```
import types
import inspect
import account

def wormhole(f):
  def wrapper(*args):
    atm_id = inspect.currentframe().f_back.f_locals[’self’].idn
    print("## At the ATM{0} has been requested a «{1}»
          on the account {2} owned by {3} for {4}€.".
          format(atm_id, f.__name__, args[0].number, args[0].owner, args[1]))
    return f(*args)
  return wrapper

excluded = [’__init__’, ’balance’]

class WormholeSetup(type):
  def __new__(meta, cls, supers, classdict):
    for name, elem in classdict.items():
      if (type(elem) is types.FunctionType) and (name not in excluded):
        classdict[name] = wormhole(elem)
    return type.__new__(meta, cls, supers, classdict)

Account = WormholeSetup(’Account’, (), dict(account.Account.__dict__))
```