Task 13: Collection

1. From the previous task change the HMBank attribute Accounts to List of Accounts and perform the same operation.

2. From the previous task change the HMBank attribute Accounts to Set of Accounts and perform the same operation.

• Avoid adding duplicate Account object to the set.

• Create Comparator<Account> object to sort the accounts based on customer name when listAccounts() method called.

3. From the previous task change the HMBank attribute Accounts to HashMap of Accounts and perform the same operation.

`Customer.cs`

csharp
CopyEdit

```
using System;using System.Text.RegularExpressions;
namespace task13.entity
{
   public class Customer
   {
      public long CustomerId { get; set; }
      public string FirstName { get; set; }
      public string LastName { get; set; }

      private string email;
      public string Email
      {
         get => email;
         set
         {
            if (!Regex.IsMatch(value, @"^[^@\s]+@[^@\s]+\.[^@\s]+$"))
               throw new ArgumentException("Invalid email format.");
            email = value;
         }
      }

      private string phone;
      public string Phone
      {
         get => phone;
         set
         {
            if (!Regex.IsMatch(value, @"^\d{10}$"))
               throw new ArgumentException("Phone number must be 10 digits.");
            phone = value;
         }
```

```
        }

        public string Address { get; set; }

        public Customer() { }

        public Customer(long id, string fname, string lname, string email, string phone,
string address)
        {
            CustomerId = id;
            FirstName = fname;
            LastName = lname;
            Email = email;
            Phone = phone;
            Address = address;
        }

        public override string ToString()
        {
            return $"{CustomerId} - {FirstName} {LastName} | {Email} | {Phone} |
{Address}";
        }
    }
}
```

Account.cs

```
namespace task13.entity
{
    public class Account
    {
        public long AccountNumber { get; set; }
        public string AccountType { get; set; }
        public double Balance { get; private set; }
        public Customer Customer { get; set; }

        public Account() { }

        public Account(long accNo, string accType, double balance, Customer customer)
        {
            AccountNumber = accNo;
            AccountType = accType;
            Balance = balance;
            Customer = customer;
        }
```

```
        public void Deposit(double amount)
        {
            if (amount > 0)
                Balance += amount;
        }

        public void Withdraw(double amount)
        {
            if (amount > Balance)
                throw new Exception("Insufficient funds.");
            Balance -= amount;
        }

        public void CalculateInterest()
        {
            if (AccountType.ToLower() == "savings")
                Balance += Balance * 0.045;
        }

        public override string ToString()
        {
            return $"{AccountNumber} - {AccountType} | Balance: {Balance:C} | Customer:
{Customer.FirstName}";
        }

        public override bool Equals(object obj)
        {
            return obj is Account acc && acc.AccountNumber == this.AccountNumber;
        }

        public override int GetHashCode()
        {
            return AccountNumber.GetHashCode();
        }
    }
}
```

HMBank_List.cs

```
using System.Collections.Generic;
using task13.entity;
using System.Linq;

namespace task13.service
{
    public class HMBank_List
    {
        private List<Account> accounts = new List<Account>();
```

```
        private long nextAccountNumber = 1001;

        public Account CreateAccount(string accType, double balance, Customer
customer)
        {
            var acc = new Account(nextAccountNumber++, accType, balance, customer);
            accounts.Add(acc);
            return acc;
        }

        public List<Account> ListAccounts()
        {
            return accounts;
        }
    }
}
```

HMBank_Set.cs

```
using System.Collections.Generic;
using task13.entity;
using System.Linq;

namespace task13.service
{
    public class HMBank_Set
    {
        private HashSet<Account> accounts = new HashSet<Account>();
        private long nextAccountNumber = 2001;

        public Account CreateAccount(string accType, double balance, Customer
customer)
        {
            var acc = new Account(nextAccountNumber++, accType, balance, customer);
            accounts.Add(acc);
            return acc;
        }

        public List<Account> ListSortedAccounts()
        {
            return accounts.OrderBy(a => a.Customer.FirstName).ToList();
        }
    }
}
```