

Java实验报告

@Author : liyajun_208111545116

实验一

搭建环境并输出*Hello World*

```
1 public static void main(String[] args){  
2     // write your code here  
3     System.out.println("Hello World");  
4 }
```

```
.jar=55238:C:\Users\liyaj\AppData\Local\Jet  
demo.demo01  
Hello World  
  
Process finished with exit code 0
```

实验二

回文数

H5 代码实现

```
1  boolean valid(int i) {
2      String s = Integer.toString(i);
3      int l = 0, r = s.length() - 1;
4      while(l < r) {
5          if(s.toCharArray()[l++] != s.toCharArray()[r--])
6              return false;
7      }
8      return true;
9  }
```

H5 结果

```
1  12121 -> true;
2  114514 -> false;
```

猜数字

H5 代码实现

```
1  public void guess(){
2      Random ra = new Random();
3      int cur = ra.nextInt(100) + 1;
4      System.out.println("please enter a num");
5      Scanner sc = new Scanner(System.in);
6      int get = sc.nextInt();
7      while(get != cur){
8          if(get > cur) System.out.println("Try a smaller
9          number");
10         else if(get < cur) System.out.println("Try a larger
11         number");
12         get = sc.nextInt();
13     }
14     System.out.println("congratulations");
15 }
```

H5 结果

```
.jar=56111.C:\Users\liyaj\AppData\Local\JetBrains\Toolbox
demo.demo01
please enter a num
50
Try a larger number
85
Try a smaller number
65
Try a larger number
75
congratulations

Process finished with exit code 0
```

完数

H5 代码实现

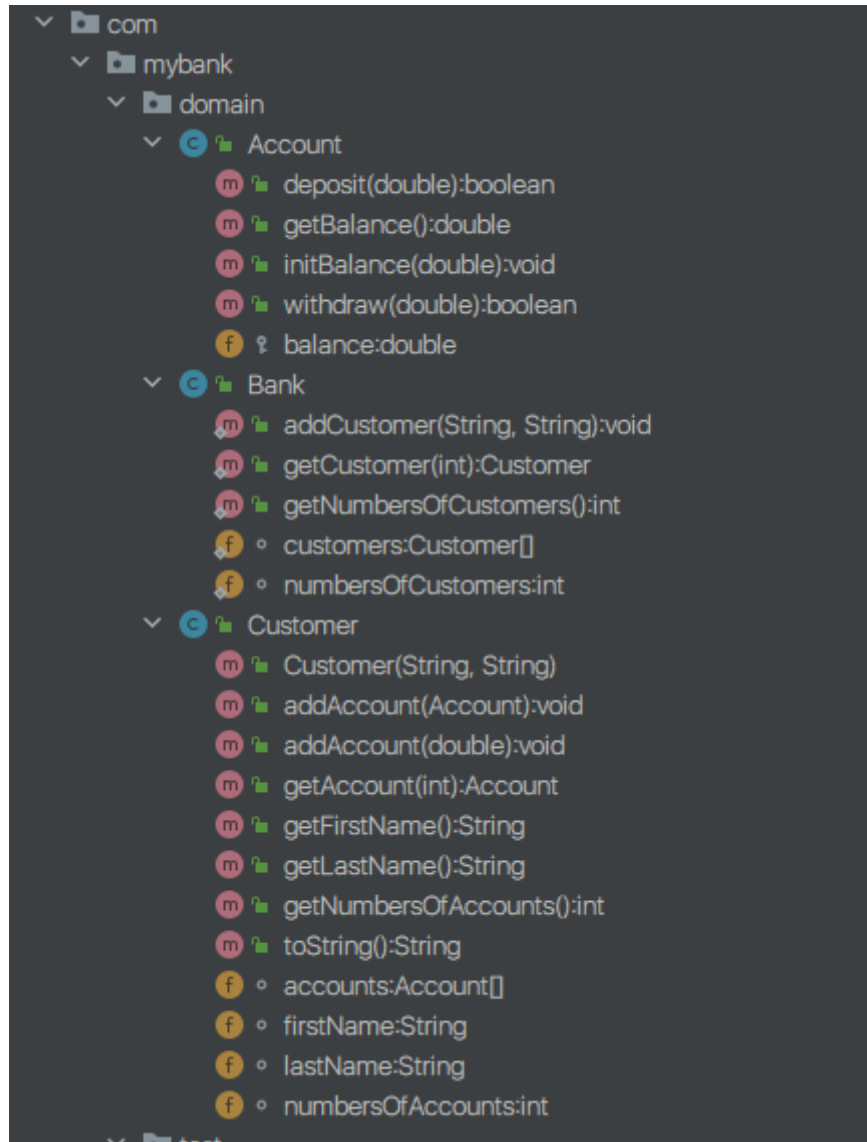
```
1 public static void main(String[] args){
2     // write your code here
3     for(int i = 1; i <= 1000; i++){
4         if(i % 2 == 0 && valid(i)) {
5             System.out.println(i);
6         }
7     }
8 }
9 boolean valid(int x) {
10     int cur = x;
11     for(int i = 1; i <= x/2; i++) {
12         if(x % i == 0) cur -= i;
13     }
14     return cur == 0;
15 }
```

H5 运行结果

```
6
28
496
```

实验三

项目结构



实现代码

H5 Account.java

```
1 package com.mybank.domain;
2
3 public class Account {
4     protected double balance;
5     //初始化余额
6     public Account(double balance){
```

```

7         this.balance = balance;
8     }
9     //存钱
10    public boolean deposit(double amt){
11        if(amt < 0) return false;
12        balance += amt;
13        return true;
14    }
15    //取钱
16    public boolean withdraw(double amt){
17        if(balance < amt) return false;
18        balance -= amt;
19        return true;
20    }
21
22    //获取余额
23    public double getBalance(){
24        return balance;
25    }
26 }
27

```

H5 Bank.java

```

1    package com.mybank.domain;
2
3    public class Bank {
4        static Customer [] customers;
5        static int numbersOfCustomers;
6        static {
7            customers = new Customer[10];
8            numbersOfCustomers = 0;
9        }
10       public static void addCustomer(String firstName, String
lastName){
11           Customer cus = new Customer(firstName,lastName);
12           try {
13               customers[numbersOfCustomers++] = cus;
14           }catch (ArrayIndexOutOfBoundsException e){
15               System.out.println(e);
16           }
17       }

```

```

18     public static int getNumbersOfCustomers(){
19         return numbersOfCustomers;
20     }
21     public static Customer getCustomer(int id){
22         try {
23             return customers[id];
24         } catch (ArrayIndexOutOfBoundsException e){
25             System.out.println(e);
26         }
27         return null;
28     }
29 }
30

```

H5 Customer.java

```

1  package com.mybank.domain;
2
3  public class Customer {
4      String firstName;
5      String lastName;
6      Account [] accounts;
7      int numbersOfAccounts;
8
9      public Customer(String firstName, String lastName) {
10         this.firstName = firstName;
11         this.lastName = lastName;
12         numbersOfAccounts = 0;
13         accounts = new Account[10];
14     }
15     public void addAccount(double amount){
16         addAccount(new Account(amount));
17     }
18     public void addAccount(Account acc){
19         try {
20             accounts[numbersOfAccounts++] = acc;
21         } catch (ArrayIndexOutOfBoundsException e){
22             System.out.println(e);
23         }
24     }
25     public Account getAccount(int id){
26         try {

```

```

27         return accounts[id];
28     }catch (ArrayIndexOutOfBoundsException e){
29         System.out.println(e);
30     }
31     return null;
32 }
33 public int getNumbersOfAccounts() {
34     return numbersOfAccounts;
35 }
36 public String getFirstName() {
37     return firstName;
38 }
39
40 public String getLastName() {
41     return lastName;
42 }
43
44 @Override
45 public String toString() {
46     return firstName + "-" + lastName ;
47 }
48 }
49

```

H5 TestBanking.java

```

1  package com.mybank.test;
2
3  import com.mybank.domain.Bank;
4
5  import java.util.Random;
6
7  public class TestBanking {
8      public static void main(String[] args) {
9          //初始化成员
10         for(int i = 0; i < 10; i++){
11             Random ra = new Random();
12             Bank.addCustomer(Integer.toString(i), "Jinx");
13             for(int j = 0; j < 10; j++){
14
15                 Bank.getCustomer(i).addAccount(ra.nextDouble(1145.14));
16             }
17         }
18     }
19 }

```

```
15     Bank.getCustomer(i).getAccount(j).withdraw(ra.nextDouble(1145
16         .14));
17     }
18     }
19     for(int i = 0; i < 10; i++){
20
21         System.out.println(Bank.getCustomer(i).toString());
22         for(int j = 0; j < 10; j++){
23             System.out.println("第" + (j + 1) + "个账户的余额
24             为");
25
26             System.out.println(String.format("%.2f", Bank.getCustomer(i).
27             getAccount(j).getBalance()));
28         }
29     }
30 }
```

测试结果


```

C:\Users\liyaj\.jdk\openjdk-17.0.2\bin\java.exe -javaagent:C:\Users\liyaj\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch
.jar=57769:C:\Users\liyaj\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch-0\213.7172.25\bin -Dfile.encoding=UTF-8 -classp
com.mybank.test.TestBanking
0-Jinx
第1个账户的余额为
72.67
第2个账户的余额为
184.98
第3个账户的余额为
30.48
第4个账户的余额为
57.66
第5个账户的余额为
721.69
第6个账户的余额为
191.60
第7个账户的余额为
764.83
第8个账户的余额为
454.07
第9个账户的余额为
223.04
第10个账户的余额为
22.55
1-Jinx
第1个账户的余额为
320.37
第2个账户的余额为
625.96
第3个账户的余额为
23.68
第4个账户的余额为
187.50
第5个账户的余额为
123.28
第6个账户的余额为
700.84
第7个账户的余额为
42.48

```

实验四

带有存款和贷款的银行系统

H5 CheakAccount.java

```

1 package experiment.four.com.mybank.domain;
2 //能够透支的账户
3 public class CheckingAccount extends Account{
4     double overdraftAmount;
5
6     public CheckingAccount(double initBalance, double
overdraftAmount){
7         super(initBalance);
8         this.overdraftAmount = overdraftAmount;
9     }
10    CheckingAccount(double initBalance){
11        super(initBalance);

```

```

12         this.overdraftAmount = 0.0;
13     }
14
15     public double getOverdraftAmount() {
16         return overdraftAmount;
17     }
18
19     @Override
20     public int withdraw(double amount) {
21         if(balance < amount){
22             //使用贷款
23             double overdraftNeeded = amount - balance;
24             if(overdraftNeeded > overdraftAmount){
25                 return 1;
26             }
27             else{
28                 balance = 0.0;
29                 overdraftAmount -= overdraftNeeded;
30                 return 2;
31             }
32         } // 能够付清
33         else{
34             balance -= amount;
35         }
36         return 3;
37     }
38 }
39

```

SavingAccount.java

```

1 package experiment.four.com.mybank.domain;
2 //能计算利息的账户
3 public class SavingsAccount extends Account{
4     private double interestRate;
5     public SavingsAccount(double initBalance,double
6     interestRate){
7         super(initBalance);
8         this.interestRate = interestRate;
9     }
10 }

```

```

9
10
11     //计算利息
12     public double accumulateInterest(){
13         return balance + balance * interestRate;
14     }
15 }
16

```

H5 Test

```

1  package experiment.four.com.mybank.test;
2
3
4  import experiment.four.com.mybank.domain.*;
5
6
7  public class TestBanking2 {
8      public static void main(String[] args) {
9          Bank.addCustomer("zhang", "san");
10         Customer c1 = Bank.getCustomer(0);
11         c1.addAccount(new SavingsAccount(500,0.017));
12         c1.addAccount(new CheckingAccount(2000, 3000));
13         for(int i = 0;i < Bank.getNumbersOfCustomers();i++) {
14             Customer cus = Bank.getCustomer(i);
15             System.out.println(cus);
16             for(int j = 0;j < cus.getNumbersOfAccounts();j++)
17         {
18                 Account acc = cus.getAccount(j);
19                 System.out.println("第" + (j+1) + "个账户");
20                 if(acc instanceof SavingsAccount sa){
21                     System.out.printf("计算余额
22     %.2f\n", sa.accumulateInterest());
23                 }
24                 else if(acc instanceof CheckingAccount ck){
25                     System.out.printf("当前余额
26     为%.2f\n", ck.getBalance());
27                     System.out.printf("能够透支的钱数
28     %.2f\n", ck.getOverdraftAmount());
29                     ck.withdraw(2004);
30                     System.out.println("取了2004元后");
31                 }
32             }
33         }
34     }
35 }
36

```

```

27         System.out.printf("下次还能透支的额度
    %.2f", ck.getOverdraftAmount());
28     }
29     else {
30         //普通账户
31         System.out.printf("余额为
    %.2f", acc.getBalance());
32     }
33 }
34 }
35 }
36 }
37

```

测试

```

C:\Users\liyaj\.jdk\openjdk-17.0.2\bin\java.exe -javaagent:C:\Users\liyaj\AppData\Local\JetBrains\Toolbox\app
.jar=64838:C:\Users\liyaj\AppData\Local\JetBrains\Toolbox\app
experiment.four.com.mybank.test.TestBanking2
zhang-san
第1个账户
计算余额 508.50
第2个账户
当前余额为2000.00
能够透支的钱数 3000.00
取了2004元后
下次还能透支的额度 2996.00
Process finished with exit code 0
|

```

实验五

动物宠物程序

实现代码

H5 Animal.java

```
1 package experiment.five.InterfaceProject;
2
3 public abstract class Animal {
4     protected int legs;
5     protected Animal(int legs){
6         this.legs = legs;
7     }
8     abstract void eat();
9     public void walk(){
10         System.out.println("Animal walk with"+legs+"legs.");
11     }
12 }
13
```

H5 Cat.java

```
1 package experiment.five.InterfaceProject;
2
3 public class Cat extends Animal implements Pet{
4     private String name;
5     public Cat(String name){
6         super(4);
7         this.name = name;
8     }
9     public Cat(){
10         this("");
11     }
12     @Override
13     void eat() {
14         System.out.println("Cat likes eating mice.");
15     }
16
17     @Override
18     public String getName() {
```

```

19         return this.name;
20     }
21
22     @Override
23     public void setName(String n) {
24         this.name = name;
25     }
26
27     @Override
28     public void play() {
29         System.out.println("Cat likes playing strings.");
30     }
31 }

```

H5 Duck.java

```

1  package experiment.five.InterfaceProject;
2
3  public class Duck extends Animal implements Pet, Sound{
4      private String name;
5
6      public Duck( String name) {
7          super(2);
8          this.name = name;
9      }
10
11     @Override
12     public void walk() {
13         System.out.println("duck swim and walks
with"+legs+"legs");
14     }
15
16     @Override
17     void eat() {
18         System.out.println("duck like eating waterweeds");
19     }
20
21     @Override
22     public String getName() {
23         return this.name;
24     }
25

```

```

26     @Override
27     public void setName(String n) {
28         this.name = name;
29     }
30
31     @Override
32     public void play() {
33         System.out.println("duck likes playing with water");
34     }
35     public void shout(String voice){
36         Sound.super.shout(voice);
37         System.out.println("i am a yellow duck");
38     }
39 }

```

H5 Fish.java

```

1  package experiment.five.InterfaceProject;
2
3  public class Fish extends Animal implements Pet{
4      private String name;
5      @Override
6      void eat() {
7          System.out.println("Fish likes eating bugs in the
8 ponds");
9      }
10
11     public Fish() {
12         super(0);
13     }
14
15     @Override
16     public String getName() {
17         return this.name;
18     }
19
20     @Override
21     public void setName(String n) {
22         this.name = n;
23     }
24
25     @Override

```

```

25     public void play() {
26         System.out.println("fish likes swimming in yje
    ponds");
27     }
28     @Override
29     public void walk(){
30         super.walk();
31         System.out.println("fish are swimming happily in the
    pool.");
32     }
33 }

```

H5 Pet.interface

```

1  package experiment.five.InterfaceProject;
2
3  public interface Pet {
4      public String getName();
5      public void setName(String n);
6      public void play();
7  }

```

H5 Sound.interface

```

1  package experiment.five.InterfaceProject;
2
3  public interface Sound {
4      public default void shout(String voice){
5          System.out.println(voice);
6      }
7  }

```

H5 Spider.java


```
1 package experiment.five.InterfaceProject;
2
3 public class Spider extends Animal{
4     public Spider(){
5         super(8);
6     }
7
8     @Override
9     void eat() {
10         System.out.println("Spider eats flying on the net");
11     }
12 }
```

H5 Test

```
1 package experiment.five.InterfaceProject;
2
3 public class TestAnimal {
4     public static void main(String[] args) {
5         Spider s1 = new Spider();
6         s1.eat();
7         Animal s2 = new Cat();
8         System.out.println(s2.legs);
9         s2.walk();
10        s2.eat();
11        Pet s3 = new Cat("fluffy");
12        s3.getName();
13        s3.play();
14        Cat s4 = new Cat("Grafield");
15        s4.getName();
16        int legs = s4.legs;
17        s4.eat();
18        s4.walk();
19        s4.play();
20        Pet s5 = new Fish();
21        s5.getName();
22        Duck s6 = new Duck("donald duck");
23        s6.shout("gagaga");
24    }
25 }
```

测试

```
C:\Users\liyaj\.jdk\openjdk-17.0.2\bin\java.exe -java
.jar=53542:C:\Users\liyaj\AppData\Local\JetBrains\Tool
experiment.five.InterfaceProject.TestAnimal
Spider eats flying on the net
4
Animal walk with4legs.
Cat likes eating mice.
Cat likes playing strings.
Cat likes eating mice.
Animal walk with4legs.
Cat likes playing strings.
gagaga
i am a yellow duck
```

实验六

实验代码

H5 容器 *Bank*

```
1 package experiment.six.com.mybank.domain;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Bank {
7     static List<Customer> customers;
8     static {
9         customers = new ArrayList<>();
10    }
11    public static void addCustomer(String firstName, String
lastName){
12        customers.add(new Customer(firstName,lastName));
13    }
14    public static int getNumbersOfCustomers(){
```

```

15         return customers.size();
16     }
17     public static Customer getCustomer(int id){
18         try {
19             return customers.get(id);
20         } catch (ArrayIndexOutOfBoundsException e){
21             System.out.println(e);
22         }
23         return null;
24     }
25 }

```

H5 容器 *Customer*

```

1  package experiment.six.com.mybank.domain;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  public class Customer {
7      String firstName;
8      String lastName;
9      List<Account> accounts;
10     //Account[] accounts;
11     int numbersOfAccounts;
12
13     public Customer(String firstName, String lastName) {
14         this.firstName = firstName;
15         this.lastName = lastName;
16         accounts = new ArrayList<>();
17     }
18     public void addAccount(double amount){
19         addAccount(new Account(amount));
20     }
21     public void addAccount(Account acc){
22         accounts.add(acc);
23     }
24     public Account getAccount(int id){
25         try {
26             return accounts.get(id);
27         } catch (ArrayIndexOutOfBoundsException e){
28             System.out.println(e.getMessage());

```

```

29         return null;
30     }
31 }
32 public int getNumbersOfAccounts() {
33     return accounts.size();
34 }
35 public String getFirstName() {
36     return firstName;
37 }
38
39 public String getLastName() {
40     return lastName;
41 }
42
43 @Override
44 public String toString() {
45     return firstName + "-" + lastName ;
46 }
47 }

```

H5 异常类 *OverdraftException*

```

1 package experiment.six.com.mybank.domain;
2
3 /**
4  * @author liyajun
5  * @date 2022/4/6 14:40
6  */
7 public class OverdraftException extends Exception{
8     private double deficit;
9
10    public OverdraftException(String message,double deficit){
11        super(message);
12        this.deficit = deficit;
13    }
14
15    public double getDeficit() {
16        return deficit;
17    }
18 }

```

测试

```
1 package experiment.six.com.mybank.test;
2
3
4 import experiment.six.com.mybank.domain.*;
5
6
7 public class TestBanking2 {
8     public static void main(String[] args) {
9         Bank.addCustomer("zhang", "san");
10        Customer c1 = Bank.getCustomer(0);
11        c1.addAccount(new SavingsAccount(500,0.017));
12        c1.addAccount(new CheckingAccount(2000, 3000));
13        for(int i = 0; i < Bank.getNumbersOfCustomers(); i++)
14        {
15            Customer cus = Bank.getCustomer(i);
16            System.out.println(cus);
17            for(int j = 0; j < cus.getNumbersOfAccounts(); j++)
18            {
19                Account acc = cus.getAccount(j);
20                System.out.println("第" + (j+1) + "个账户");
21                if(acc instanceof SavingsAccount sa){
22                    System.out.printf("计算余额
23                    %.2f\n", sa.accumulateInterest());
24                }
25                else if(acc instanceof CheckingAccount ck){
26                    System.out.printf("当前余额
27                    为%.2f\n", ck.getBalance());
28                    System.out.printf("能够透支的钱数
29                    %.2f\n", ck.getOverdraftAmount());
30                    try {
31                        ck.withdraw(6000);
32                    } catch (OverdraftException e) {
33                        System.out.println(e.getMessage()+"赤字为"+e.getDeficit());
34                        e.printStackTrace();
35                    }
36                }
37            }
38        }
39    }
40}
```

```

34         try {
35             acc.withdraw(60000);
36         } catch (OverdraftException e) {
37             System.out.println(e.getMessage());
38             e.printStackTrace();
39         }
40         System.out.printf("余额为
%.2f", acc.getBalance());
41     }
42 }
43 }
44 }
45 }
46

```

```

C:\Users\liyaj\.jdk\openjdk-17.0.2\bin\java.exe -javaagent:C:\Users\liyaj\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch-0\213.7172.25\lib\idea_rt
.jar=64712:C:\Users\liyaj\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch-0\213.7172.25\bin -Dfile.encoding=UTF-8 -classpath C:\Users\liyaj\IdeaProjects\2022_Java\out\production\2022_Java
experiment.six.com.mybank.test.TestBanking2
zhang-san
第1个帐户
计算余额 500.50
第2个帐户
当前余额为2000.00
能够透支的数额 3000.00
透支额度不足赤字为4000.0
experiment.six.com.mybank.domain.OverdraftException Create breakpoint : 透支额度不足
    at experiment.six.com.mybank.domain.CheckingAccount.withdraw(CheckingAccount.java:26)
    at experiment.six.com.mybank.test.TestBanking2.main(TestBanking2.java:26)
Process finished with exit code 0

```