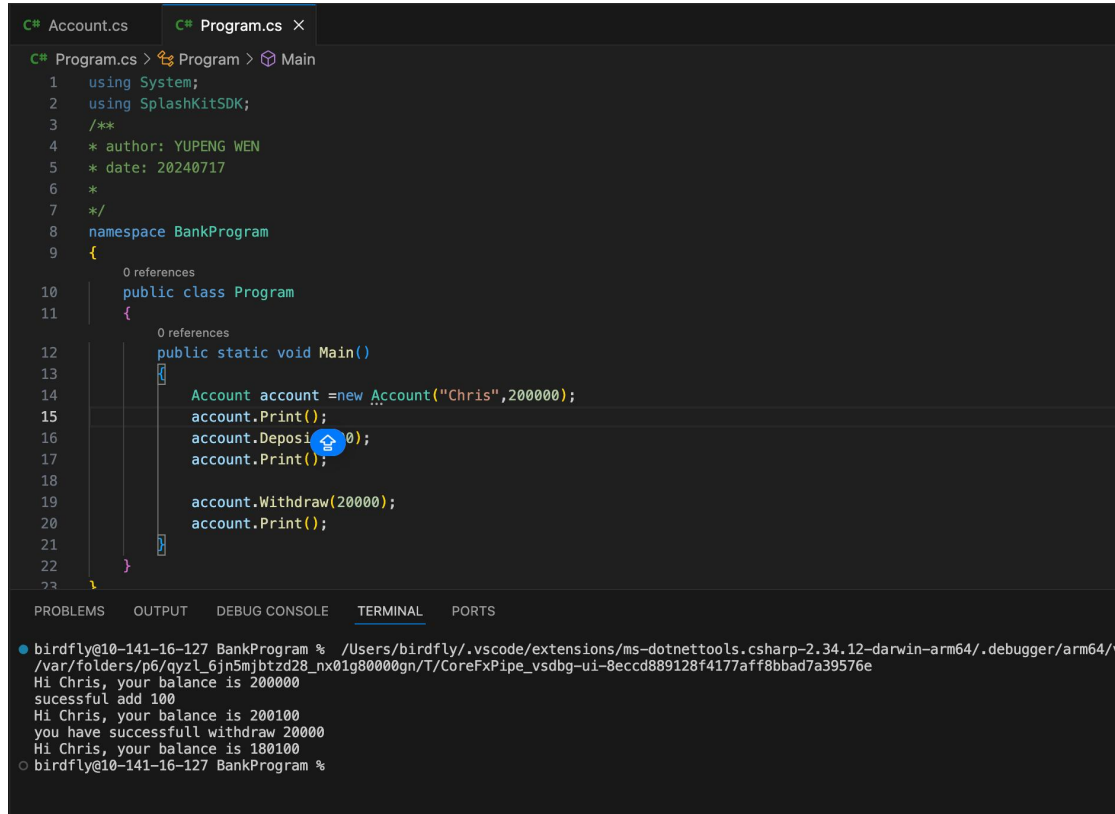


# Result



The screenshot shows a Visual Studio Code editor with two tabs: 'Account.cs' and 'Program.cs'. The 'Program.cs' tab is active, displaying a C# program. The code defines a 'BankProgram' namespace containing a 'Program' class with a 'Main' method. The 'Main' method creates an 'Account' object named 'account' with the name 'Chris' and an initial balance of 200,000. It then prints the balance, deposits 100, prints the balance again, and finally withdraws 200,000, printing the balance one last time. The terminal at the bottom shows the output of the program, which matches the code's logic: it prints 'Hi Chris, your balance is 200000', 'successful add 100', 'Hi Chris, your balance is 200100', 'you have successfull withdraw 20000', and 'Hi Chris, your balance is 180100'.

```
C# Account.cs C# Program.cs X
C# Program.cs > Program > Main
1 using System;
2 using SplashKitSDK;
3 /**
4  * author: YUPENG WEN
5  * date: 20240717
6  */
7 */
8 namespace BankProgram
9 {
10     0 references
11     public class Program
12     {
13         0 references
14         public static void Main()
15         {
16             Account account =new Account("Chris",200000);
17             account.Print();
18             account.Deposit(100);
19             account.Print();
20             account.Withdraw(20000);
21             account.Print();
22         }
23     }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
birdfly@10-141-16-127 BankProgram % /Users/birdfly/.vscode/extensions/ms-dotnettools.csharp-2.34.12-darwin-arm64/.debugger/arm64/
/var/folders/p6/qyzl_6jn5mjbtd28_nx01g80000gn/T/CoreFxPipe_vsdg-ui-8eccd889128f4177aff8bbad7a39576e
Hi Chris, your balance is 200000
successful add 100
Hi Chris, your balance is 200100
you have successfull withdraw 20000
Hi Chris, your balance is 180100
birdfly@10-141-16-127 BankProgram %
```

## Q&A

How classes are used to define objects.

class as a template to create object, class describe what fields do object have and what things can object do.

How methods, fields, and properties all work together when you create a class

when create a class, we have to think, which fields are static, and which fields are non-static. Certainly, the same as methods. These static fields and methods can be called by class. And all the non-static fields only can be accessed after create an object.

How fields give knowledge to each object created from the class

When create object, all the fields will be initial, just like when you declare an int field, it will init to an zero.

How methods give capabilities to each object created from the class

method can update the fields or modify other object.