```
from datetime import datetime
import pandas as pd

from extractors.direct_downloads import extract_with_request
from models.registry import Registry
from utils.helpers import clean_str, date_to_iso


class KOR(Registry):
    def __init__(self, **kwargs):
        """
        Child container for extracting, transforming, and loading registry data for South Korea

        :param kwargs: see models.registry for details
        """

        [illegible]

        # assign constant values for final registry
        [illegible]
        [illegible]
        [illegible]
        [illegible]

        # assign aircrafts input data source and filename
        [illegible]
        [illegible]
        # assign lightweight aircrafts input data source and filename
        [illegible]
        [illegible]

        # assign registry input data extraction method
        [illegible]
        # assign registry input data reading method
        [illegible]
        # assign registry data wrangling method
        [illegible]

        # extract, transform, and load registry data
        [illegible]

    def extract_registry(self):
        """
        Extracts registry input data from remote source

        :return: effect - creates [DATA_DIR|TEST_DIR]/[REGISTRY_INPUT_DIR]/[self.country_code]/[self.aircrafts_filename] file;
            effect - creates [DATA_DIR|TEST_DIR]/[REGISTRY_INPUT_DIR]/[self.country_code]/[self.lightweight_aircrafts_filename] file
        """

        # extract aircrafts input data
        [illegible]
        [illegible]
        [illegible]

        # extract lightweight aircrafts input data
        [illegible]
        [illegible]
        [illegible]
                                                    [illegible]

        # read registry input data
        [illegible]

    def read_registry(self, aircrafts=None, lightweight_aircrafts=None):
        """
        Reads registry input data

        :param (list) aircrafts: set of key-value pairs representing extracted aircrafts input data
        :param (list) lightweight_aircrafts: key-value pairs representing extracted lightweight aircrafts input data
        :return: effect - modifies self.registry
        """

        [illegible]
            # handle when aircrafts and lightweight aircrafts input data has been extracted but not in current session
            [illegible]
            [illegible]
            [illegible]
            [illegible]
        [illegible]
            [illegible]
            [illegible]

        # assign registry input data
        [illegible]

    def wrangle_registry(self):
        """
        Transforms registry data

        :return: effect - modifies self.wrangled_records
        """

        # loop through raw registrations
        [illegible]
            # initialize storage for wrangled representation of registration in focus
            [illegible]
                [illegible]
                [illegible]
                [illegible]
                [illegible]
                [illegible]
            [illegible]

            # update list of transformed values to populate final registry
            [illegible]

    @staticmethod
    def __get_mark(mark):
        """
        Formats registration mark by inserting a hyphen after country prefix

        :param (str) mark: registration mark to hyphenate
        :return: formatted copy of 'mark' argument
        """

        [illegible]

        [illegible]

    @staticmethod
    def __get_year(date):
        """
        Extracts the year from a date

        :param (str) date: registration date to get year from
        :return: year value from 'date' argument
        """

        [illegible]

        [illegible]
            [illegible]

        [illegible]
        [illegible]

        [illegible]
```