**ÔN TẬP**

**Câu 1** (1 điểm): Đảm bảo chất lượng phần mềm là gì? Thường gồm những hoạt động nào?

Trả lời:

là một quy trình hệ thống được sử dụng để đảm bảo rằng phần mềm phát triển đáp ứng các yêu cầu về chất lượng và tuân thủ các tiêu chuẩn, quy định và quy trình đề ra. Mục tiêu chính của SQA là cung cấp sản phẩm phần mềm chất lượng cao, giảm thiểu lỗi và nâng cao sự hài lòng của khách hàng.

**Các hoạt động thường gặp trong Đảm bảo chất lượng phần mềm:**

1. **Lập kế hoạch chất lượng (Quality Planning)**:
   * Xác định các tiêu chuẩn và quy trình cần tuân theo.
   * Thiết lập các mục tiêu chất lượng cụ thể cho dự án phần mềm.
2. **Quản lý chất lượng (Quality Management)**:
   * Đảm bảo rằng các hoạt động và kết quả của dự án tuân thủ các tiêu chuẩn và quy trình đã được xác định.
   * Theo dõi và đánh giá tiến độ thực hiện các hoạt động chất lượng.
3. **Kiểm thử phần mềm (Software Testing)**:
   * Thực hiện các loại kiểm thử như kiểm thử đơn vị (unit testing), kiểm thử tích hợp (integration testing), kiểm thử hệ thống (system testing) và kiểm thử chấp nhận (acceptance testing).
   * Xác định và sửa lỗi trong phần mềm.
4. **Đánh giá chất lượng (Quality Assessment)**:
   * Sử dụng các công cụ và kỹ thuật để đo lường và đánh giá chất lượng phần mềm.
   * Tiến hành các đánh giá chính thức (formal review) như kiểm tra mã nguồn (code review) và kiểm tra thiết kế (design review).
5. **Kiểm tra và đảm bảo quy trình (Process Assurance)**:
   * Kiểm tra các quy trình phát triển phần mềm để đảm bảo chúng được tuân thủ và hiệu quả.
   * Cải tiến liên tục các quy trình dựa trên phản hồi và kết quả thực tế.
6. **Quản lý cấu hình (Configuration Management)**:
   * Quản lý các thay đổi và phiên bản của phần mềm để đảm bảo tính nhất quán và tính toàn vẹn của sản phẩm.
7. **Đánh giá rủi ro (Risk Management)**:
   * Xác định, đánh giá và quản lý các rủi ro có thể ảnh hưởng đến chất lượng phần mềm.
   * Lên kế hoạch và thực hiện các biện pháp giảm thiểu rủi ro.

**Câu 2** (2 điểm): Xây dựng checklist GUI cho Hệ thống Quản lý tiền nước của thành phố.

**Câu 3** (3 điểm):

Biểu giá nước cho hộ gia đình dùng sinh hoạt cho ở bảng sau (mức sử dụng nước là số nguyên).

*Đơn vị tính đồng/m3*

|  |  |  |
| --- | --- | --- |
| **Số TT** | **Mức sử dụng nước sinh hoạt của hộ dân cư (m3/tháng/hộ gia đình)** | **Giá bán nước** |
| 1 | 10m3 đầu tiên x = 10 | 5 973 |
| 2 | Từ trên 10m3 đến 20m3 x = 20 , x = 11 | 7 052 |
| 3 | Từ trên 20m3 đến 30m3 x = 30, x = 21 | 8 669 |
| 4 | Trên 30m3 x = 31 | 15 929 |

Ngoài ra, khách hàng cần phải thanh toán 5% thuế suất GTGT, và 10% phí bảo vệ môi trường với nước thải sinh hoạt trên giá bán nước.

Hệ thống quản lý tiền nước của thành phố cần xuất hoá đơn cho khách hàng với thông tin về số nước và chi tiết tiền như bên dưới (trường hợp số lượng tiêu thụ >10m3 thì bổ xung thêm dòng chi tiết định mức tiêu thụ, đơn giá, thành tiền tương ứng):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Số đọc tháng này** | **Số đọc tháng trước** | **Số lượng tiêu thụ (m3)** | **Định mức tiêu thụ** | **Đơn giá (đ)** | **Thành tiền (đ)** |
| 452 | 445 | 17 |  |  |  |
|  |  |  | 10 | 5 973 | 59730 |
| CỘNG TIỀN HÀNG | | | | | 41 811 |
| Thuế suất GTGT 5% | | | | | 2 091 |
| Phí bảo vệ môi trường với NTSH 10% | | | | | 4 181 |
| Tổng cộng tiền thanh toán | | | | | 48 083 |

***Thực hiện thiết kế test case theo phương pháp Test biên và Test phân lớp tương đương.***

**Câu 4** (4 điểm):

Hệ thống quản lý tiền nước của thành phố cần phát triển module tính tiền nước như đặc tả ở Câu 3.

a. Thực hiện lập trình bằng Java

b. Thiết kế test case để đạt mức phủ cấp 2 (phủ nhánh) và vòng lặp (nếu có)

c. Viết code Junit tương ứng

Câu 2: checklist GUI

**Checklist cho GUI (Giao diện Người dùng)**

**Tổng quan**

1. Giao diện người dùng có đúng với prototype không?
2. Danh sách menu và màn hình có thống nhất không?
3. Các link HTML có hoạt động không?
4. Tất cả các cửa sổ đều có thể truy cập từ toolbar?
5. Tất cả các trang web/cửa sổ đều có thể truy cập từ menu?
6. Các màn hình được gọi từ button có hiển thị đúng không?
7. Các label, textbox, combobox, etc có đúng font chữ, cỡ chữ, màu chữ theo yêu cầu không?
8. Căn lề, độ rộng, khoảng cách có đồng bộ và đúng theo yêu cầu không?
9. Dữ liệu kiểu chữ căn lề trái?
10. Dữ liệu kiểu số căn lề phải?
11. Các form có được bố trí hợp lý và dễ sử dụng không?
12. Nếu các phím tắt được sử dụng, các phím tắt được gán có hoạt động đúng không?
13. Màn hình không có lỗi chính tả, cấu trúc câu, ngữ pháp?
14. Nếu các chữ viết tắt được sử dụng, nó có thống nhất trong tất cả các giao diện không? Người dùng có hiểu được không?
15. Định dạng số, ngày tháng, thời gian có nhất quán không?
16. Có hiển thị mô tả khi di chuyển chuột vào tooltip không?
17. Nếu có tooltip, nó có ý nghĩa hoặc có hữu ích không?
18. Có một button hoặc check box được xét mặc định chưa?
19. Con trỏ di chuyển lần lượt theo thứ tự: từ trái qua phải, từ trên xuống dưới khi nhấn Tab liên tục?
20. Con trỏ di chuyển lần lượt theo thứ tự: từ dưới lên trên, từ phải qua trái khi nhấn Shift-Tab?
21. Nếu chuột không focus vào button nào thì đã thực hiện chức năng của button chính khi nhấn Enter chưa?
22. Nếu đang focus vào 1 button thì đã thực hiện chức năng của button khi nhấn Enter chưa?
23. Màn hình thu nhỏ, phóng to tương ứng và không bị vỡ giao diện khi nhấn Ctrl - và Ctrl + không?
24. Các trường bắt buộc có dấu \* chưa?
25. Thông tin bản ghi hiển thị trên giao diện có đúng với thông tin bản ghi được lưu trong Database không?
26. Kiểm tra thay đổi ngôn ngữ, ngôn ngữ thay đổi có đúng nội dung không?

**Bố cục giao diện**

1. Kiểm tra title của màn hình.
2. Kiểm tra focus của chuột.
3. Kiểm tra hiển thị thông tin các trường và button trên màn hình.
   * Hiển thị title của chức năng.
   * Focus được set vào trường đầu tiên có thể edit.
   * Hiển thị đầy đủ các trường (Dropdownlist, Table, Button, Hyperlink).

**Kiểm tra chức năng**

1. Kiểm tra giao diện khi thu nhỏ, phóng to:
   * Nhấn phím Ctrl -.
   * Nhấn phim Ctrl +.
   * Màn hình thu nhỏ, phóng to tương ứng và không bị vỡ giao diện.
2. Kiểm tra thứ tự di chuyển trỏ trên màn hình khi nhấn phím Tab:
   * Focus vào màn hình. Nhấn Tab liên tục.
   * Con trỏ di chuyển lần lượt theo thứ tự: Từ phải qua trái, từ trên xuống dưới.
3. Kiểm tra thứ tự con trỏ di chuyển ngược lại trên màn hình khi nhấn Shift-Tab:
   * Focus vào màn hình. Nhấn phím Shift-Tab liên tục.
   * Con trỏ di chuyển lần lượt theo thứ tự: từ dưới lên trên, từ phải qua trái.
4. Kiểm tra thực hiện chức năng chính của màn hình khi nhấn Enter:
   * Nhấn phím Enter.
     + Nếu chuột không focus vào button nào thì Thực hiện chức năng của button chính.
     + Nếu đang focus vào 1 button thì sẽ thực hiện chức năng của button.

**Tổng thể giao diện**

1. Kiểm tra về bố cục, font chữ, chính tả, màu chữ.
   * Các label, textbox cùng font chữ cỡ chữ, căn lề trái, có độ dài, rộng và khoảng cách bằng nhau, không xô lệch.
   * Không có lỗi về chính tả, cấu trúc câu, ngữ pháp trên màn hình.
   * Form được bố trí hợp lý và dễ sử dụng.

Câu 3:

Phân tích giá trị biên:

+ bậc 1 : mức sử dụng nước tiêu thụ đầu tiên <= 10m3 -> Biên 0,10

+ bậc 2 : mức sử dụng nước tiêu thụ thứ hai từ 10 < mức tiêu thụ <= 20 -> biên 11,20

+ bậc 3: mức sử dụng nước tiêu thụ thứ ba từ 20 < mức tiêu thụ <= 30 -> biên 21, 30

+ bậc 4: mức sử dụng nước tiêu thụ thứ từ mức tiêu thụ > 30 -> biên 31

|  |  |  |
| --- | --- | --- |
| Biên | Input | Expected output |
| Biên bậc 1.1 | 0 | Tổng phí = 0 |
| Biên bậc 1.2 | 10 | Tổng mức tiêu thụ = 10 \* 5973 = 59730  Định mức tiêu thụ = 10, đơn giá= 5973, Thành tiền = 59730  Tổng phí = 59730 + 59730\*10% + 59730 \* 5%= 68690 |
| Biên bậc 2.1 | 11 | Tổng mức tiêu thụ = 10 \* 5973 + (11-10) \* 7052 = 66782  Định mức tiêu thụ1 = 10, đơn giá= 5973, Thành tiền = 59730  Định mức tiêu thụ2 = 11-10, đơn giá =7052, Thành tiền = 7052  Tổng phí = 66782 + 3340 + 6679 = 76801 |
| Biên bậc 2.2 | 20 | Tổng mức tiêu thụ = 10 \* 5973 + (20-10) \* 7052 = 130250  Định mức tiêu thụ1 = 10, đơn giá = 5973, thành tiền= 59730  Định mức tiêu thụ 2 = 20-10= 10, đơn giá = 7052, Thành tiền = 70520  Tổng phí = 130250 + 6513 + 13025 = 149788 |
| Biên bậc 3.1 | 21 | Tổng mức tiêu thụ = 10 \* 5973 + 10\* 7052 + 1\* 8669 = 138919  Định mức tiêu thụ1 = 10, đơn giá = 5973, thành tiền= 59730  Định mức tiêu thụ 2 = 10= 10, đơn giá = 7052, Thành tiền = 70520  Định mức tiê thụ 3 = 1, đơn giá = 8669, thành tiền = 8669  Tổng phí = 138919 + 6946 + 13892 = 159757 |
| Biên bậc 3.2 | 30 | Tổng mức tiêu thụ = 10 \* 5973 + 10\* 7052 + 1\* 8669 = 138919  Định mức tiêu thụ1 = 10, đơn giá = 5973, thành tiền= 59730  Định mức tiêu thụ 2 = 10= 10, đơn giá = 7052, Thành tiền = 70520  Định mức tiê thụ 3 = 10, đơn giá = 8669, thành tiền = 86690  Tổng phí = 138919 + 6946 + 13892 = 159757 |
| Biên bậc 4 | 31 | Tự tính |

Phân vùng tương đương

Valid:

+ bậc 1 : mức sử dụng nước tiêu thụ đầu tiên 0<= mức <= 10m3 -> mức = 10, mức = 0

+ bậc 2 : mức sử dụng nước tiêu thụ thứ hai từ 10 < mức tiêu thụ <= 20 -> mức = 18

+ bậc 3: mức sử dụng nước tiêu thụ thứ ba từ 20 < mức tiêu thụ <= 30 -> mức = 25

+ bậc 4: mức sử dụng nước tiêu thụ thứ từ mức tiêu thụ > 30 -> mức 50

* 4 testcase

Invalid:

+ mức tiêu thụ < 0 -> mức = -1

+ mức tiêu thụ không là số nguyên -> mức = 1.5

+ mức tiêu thụ input nhập vào không phải định dạng số -> mức abc

* 3 testcase

|  |  |  |
| --- | --- | --- |
| Vùng | Input | Expected output |
| Invalid | | |
| Invalid 1 | -1 | Mức tiêu thụ phải >= 0 |
| Invalid 2 | 1.5 | Mức tiêu thụ phải là một số nguyên |
| Invalid 3 | Abc | Mức tiêu thụ phải là định dạng số nguyên |
| Valid | | |
| Valid bậc 1.1 | 0 | Tính mức tiêu thụ bậc 1 |
| Valid bậc 1.2 | 10 | Tính mức tiêu thụ bậc 1 |
| Làm theo giống với biên nhưng viết ngắn gọn thôi | | |

package org.example;  
  
import java.util.Scanner;  
  
public class Tinhtiennuoc {  
 public Tinhtiennuoc() {  
 }  
  
 public int Tinhtiennuoc1(int muctieuthu)  
 {  
 int tong = 0;  
 int muc1;  
 int muc2;  
 int muc3;  
 int muc4;  
 System.*out*.println("muc tieu thu:" + muctieuthu);  
 if(muctieuthu < 0)  
 {  
 System.*out*.println("so khong hop le");  
 return -1;  
 }  
 else{  
 if( muctieuthu <=10)  
 {  
 muc1 = muctieuthu;  
 System.*out*.println("muc1: " + muc1 + ", " +"don gia: " + 5973 +" "+ "tong gia muc1: " + (muc1 \* 5973));  
 tong = muc1 \* 5973;  
 }  
 else if(muctieuthu>10 && muctieuthu<=20){  
 muc1 = 10;  
 System.*out*.println("muc1: " + muc1 + ", " +"don gia: " + 5973 +", "+ "tong gia muc1: " + (10 \* 5973));  
 muc2 = muctieuthu - 10;  
 System.*out*.println("muc2: " + muc2 + ", " +"don gia: " + 7052 +", "+ "tong gia muc2: " + (muc2 \* 7052));  
 tong = muc1\*5973 + muc2\*7052;  
 }  
 else if(muctieuthu>20 && muctieuthu<=30)  
 {  
 muc1 = 10;  
 System.*out*.println("muc1: " + muc1 + ", " +"don gia: " + 5973 +", "+ "tong gia muc1: " + (muc1 \* 7052));  
 muc2 = 10;  
 System.*out*.println("muc2: " + muc2 + ", " +"don gia: " + 7052 +", "+ "tong gia muc2: " + (muc2 \* 7052));  
 muc3 = muctieuthu - 20;  
 System.*out*.println("muc3: " + muc3 + ", " +"don gia: " + 8669 +", "+ "tong gia muc3: " + (muc3 \* 8669));  
 tong = muc1\*5973 + muc2\*7052 + muc3\*8669;  
 }  
 else if(muctieuthu>30)  
 {  
 muc1 = 10;  
 System.*out*.println("muc1: " + muc1 + ", " +"don gia: " + 5973 +", "+ "tong gia muc1: " + (muc1 \* 7052));  
 muc2 = 10;  
 System.*out*.println("muc2: " + muc2 + ", " +"don gia: " + 7052 +", "+ "tong gia muc2: " + (muc2 \* 7052));  
 muc3 = 10;  
 System.*out*.println("muc3: " + muc3 + ", " +"don gia: " + 8669 +", "+ "tong gia muc3: " + (muc3 \* 8669));  
 muc4 = muctieuthu - 30;  
 System.*out*.println("muc4: " + muc4 + ", " +"don gia: " + 15929 +", "+ "tong gia muc4: " + (muc4 \* 15929));  
 tong = muc1\*5973 + muc2\*7052 + muc3\*8669 + muc4\*15929;  
 }  
 }  
  
 int thue = (int) Math.*round*(tong \* 0.05);  
 System.*out*.println("thue 5% : " + thue);  
 int phimt = (int) Math.*round*(tong \* 0.1);  
 System.*out*.println("phi mt: "+ phimt);  
 int tongcong = tong + thue + phimt;  
 return tongcong;  
 }  
  
 }

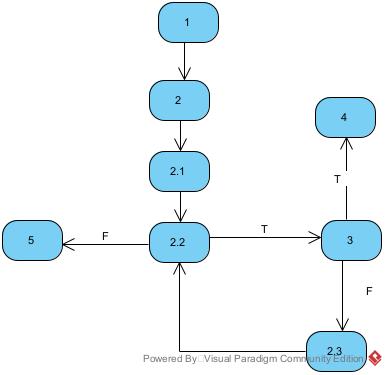
**Câu hỏi 3.3: Hàm bên dưới trả về chỉ số phần tử cuối cùng trong x có giá trị bằng y. Nếu không tồn tại, trả về giá trị -1.**

1. **int findLast(int[] x, int y){**
2. **for (int i = x.length -1(2.1); i > 0(2.2); i—(2.3)){**
3. **if (x[i] == y)**
4. **return i;**

**}**

1. **return -1;**

**}**

****

**Đường chạy**

**1 -> 2 -> 2.1 -> 2.2 -> 3 -> 4**

**1 -> 2 -> 2.1-> 2.2 -> 3 -> 2.3 -> 2.2 -> 3 -> 4**

**1 -> 2 -> 2.1 -> 2.2 -> 3 -> 2.3 -> 2.2 -> 5**

**1-> 2 -> 2.1 -> 2.2 -> 5**

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| **1 -> 2 -> 2.1 -> 2.2 -> 3 -> 4** | X[] = {1,2,3,4,5}, y = 1 | 1 |
| **1 -> 2 -> 2.1-> 2.2 -> 3 -> 2.3 -> 2.2 -> 3 -> 4** | X[] = {1,2,3,4,5}, y = 3 | 3 |
| **1 -> 2 -> 2.1 -> 2.2 -> 3 -> 2.3 -> 2.2 -> 5** | X[] = {1,2,3,4,5}, y = 6 | -1 |
| **1-> 2 -> 2.1 -> 2.2 -> 5** | X[]={},y=1 | -1 |