- parserEventHandler: AbstractParserEventHandler = null  
- xmlReader: XMLReader  
  
- convertToFileURL(String): String  
+ parse(String): void  
+ Parser(AbstractParserEventHandler)

calls overridden  
methods of

Do not modify blue  
classes

Prints XML info to console

parserdemo::ElementAnnouncer

+ characters(char[], int, int): void  
+ endDocument(): void  
+ endElement(String, String, String): void  
+ startElement(String, String, String, Attributes): void

studentroster::RosterBuilder

- currentState: PossibleStates = PossibleStates....  
- currentStudent: Student  
- name: String  
- roster: List<Student>  
  
+ characters(char[], int, int): void  
+ endDocument(): void  
+ endElement(String, String, String): void  
+ getName(): String  
+ getRoster(): List<Student>  
+ RosterBuilder()  
+ startElement(String, String, String, Attributes): void

borrow code from this  
example

ContentHandler  
ErrorHandler  
LexicalHandler

parser::AbstractParserEventHandler

- currentTag: String  
# locator: Locator  
- logEnabled: boolean = true  
  
+ characters(char[], int, int): void  
+ comment(char[], int, int): void  
+ enableLogging(boolean): void  
+ endCDATA(): void  
+ endDocument(): void  
+ endDTD(): void  
+ endElement(String, String, String): void  
+ endEntity(String): void  
+ endPrefixMapping(String): void  
+ error(SAXParseException): void  
+ fatalError(SAXParseException): void  
+ getColumn(): int  
+ getLine(): int  
+ ignorableWhitespace(char[], int, int): void  
# log(String, String): void  
+ processingInstruction(String, String): void  
+ setDocumentLocator(Locator): void  
+ skippedEntity(String): void  
+ startCDATA(): void  
+ startDocument(): void  
+ startDTD(String, String, String): void  
+ startElement(String, String, String, Attributes): void  
+ startEntity(String): void  
+ startPrefixMapping(String, String): void  
+ warning(SAXParseException): void

Serializable  
studentroster::Student

- firstname: String  
- id: int  
- lastname: String  
- program: String  
  
0..\*  
+ getId(): int  
+ getProgram(): String  
+ setId(int): void  
+ setProgram(String): void  
+ Student(String, String)  
+ toString(): String

trackbuilder::GPSTrackBuilder

- currentState: PossibleStates = PossibleStates....  
  
+ characters(char[], int, int): void  
+ endDocument(): void  
+ endElement(String, String, String): void  
+ startElement(String, String, String, Attributes): void

creates  
from  
GPX file

You need to  
design/complete these  
for the GPS app

Track

has many

GPSCoordinate