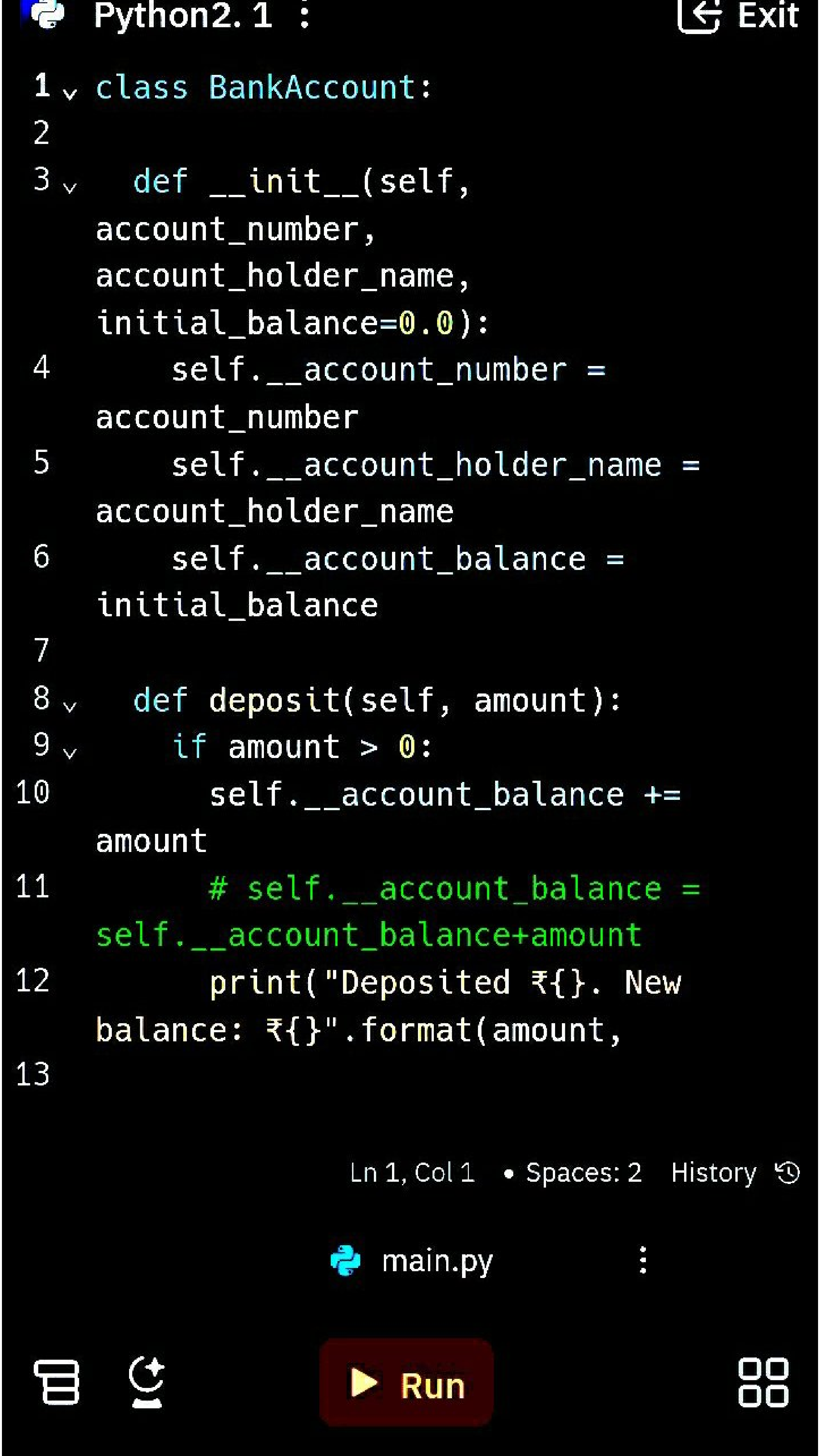
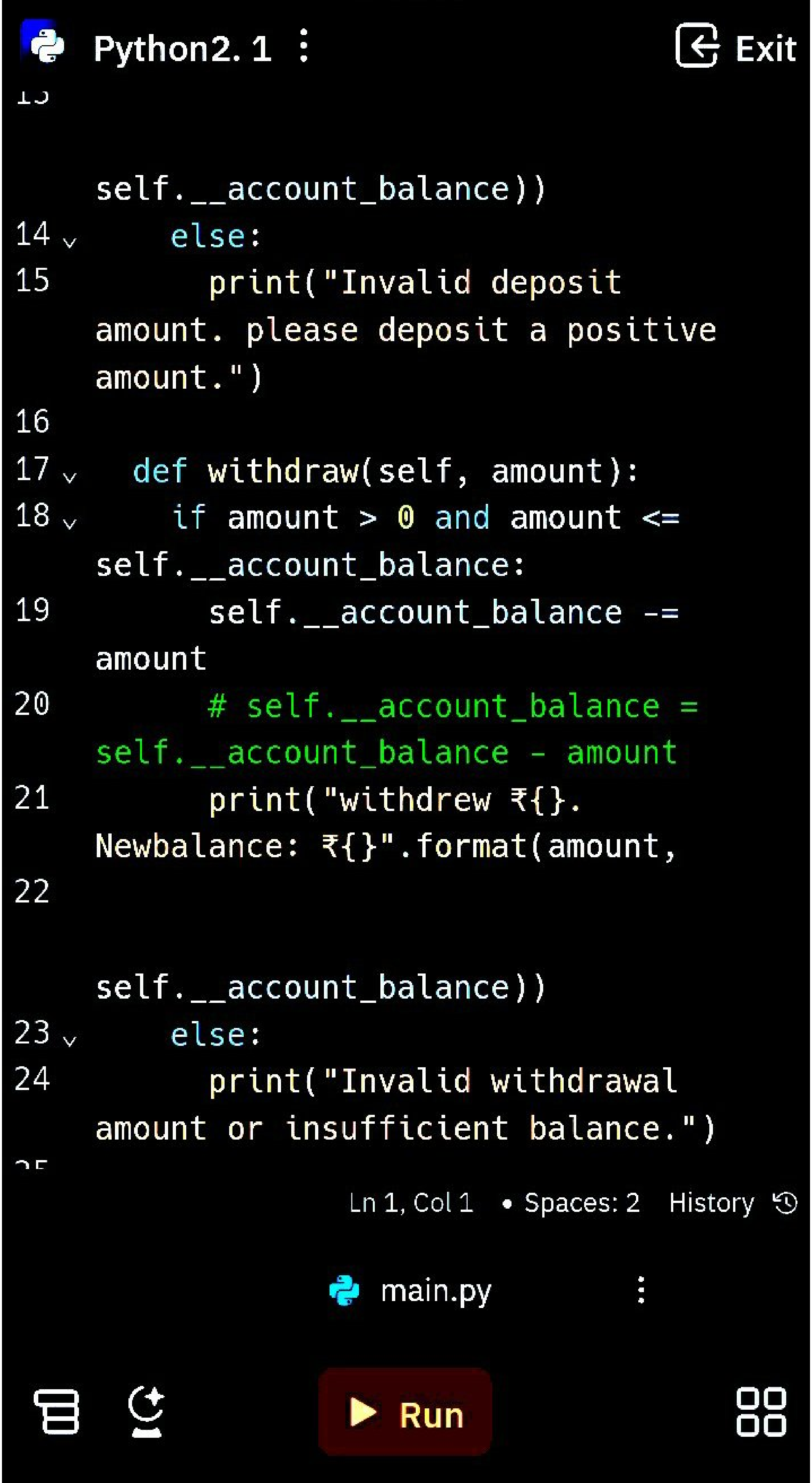
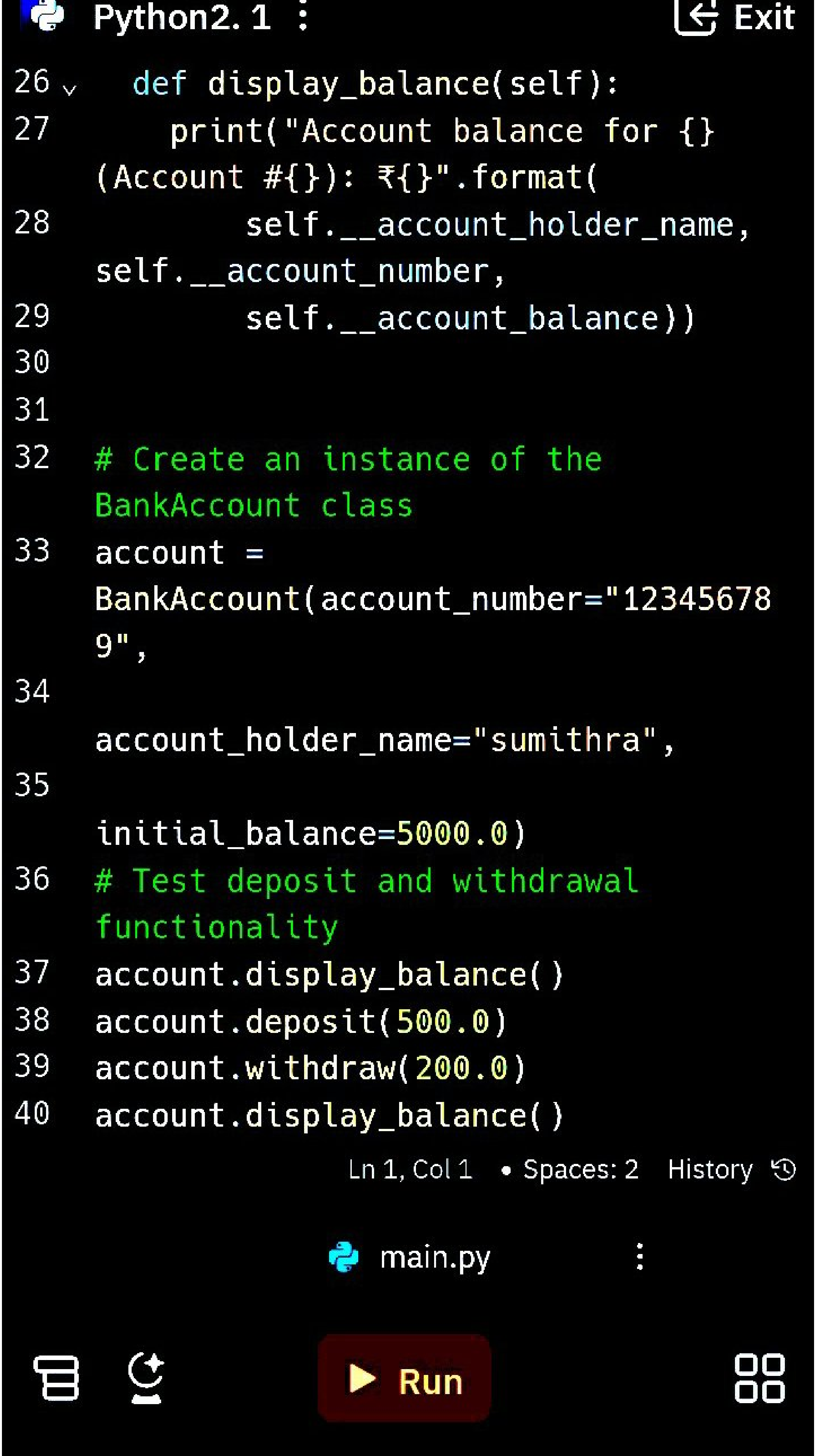


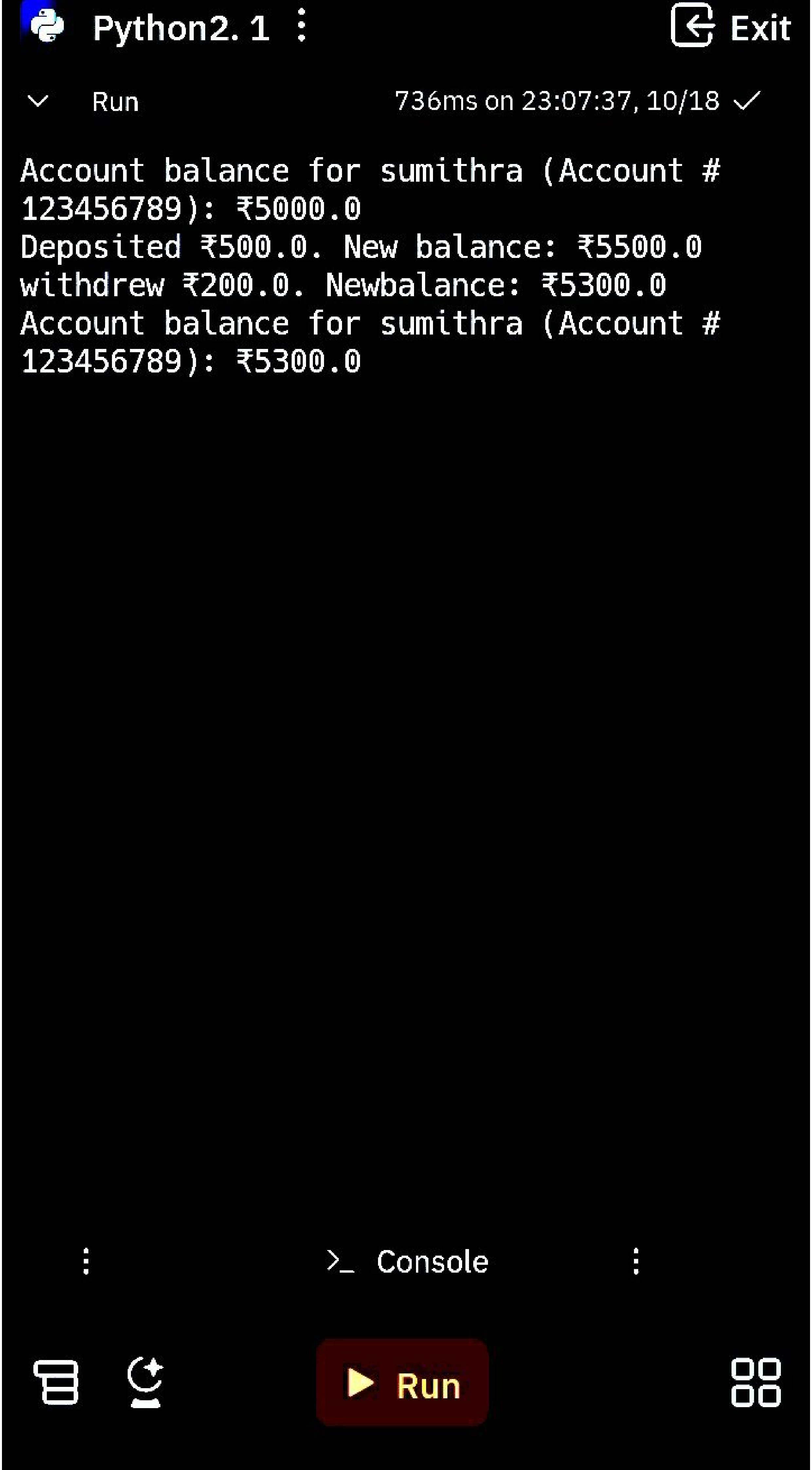
```python
class BankAccount:

  def __init__(self, account_number, account_holder_name, initial_balance=0.0):
    self.__account_number = account_number
    self.__account_holder_name = account_holder_name
    self.__account_balance = initial_balance

  def deposit(self, amount):
    if amount > 0:
      self.__account_balance += amount
      # self.__account_balance = self.__account_balance+amount
      print("Deposited ₹{}. New balance: ₹{}".format(amount,

```



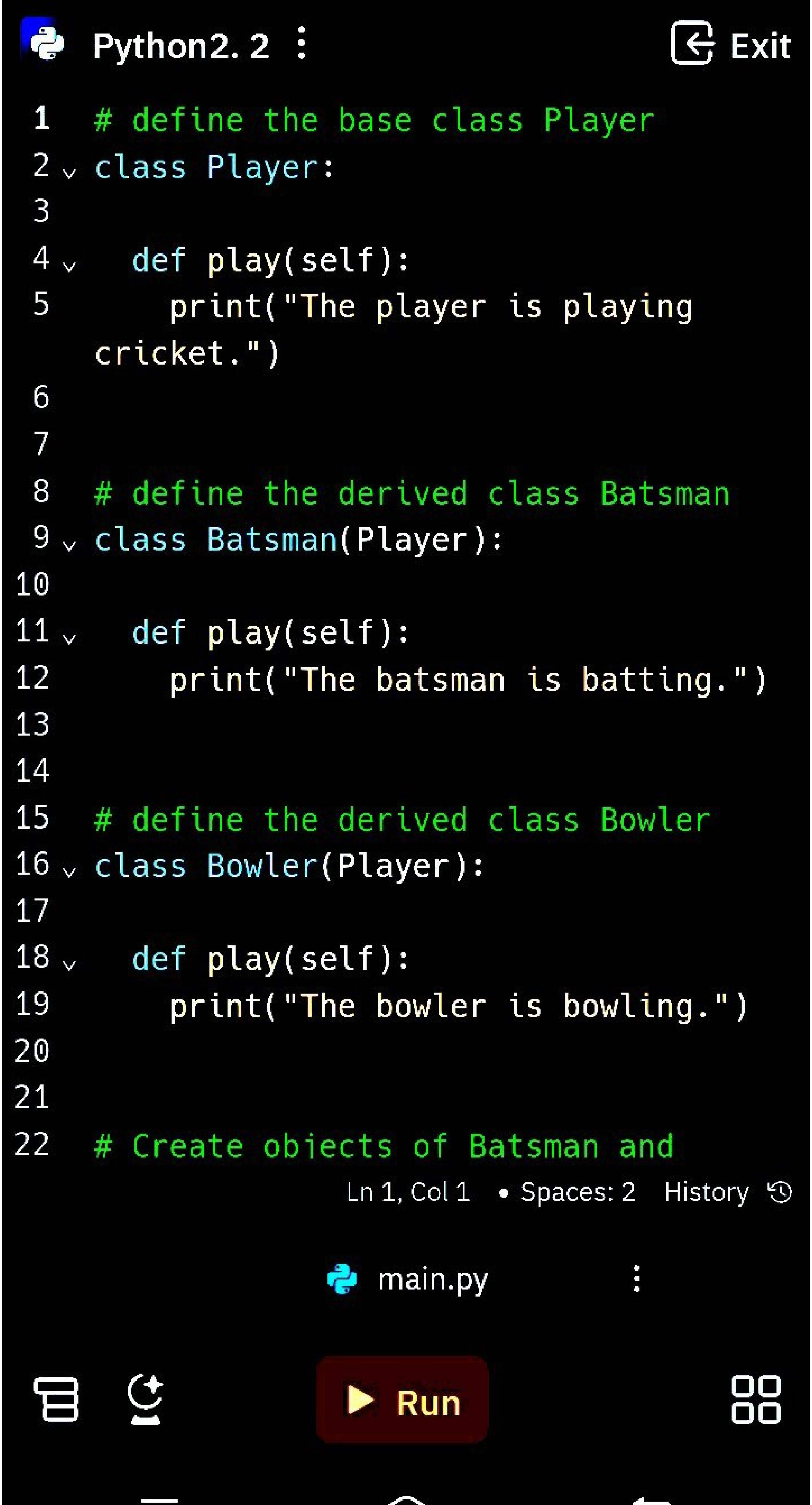
```
        self.__account_balance))
    else:
      print("Invalid deposit amount. please deposit a positive amount.")

  def withdraw(self, amount):
    if amount > 0 and amount <= self.__account_balance:
      self.__account_balance -= amount
      # self.__account_balance = self.__account_balance - amount
      print("withdrew ₹{}. Newbalance: ₹{}".format(amount,
        self.__account_balance))
    else:
      print("Invalid withdrawal amount or insufficient balance.")
```



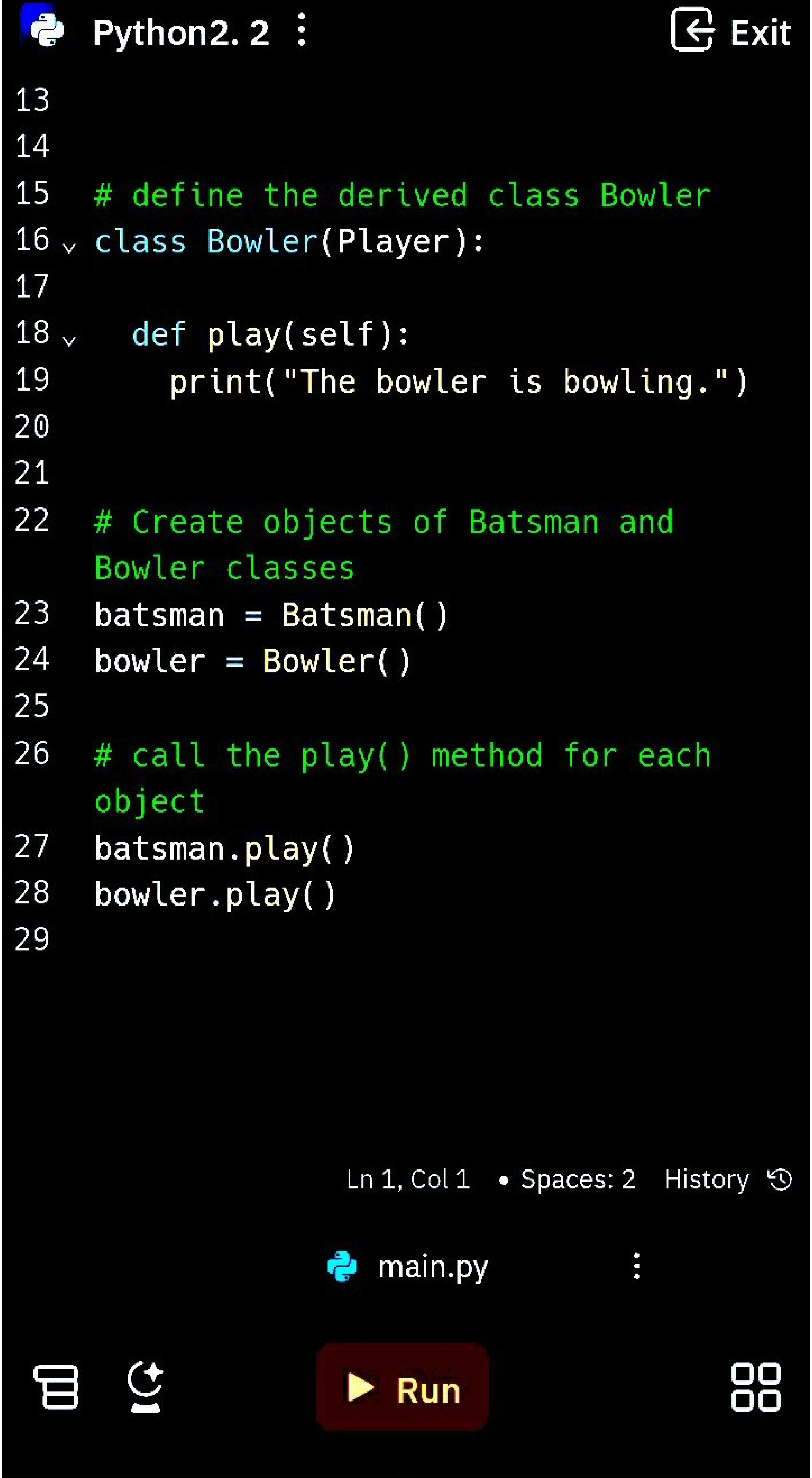
```
  def display_balance(self):
    print("Account balance for {} (Account #{}): ₹{}".format(
        self.__account_holder_name, self.__account_number,
        self.__account_balance))


# Create an instance of the BankAccount class
account = BankAccount(account_number="123456789",
                      account_holder_name="sumithra",
                      initial_balance=5000.0)
# Test deposit and withdrawal functionality
account.display_balance()
account.deposit(500.0)
account.withdraw(200.0)
account.display_balance()
```



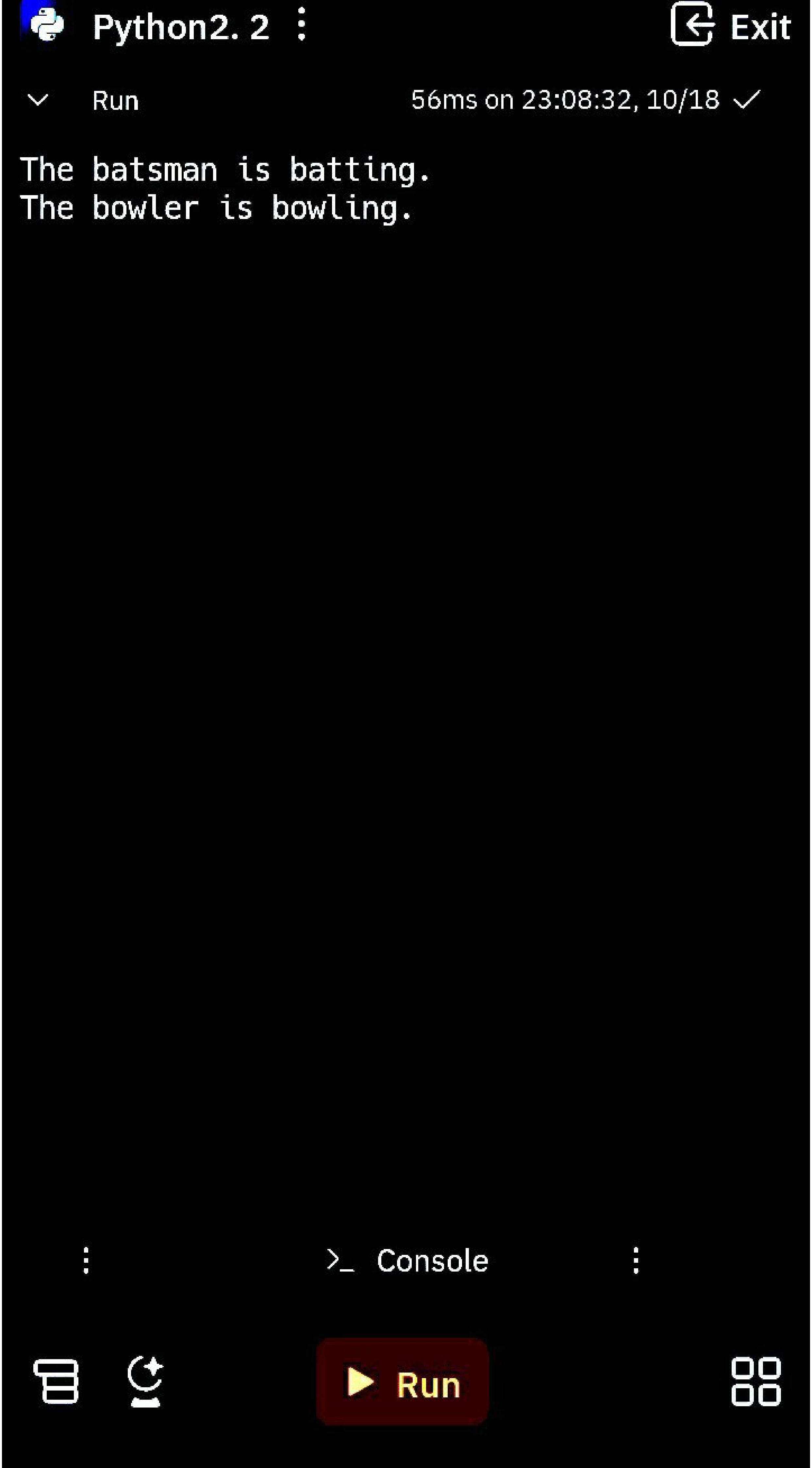
```
Account balance for sumithra (Account #
123456789): ₹5000.0
Deposited ₹500.0. New balance: ₹5500.0
withdrew ₹200.0. Newbalance: ₹5300.0
Account balance for sumithra (Account #
123456789): ₹5300.0
```



```python
# define the base class Player
class Player:

  def play(self):
    print("The player is playing cricket.")


# define the derived class Batsman
class Batsman(Player):

  def play(self):
    print("The batsman is batting.")


# define the derived class Bowler
class Bowler(Player):

  def play(self):
    print("The bowler is bowling.")


# Create objects of Batsman and
```



```python


# define the derived class Bowler
class Bowler(Player):

  def play(self):
    print("The bowler is bowling.")


# Create objects of Batsman and Bowler classes
batsman = Batsman()
bowler = Bowler()

# call the play() method for each object
batsman.play()
bowler.play()

```



```
The batsman is batting.
The bowler is bowling.
```