```
from PyQt5 import QtDBus
from PyQt5.QtCore import QVariant
from vorta.keyring.abc import VortaKeyring


class VortaKWallet5Keyring(VortaKeyring):
    """A wrapper for the qtdbus package to support the custom keyring backend"""

    folder_name = 'Vorta'
    service_name = "org.kde.kwalletd5"
    object_path = "/modules/kwalletd5"
    interface_name = 'org.kde.KWallet'

    def __init__(self):
        """
        Test whether DBus and KDEWallet are available.
        """
        self.iface = QtDBus.QDBusInterface(
            self.service_name,
            self.object_path,
            self.interface_name,
            QtDBus.QDBusConnection.sessionBus())
        if not (self.iface.isValid() and self.get_result("isEnabled")):
            raise KWalletNotAvailableException

    def set_password(self, service, repo_url, password):
        self.get_result("writePassword", args=[self.handle, self.folder_name,
repo_url, password, service])

    def get_password(self, service, repo_url):
        if not (self.is_unlocked and self.get_result("hasEntry",
                                                     args=[self.handle,
self.folder_name, repo_url, service])):
            return None
        return self.get_result("readPassword", args=[self.handle,
self.folder_name, repo_url, service])

    def get_result(self, method, args=[]):
        if args:
            result = self.iface.callWithArgumentList(QtDBus.QDBus.AutoDetect,
method, args)
        else:
            result = self.iface.call(QtDBus.QDBus.AutoDetect, method)
        return result.arguments()[0]

    @property
    def is_unlocked(self):
        self.try_unlock()
        return self.handle > 0

    def try_unlock(self):
        wallet_name = self.get_result("networkWallet")
        wId = QVariant(0)
        wId.convert(4)
        output = self.get_result("open", args=[wallet_name, wId, 'vorta-repo'])
        self.handle = int(output)


class KWalletNotAvailableException(Exception):
    pass
```