Класс TheoProject.

TeoProject инкапсулирует данные в списке llTeoStations и методы для обработки, уравнивания и получения результатов в виде различных файлов.

Методы TeoProject:

* loadPolList(String pathWorkDir) – выбрать файл \*.pol на локальном диске используя MyChooser, извлечь имя файла и изменить залоловок MainFrame, заполнить llStations, новыми Station;
* writePol – записывает на диск текущий TheoProject
* getListPol – закрытый метод. Возвращает LinkedList<String> , содержащий элементы – строки файла Pol
* getTheoStation(int idx) – возвращает ссылку на объект класса TheoStation по индексу;
* getSizeTheoStations – возвращает int размер списка сранций llTheoStations
* getAbsolutePolPath – возвращает String значение закрытой переменной absolutePolPath;
* getBindType – возвращает перечислимый тип привязки (BindType):
  + ZZ – неизвестный тип привязки
  + TT – ход, опирающийся на твёрдые стороны в начале и конце
  + TO – ход, опирающийся на твёрдую сторону в начале и твёрдую точку в конце
  + OT – ход, опирающийся на твёрдую точку в начале и твёрдую сторону в конце
  + OO – ход, опирающийся на твёрдые пункты в начале и в конце (координатная привязка)
  + TZ – ход, опирающийся на твёрдую сторону только в начале (висячий ход)
  + ZT – ход, опирающийся на твёрдую сторону только в конце (висячий ход)
* setDirections – определяет дирекционные углы в указанном диапазоне и направлении.
* iniDDs – инициализирует значения поправок ddHor, ddX, ddY, ddZ для всего theoProject
* perimeter – определяет сумму сторон в указанном диапазоне
* setDDHors – рассчитывает угловую невязку и определяет поправки к горизонтальным углам
* setDXDY – определяет приращения координат dX, dY в указанном диапазоне
* setXYZ – определяет координаты в указанном диапазоне
* setDDXY – расчитывает линейные невязки fX, fY и определяет поправки ddX, ddY (соответствующие поля llTheoStations.get(i))
* setDDZ – расчитывает высотную невязку fZ и определяет поправки ddZ

(LinkedList<TeoStation> llTheoStations. Это упорядоченный набор измерений, необходимый для получения координат определяемых пунктов методом полигонометрии и тригонометрического нивелирования.

Класс TheoStation.

Класс TheoStation - включает координаты исходных пунктов и результаты линейно угловых наблюдений в виде левых горизонтальных углов между направлениями, горизонтальных проложений и превышений между пунктами.

TeoStation инкапсулирует исходные данные для каждой станции тахеометрического хода в закрытых переменных и предоставляет доступ к ним через открытые методы.

Класс ExtractProject.

Класс ExtractProject наследует LinkedList<ExtractStation> инкапсулирует необходимые для TheoProject данные, извлечённые из SurveyProject и предоставляет открытые методы для создания TheoProject и анализа точности измерений, использованных для реализации этой задачи

Класс ExtractStation.

Класс ExtractStation включает извлечённые из SurveyProject данные и предоставляет методы для получения производных от них величин средних значений и расхождений между двумя измерениями.

Класс ShowAbout.

Класс ShowAbout наследует JDialog предоставляет пользователю текст лицензионного соглашения и информацию о версии программы.

Класс ShowHelp.

Класс ShowHelp наследует JFrame предоставляет пользователю встроенную справочную систему программы Taheoport. Навигация по справочной системе, реализованной в виде набора панелей JPanel осуществляется через обработку событий дерева JTree. В зависимости от выбранного узла этого дерева отображается соответствующая панель в панели содержимого через объект CardLayout.

Класс Shell.

Класс Shell определяет внешний вид графического интерфейса пользователя, в том числе язык интерфейса и ведомостей в соответствии с установками пользователя.

getTitles – возвращает HashMap<String>, содержащий строки для текстовых элементов в соответствии со значением panentFrame.getOptions.getLanguage.

getTopReportXY, getTopReportZ, getTopReportExtract, getTopReportSurvey – возвращают LinkedList<String>, содержащих строки верхнего колонтитула ведомостей вычисления координат и высот точек полигона и ведомости вычисления координат съёмочных пикетов в соответствии со значением panentFrame.getOptions.getLanguage.

getTitlesReportXY, getTitlesReportZ, getTitlesReportExtract – возвращают HashMap<String>, содержащий строки для оформления отчётов в соответствии со значением panentFrame.getOptions.getLanguage.

Уравнивание полигонометрического хода.

Уравнивание хода, опирающегося на две твёрдые стороны:

* Определить индексы наибольшей и наименьшей сторон;
* Определить дирекционные углы первой и последней сторон;
* Определить величину угловой невязки;
* Определить поправки для каждого измеренного угла и сравнить сумму поправок с невязкой ;
* Определить дирекционные углы сторон и приращения координат, учитывая поправки в углы;
* Определить величины линейных невязок;
* Определить поправки к приращениям координат пропорционально длинам сторон и сравнить суммы поправок с невязками по соответствующим осям. В случае расхождения вследствие округления, добавить разницу в сторону с наибольшей стороной;
* Определить координаты X и Y определяемых пунктов, учитывая поправки к приращениям.

Уравнивание хода, опирающегося на твёрдую сторону в начале и твёрдый пункт в конце:

* Определить индексы наибольшей и наименьшей сторон;
* Определить дирекционный угол первой (твёрдой) стороны;
* Определить дирекционные углы сторон и приращения координат;
* Определить предварительные координаты конечного пункта;
* Определить величины линейных невязок сопоставив предварительные и теоретические (каталожные) координаты конечного пункта;
* Определить поправки к приращениям координат пропорционально длинам сторон и сравнить суммы поправок с невязками по соответствующим осям. В случае расхождения вследствие округления, добавить разницу в сторону с наибольшей стороной;
* Определить координаты X и Y определяемых пунктов, учитывая поправки к приращениям.

Уравнивание хода, опирающегося на конечные твёрдые пункты (координатная привязка):

* Определить индексы наибольшей и наименьшей сторон;
* Определить дирекционный угол первой стороны хода равным 0.0000;
* Определить дирекционные углы сторон и приращения координат;
* Определить предварительные координаты конечного пункта;
* Определить дирекционный угол направления с начального пункта на конечный, используя предварительные координаты последнего;
* Определить дирекционный угол направления с начального пункта на конечный, используя теоретические (каталожные) координаты последнего;
* Определить угол разворота как разницу вышеупомянутых дирекционных углов.
* Определить дирекционный угол первой стороны хода равным углу разворота;
* Определить дирекционные углы сторон и приращения координат;
* Определить предварительные координаты конечного пункта;
* Определить величины линейных невязок сопоставив предварительные и теоретические (каталожные) координаты конечного пункта;
* Определить поправки к приращениям координат пропорционально длинам сторон и сравнить суммы поправок с невязками по соответствующим осям. В случае расхождения вследствие округления, добавить разницу в сторону с наибольшей стороной;
* Определить координаты X и Y определяемых пунктов, учитывая поправки к приращениям.

Уравнивание хода, опирающегося на твёрдую сторону в начале хода (висячий ход):

* Определить дирекционный угол первой (твёрдой) стороны;
* Определить дирекционные углы сторон и приращения координат;
* Определить координаты конечного пункта;

Уравнивание нивелирования.

* Определить величину высотной невязки;
* Определить поправки в превышения пропорционально длинам сторон и сравнить сумму с невязкой. В случае расхождения вследствие округления, добавить разницу в сторону с наибольшей стороной;
* Определить координаты Z для определяемых пунктов.