

**ĐẠI HỌC KHOA HỌC HUẾ**

-----🙞🙜🕮🙞🙜-----

****

**KIỂM ĐỊNH PHẦN MỀM**

**ĐỀ TÀI**

**KIỂM THỬ PHẦN MỀM QUẢN LÝ HỌC SINH**

**TRUNG HỌC PHỔ THÔNG**

Giáo viên hướng dẫn: Sinh viên thực hiện:

*Lê Văn Tường Lân Thái Trung Đức*

*Lê Chiêu Quốc*

*Trần Thị Hồng Uyên*

*Phạm Như Ngọc Tuấn*

*Lê Anh Đào*

***HUẾ, 12/2016***

**MỤC LỤC**

[**I. QUẢN LÝ DỰ ÁN:** 3](#_Toc469684766)

[***1. Danh sách thành viên:*** 3](#_Toc469684767)

[***2. Phân công công việc:*** 3](#_Toc469684768)

[***3. Theo dõi dự án:*** 3](#_Toc469684769)

[***4. Đánh giá thành viên:*** 4](#_Toc469684770)

[**II. THÔNG TIN DỰ ÁN:** 5](#_Toc469684771)

[***1. Tổng quan:*** 5](#_Toc469684772)

[***2. Mô tả chức năng:*** 5](#_Toc469684773)

[***3. Giao diện chức năng:*** 6](#_Toc469684774)

[***4. Dữ liệu liên quan:*** 8](#_Toc469684775)

[*4.1 Mối quan hệ giữa các bảng:* 8](#_Toc469684776)

[*4.2 Các bảng dữ liệu liên quan:* 9](#_Toc469684777)

[*5.* ***Code chức năng****:* 10](#_Toc469684778)

[**III. KIỂM THỬ CHỨC NĂNG:** 13](#_Toc469684779)

[***1. Phương pháp kiểm thử Module:*** 13](#_Toc469684780)

[***2. Phương pháp kiểm thử hộp đen:*** 19](#_Toc469684781)

[*2.1. Xác định bài toán:* 19](#_Toc469684782)

[*2.2. Xác định các TestCase dựa vào phân lớp tương đương:* 20](#_Toc469684783)

[*2.3. Xác định các TestCase:* 20](#_Toc469684784)

[*2.4. Kết quả mong muốn khi sử dụng TestCase:* 21](#_Toc469684785)

[*2.5. Biểu đồ nguyên nhân – kết quả:* 21](#_Toc469684786)

[***3. Phương pháp kiểm thử hộp trắng:*** 22](#_Toc469684787)

[*3.1 Xác định các nút:* 22](#_Toc469684788)

[*3.2 Vẽ đồ thị thể hiện đường diễn tiến của chương trình* 24](#_Toc469684789)

[*3.3 Xác định số đường kiểm thử V(G):* 25](#_Toc469684790)

[*3.4 Xác định các TestCase:* 25](#_Toc469684791)

# **I. QUẢN LÝ DỰ ÁN:**

## ***1. Danh sách thành viên:***

- Thái Trung Đức (Nhóm Trưởng) - dbom.1006@gmail.com - 01694969244

- Lê Chiêu Quốc - quoclee48@gmail.com - 0978938234

- Trần Thị Hồng Uyên - honguyen.cnttk37b@gmail.com - 0969748792

- Phạm Như Ngọc Tuấn - phamnhungoctuan@gmail.com - 01648645043

- Lê Anh Đào - daodidem@gmail.com – 01663387346

## ***2. Phân công công việc:***

|  |  |
| --- | --- |
| Tên công việc | Người chịu trách nhiệm |
| Tìm dự án | Thái Trung Đức |
| Chỉnh sửa dự án phù hợp với môn học | Thái Trung Đức |
| Kiểm thử Module | Phạm Như Ngọc Tuấn, Lê Anh Đào |
| Kiểm thử Hộp Trắng | Lê Chiêu Quốc, Trần Thị Hồng Uyên |
| Kiểm thử Hộp Đen | Lê Chiêu Quốc, Trần Thị Hồng Uyên |
| Tổng hợp, viết báo cáo | Thái Trung Đức |

## ***3. Theo dõi dự án:***

|  |  |
| --- | --- |
| Thời gian | Kết quả công việc |
| Buổi họp lần 1  11/09 | - Tìm dự án (Hoàn thành)  - Chỉnh sửa dự án (50%) |
| Buổi họp lần 2  16/10 | - Chỉnh sửa dự án (Hoàn thành)  - Thảo luận, phân chia công việc (Hoàn thành) |
| Buổi họp lần 3  22/10 | - Kiểm thử Module (30%)  - Kiểm thử Hộp Đen (80%) |
| Buổi họp lần 4  18/11 | - Kiểm thử Module (70%)  - Kiểm thử Hộp Đen (Hoàn thành)  - Kiểm thử Hộp Trắng (40%) |
| Buổi họp lần 5  6/12 | - Kiểm thử Module (Hoàn thành)  - Kiểm thử Hộp Trắng (Hoàn thành) |

## ***4. Đánh giá thành viên:***

- Các đánh giá dựa trên khối lượng mức độ tham gia của các thành viên, công việc thực hiện và việc hoàn thành sớm hay muộn.

|  |  |  |  |
| --- | --- | --- | --- |
| Thành viên | Công việc | Đánh giá | Tổng |
| Thái Trung Đức | 15% | Tốt | 8/10 |
| Lê Chiêu Quốc | 25% | Tốt, tích cực | 10/10 |
| Trần Thị Hồng Uyên | 20% | Tốt | 9/10 |
| Phạm Như Ngọc Tuấn | 20% | Tốt | 9/10 |
| Lê Anh Đào | 20% | Tốt | 9/10 |

# **II. THÔNG TIN DỰ ÁN:**

Ở đây, giới thiệu dự án mình thực hiện, sơ bộ, rồi chọn 1 chức năng cần kiểm thử để làm cả 3 phương pháp, như thầy đã dặn là nhóm mình làm chức năng đánh giá hạnh kiểm theo năm học nên chọn cái ni để làm.

## ***1. Tổng quan:***

- Tên phần mềm: Quản lý học sinh Trung Học Phổ Thông.

- Đơn vị thực hiện: Nguyễn Hoàng Tùng, Khoa Kỹ Thuật – Công Nghệ - Môi trường, Trường Đại Học An Giang

- Mục đích: Phần mềm giúp áp dụng CNTT vào quản lý học sinh trung học phổ thông được dễ dàng và hiệu quả, nhất là ở các trường học lớn, số học sinh đông, từ việc tiếp nhận học sinh (quản lý hồ sơ học sinh), phân lớp, phân giáo viên cho đến quản lý điểm, kết quả học tập, xuất báo cáo thống kê.

## ***2. Mô tả chức năng:***

* **ĐÁNH GIÁ HỌC LỰC THEO NĂM HỌC**

Bên cạnh nhiều nghiệp vụ trong Hệ thống Quản lý điểm của học sinh, việc đánh giá và phân loại Học lực cho học sinh đóng vai trò thiết yếu trong hệ thống.

* Chức năng cho phép đánh giá học lực của danh sách học sinh trong một lớp và trong một năm.
* Mỗi học sinh được đánh giá học lực theo điểm Trung Bình (ĐTB) cả năm theo tiêu chuẩn nhất định.
* Cụ thể hơn về tiêu chuẩn đánh giá, phân loại học lực:

+ Loại GIỎI: ĐTB các môn từ 8.0 trở lên, không có môn nào dưới 6.5

+ Loại KHÁ: ĐTB các môn từ 6.5 đến 7.9, không có môn nào dưới 5.0

+ Loại TB: ĐTB các môn từ 5.0 đến 6.4, không có môn nào dưới 3.5

+ Loại YẾU: ĐTB các môn từ 3.5 đến 4.9, không có môn nào dưới 2.0

+ Loại KÉM: Những trường hơp còn lại.

* Điểm Trung Bình của mỗi học sinh theo năm học phụ thuộc vào các cột điểm của học sinh và có công thức tính theo quy định.
* Có các loại điểm cần lưu trữ: **Điểm kiểm tra miệng**, **kiểm tra 15 phút**, **kiểm tra 1 tiết** 🢡 Điểm trung bình kiểm tra, **Điểm thi học kỳ** (hệ số 1) 🢡 Điểm trung bình môn học kỳ 🢡 Điểm trung bình môn cả năm. Hệ số và thang điểm phải theo quy định.

+ Điểm trung bình kiểm tra: là trung bình cộng của điểm kiểm tra miệng, 15 phút và 1 tiết:

DTBKT

+ Điểm trung bình môn học kỳ: là trung bình cộng của điểm thi học kỳ và điểm trung bình kiểm tra:

DTBMHK

+ Điểm trung bình môn cả năm: là trung bình cộng của điểm trung bình môn học kỳ 1 và điểm trung bình môn học kỳ 2:

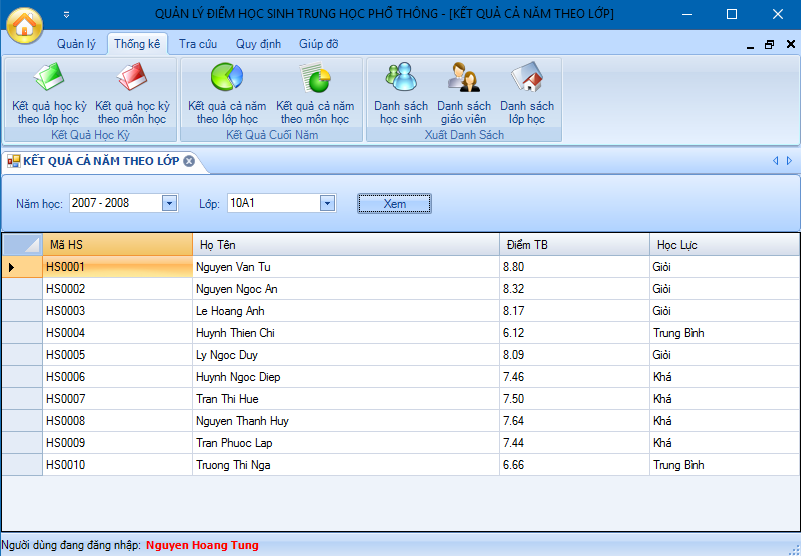
DTBMCN

+ Trung bình cộng của điểm trung bình môn cả năm (ĐTBMCN) của tất cả các môn nhân với hệ số (a, b,…) của từng môn:

DTBCMCN2

## ***3. Giao diện chức năng:***

Để xem xếp loại học lực cả năm, người dùng thực hiện chức năng “**Kết quả cả năm theo lớp học**” trên thanh Menu bar.

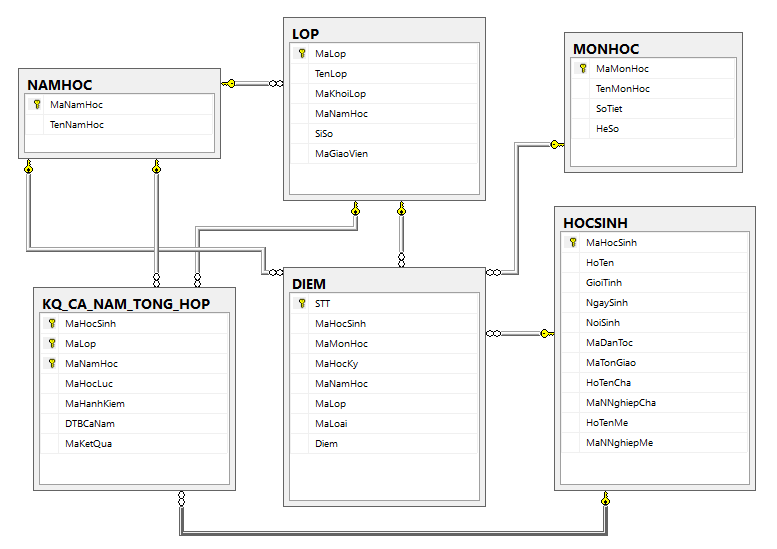


*Giao diện chức năng sau khi đã thực hiện*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Screen** | Kết quả cả năm theo lớp | | | | |
| **Description** | | Thống kê xếp loại học lực cả năm của học sinh. | | | |
| **Screen Access** | | Người dùng click vào **icon**  ở tab **Thống kê**. | | | |
| **Screen Content** | | | | | |
| **Item** | | **Type** | **Data** | **Description** | |
| Năm học | | Combobox |  | - Ô chọn năm học muốn xem kết quả học lực. | |
| Lớp | | Combobox |  | - Ô chọn lớp muốn xem kết quả học lực. | |
| Xem | | Button |  | - Click để xem danh sách. | |
| Danh sách học sinh | | DataGridview |  | - Hiển thị danh sách kết quả của học sinh bao gồm: Mã HS, Họ tên, Giới tính nữ, Ngày sinh, Điểm TB, **Học lực**, Hạnh kiểm. | |
| **Screen Actions** | | | | | |
| **Action Name** | | **Description** | | **Success** | **Failure** |
| Xem kết quả | | - Chọn năm học, lớp.. - Click Xem. | | Hiển thị danh sách kết quả học lực cả năm của học sinh. | Hiển thị danh sách rỗng. |

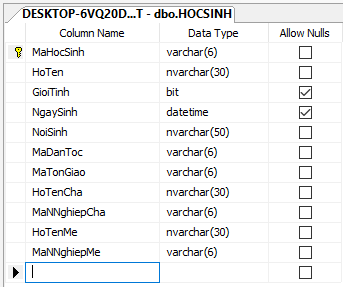
## ***4. Dữ liệu liên quan:***

### *4.1 Mối quan hệ giữa các bảng:*

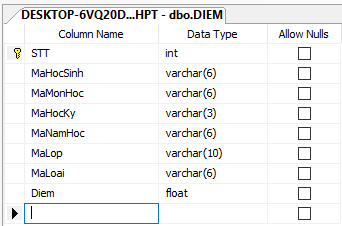


### *4.2 Các bảng dữ liệu liên quan:*

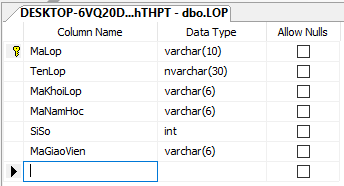
Bảng **HOCSINH**



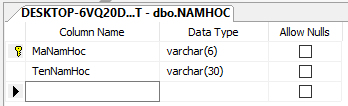
Bảng **DIEM**



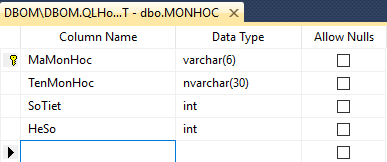
Bảng **LOP**



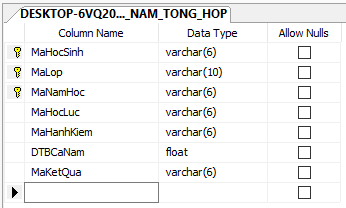
Bảng **NAMHOC**



Bảng **MONHOC**



Bảng **KQ\_CA\_NAM\_LOP\_HOC**



## *5.* ***Code chức năng****:*

private void DanhGiaHocLucTheoNam()

{

List<KetQuaHocLucInfo> lstKQ = new List<KetQuaHocLucInfo>();

//Khởi tạo các đối tượng get dữ liệu

var maLop = cmbLop.SelectedValue.ToString();

var namHoc = cmbNamHoc.SelectedValue.ToString();

String[] HocLuc = { "Giỏi", "Khá", "Trung Bình", "Yếu", "Kém" };

float[] diemCanDuoi = { 8, 6.5f, 5, 3.5f, 0};

HocSinhData m\_HocSinhData = new HocSinhData();

MonHocData m\_MonHocData = new MonHocData();

DiemCtrl m\_DiemCtrl = new DiemCtrl();

DataTable m\_DT = m\_MonHocData.LayDsMonHoc(namHoc, maLop);

List<HocSinhInfo> lstHS = m\_HocSinhData.LayDSHocSinhTheoLop(namHoc, maLop);

//Tính điểm TB và xếp loại học lực cho mỗi học sinh trong lớp

foreach (HocSinhInfo hs in lstHS)

{

//Khởi tạo các biến điểm cho mỗi học sinh

float tongDiem = 0; float tongDiemCacMon = 0;

float diemTBTungMon = 0; int tongHeSoCacMon = 0;

String xepLoai = "";

float[] arrayDiemTBTungMon = new float[m\_DT.Rows.Count];

int soMonHoc = 0;

//Tính điểm trung bình

foreach (DataRow row in m\_DT.Rows){

diemTBTungMon = m\_DiemCtrl.DiemTrungBinhMonCaNam(hs.MaHocSinh, row["MaMonHoc"].ToString(), namHoc, maLop);

arrayDiemTBTungMon[soMonHoc++] = diemTBTungMon;

tongDiemCacMon += diemTBTungMon \* Convert.ToInt32(row["HeSo"].ToString());

tongHeSoCacMon += Convert.ToInt32(row["HeSo"].ToString());

}

if (tongHeSoCacMon > 0) tongDiem = tongDiemCacMon / tongHeSoCacMon;

else tongDiem = 0;

//Tìm điểm TB Môn nhỏ nhất

float diemTBMonNhoNhat = arrayDiemTBTungMon[0];

for (int i = 1; i < arrayDiemTBTungMon.Length - 1; i++)

diemTBMonNhoNhat = Math.Min(diemTBMonNhoNhat, arrayDiemTBTungMon[i]);

//Xét học lực theo điểm TB và điểm TB môn nhỏ nhất

for (int i = 0; i < 4; i++){

if (tongDiem >= diemCanDuoi[i] && diemTBMonNhoNhat >= diemCanDuoi[i + 1]){

xepLoai = HocLuc[i];

break;

}

}

if (xepLoai == "") xepLoai = HocLuc[4].ToString();

lstKQ.Add(new KetQuaHocLucInfo(hs.MaHocSinh, hs.HoTen, tongDiem, xepLoai));

}

//Hiển thị ra gridview

bSKQCNTH.DataSource = lstKQ;

gvHocLuc.DataSource = bSKQCNTH;

}

# **III. KIỂM THỬ CHỨC NĂNG:**

## ***1. Phương pháp kiểm thử Module:***

Gọi mệnh đề thể hiện tính chất dữ liệu vào của chương trình **{A}** và mệnh đề  
thể hiện tính chất dữ liệu ra cần có **{B}**, ta có:

**{A: maLop ∈ string, namHoc ∈ string, HS0..HSn ∈ List<HocSinhInfo>, MH0..MHn ∈ List<MonHocInfo>, DiemTBMon0..DiemTBMonn ∈ Array<float>, HocLuc0..HocLuc4 ∈ Array<String>}**

**{B: lstKQ ∈ List<KetQuaHocLucInfo>}**

**Trong đó:**

* **maLop** là mã lớp được lấy từ combobox theo kiểu string
* **namHoc** là năm học được lấy từ combobox theo kiểu string
* **lstKQ**: là danh sách kết quả học lực của cả lớp theo năm theo kiểu KetQuaHocLucInfo
* **MH0..MHn:** là danh sách các môn học

Chứng minh **{A}P{B}**

**Dự Đoán**

* Dự đoán {**C1**} là bất biến của đoạn trình **Q1**

{**C1:**  HS0..HSn ∈ List<HocSinhInfo>; TinhHocLucVaXepLoai(HS0, HSn) = DaTinhHocLucVaXepLoai (HS0, HSi-1) U TinhHocLucVaXepLoai(HSi, HSn)}

* Dự đoán {**C2**} là bất biến của đoạn trình **Q2**

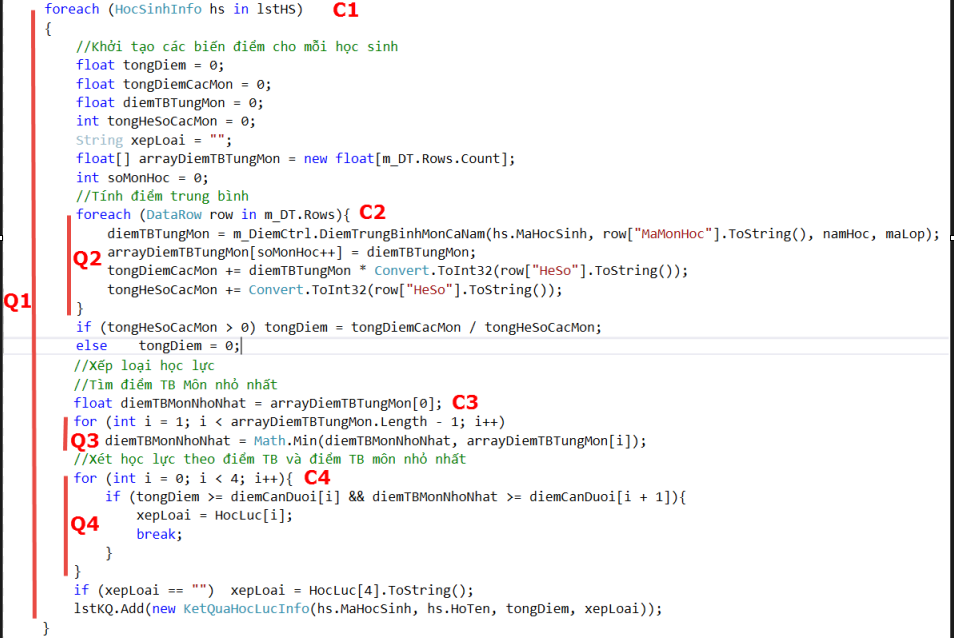
{**C2:**  MH0..MHn ∈ List<MonHocInfo>; TinhTongDiem(MH0, MHn) = DaTinhTongDiem (MH0, MHi-1) U TinhTongDiem (MHi, MHn)}

* Dự đoán {**C3**} là bất biến của đoạn trình **Q3**

{**C3:**  DiemTBMon0..DiemTBMonn ∈ Array<float>; TimDiemTBNhoNhat(DiemTBMon0 ,DiemTBMonn) = DaTimDiemTBNhoNhat (DiemTBMon0, DiemTBMoni-1) U TimDiemTBNhoNhat (DiemTBMoni, DiemTBMonn)}

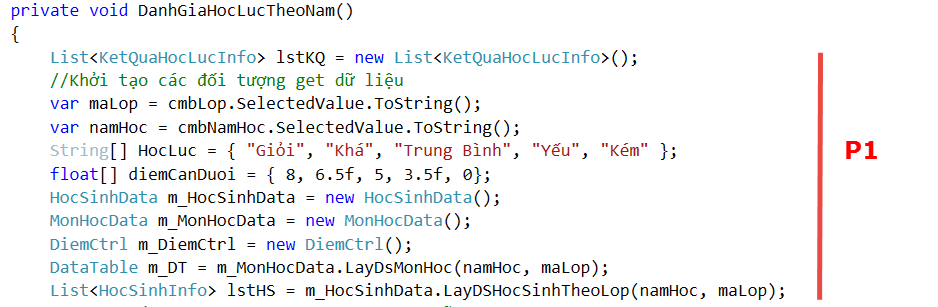
* Dự đoán {**C4}** là bất biến của đoạn trình **Q4**

{**C4:**  i = [0..4] ∈ int; XepHocLuc(0, n) = DaXepHocLuc (0, i-1) U XepHocLuc (i, n)}



**Chứng minh**

* {**A**} **P1**{**C­1**}, với đoạn trình **P1** như sau:



Ta có:  
{**A**} List<KetQuaHocLucInfo> lstKQ = new List<KetQuaHocLucInfo>();{**A­1**}

{**A­1**}String[] HocLuc = { "Giỏi", "Khá", "Trung Bình", "Yếu", "Kém" };{**A2**}

{**A­2**}float[] diemCanDuoi = { 8, 6.5f, 5, 3.5f, 0};{**A­3**}

{**A­3**}HocSinhData m\_HocSinhData = new HocSinhData();{**A4**}

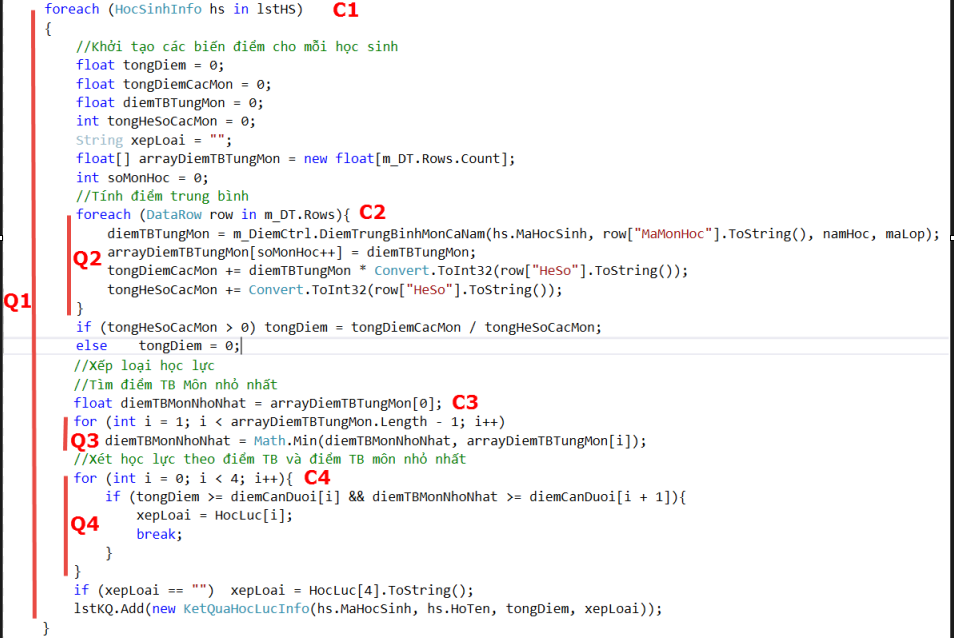
{**A­4**}MonHocData m\_MonHocData = new MonHocData();{**A5**}

{**A­5**}DiemCtrl m\_DiemCtrl = new DiemCtrl();{**A6**}

{**A­6**}DataTable m\_DT = m\_MonHocData.LayDsMonHoc(namHoc, maLop); {**A­7**}

{**A­7**}List<HocSinhInfo> lstHS = m\_HocSinhData.LayDSHocSinhTheoLop(namHoc, maLop); {**C1**}

* Chứng minh {**C**1} là bất biến của đoạn trình **Q1**, với đoạn trình **Q1** như sau:



Ta cần có: {**C1, E1**: lstHS ∈ List<HocSinhInfo>, HS0..HSn ∈ HocSinhInfo, i, n ∈ int; n = lstHS.count, HSi= lstHS[i]; TinhHocLucVaXepLoai(HS0, HSn) = DaTinhHocLucVaXepLoai (HS0, HSi-1) U TinhHocLucVaXepLoai(HSi, HSn); i < n;} **Q1 {C1}.**

Ta có:

{C1}float tongDiem = 0; {C11}

{C11}float tongDiemCacMon = 0; {C12}

{C12}float diemTBTungMon = 0; {C13}

{C13}int tongHeSoCacMon = 0; {C14}

{C14}String xepLoai = "";{C15}

{C15}float[] arrayDiemTBTungMon = new float[m\_DT.Rows.Count]; {C16}

{C16}int soMonHoc = 0; {C2}

* Chứng minh {**C**2} là bất biến của đoạn trình **Q2**.

Ta cần có: {**C2, E2**: lstMH ∈ List<MonHoc>, MH0..MHn ∈ MonHoc, i, n ∈ int; n = lstMH.count, MHi= lstMH[i]; TinhTongDiem (MH0, MHn) = DaTinhTongDiem(MH0, MHi-1) U TinhTongDiem(MHi, MHn); TinhTongHeSo(MH0, MHn) = DaTinhTongHeSo(MH0, MHi-1) U TinhTongHeSo(MHi, MHn); i < n;} **Q2 {C2}**

Theo tính chất của phép gán, ta có:

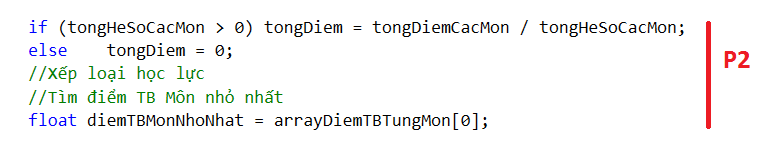
{**C21**} ≡ {C2[TinhTongHeSo|TinhTongHeSo +HeSo]: lstMH ∈ List<MonHoc>, MH0..MHn ∈ MonHoc, i, n ∈ int; n = lstMH.count, MHi= lstMH[i]; TinhTongDiem (MH0, MHn) = DaTinhTongDiem(MH0, MHi-1) U TinhTongDiem(MHi, MHn); **TinhTongHeSo(MH0, MHn) = DaTinhTongHeSo(MH0, MHi-1) U (TinhTongHeSo(MHi, MHn) + HeSo(MHi, MHn));**}

{**C22**} ≡ {C21[TinhTongDiem | TinhTongDiem + DiemTBTungMon \* HeSo]: lstMH ∈ List<MonHoc>, MH0..MHn ∈ MonHoc, i, n ∈ int; n = lstMH.count, MHi= lstMH[i]; **TinhTongDiem (MH0, MHn) = DaTinhTongDiem(MH0, MHi-1) U TinhTongDiem(MHi, MHn) + DiemTBTungMon(MHi, MHn) \* HeSo(MHi, MHn));** TinhTongHeSo(MH0, MHn) = DaTinhTongHeSo(MH0, MHi-1) U (TinhTongHeSo(MHi, MHn) + HeSo(MHi, MHn));}

**{C2, E2}Q2{C2}** nên kết thúc Q2 ta có mệnh đề **{C2, !E2}:**

**{C2, !E2:** lstMH ∈ List<MonHoc>, MH0..MHn ∈ MonHoc, i, n ∈ int; n = lstMH.count, MHi= lstMH[i]; **TinhTongDiem (MH0, MHn) = DaTinhTongDiem(MH0, MHi-1) U TinhTongDiem(MHi, MHn); TinhTongHeSo(MH0, MHn) = DaTinhTongHeSo(MH0, MHi-1) U TinhTongHeSo(MHi, MHn);** i >=n;**}**

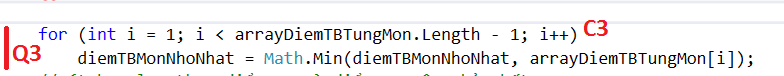
* **Chứng minh {C2, !E2}P2{C3}** với P2 là đoạn trình như sau:

****

**{C2, !E2}** if (tongHeSoCacMon > 0) tongDiem = tongDiemCacMon / tongHeSoCacMon; else tongDiem = 0**;{C21}**

**Dễ dàng ta thấy:**

* **{C2, !E2, E3: }** tongDiem = tongDiemCacMon / tongHeSoCacMon;{C21}
* **{C2, !E2, !E3: }** tongDiem = 0;{C21}
* **{C21}** diemTBMonNhoNhat = arrayDiemTBTungMon[0];{C3}
* Chứng minh {**C**3} là bất biến **Q3**, với Q3 là đoạn trình như sau:

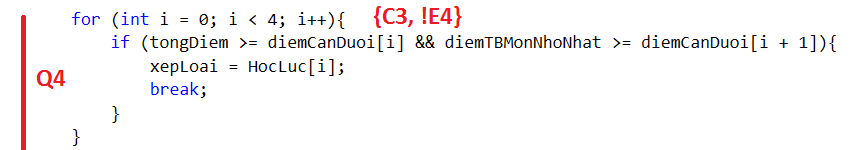
****

Ta có: {**C3, E4:** arrayDiemTBTungMon ∈ Array<float>, DiemTBMon0..DiemTBMonn ∈ float; i, n ∈ int; i = 1; n = arrayDiemTBTungMon.Length - 1; TimDiemTBNhoNhat(DiemTBMon0 ,DiemTBMonn) = DaTimDiemTBNhoNhat (DiemTBMon0, DiemTBMoni-1) U TimDiemTBNhoNhat (DiemTBMoni, DiemTBMonn); i < n - 1} **Q3 {C3}**

Vậy {C3} là bất biến của Q3. Nên kết thúc Q3, ta có mệnh đề **{C3, !E4}:**

{**C3, !E4:** arrayDiemTBTungMon ∈ Array<float>, DiemTBMon0..DiemTBMonn ∈ float; i, n ∈ int; i = 1; n = arrayDiemTBTungMon.Length - 1; TimDiemTBNhoNhat(DiemTBMon0 ,DiemTBMonn) = DaTimDiemTBNhoNhat (DiemTBMon0, DiemTBMoni-1) U TimDiemTBNhoNhat (DiemTBMoni, DiemTBMonn); i >= n - 1} **Q3 {C3}**

* Chứng minh **{C3, !E4}** là bất biến **Q4**, đặt **{C3, !E4} là {C4}** với Q4 là đoạn trình như sau:

****

Ta cần có: {**C4 , E5:** i = [0..4] ∈ int; i = 0; XepHocLuc(0, n) = DaXepHocLuc (0, i-1) U XepHocLuc (i, n); i < 4} **Q4 {C4}**

**Ta cần chứng tỏ:**

{**C4, E5**} if (tongDiem >= diemCanDuoi[i] && diemTBMonNhoNhat >= diemCanDuoi[i + 1]){

xepLoai = HocLuc[i]; break;} **{C4}**

Dễ dàng ta có

{**C4**, E5, E6: tongDiem >= diemCanDuoi[i] && diemTBMonNhoNhat >= diemCanDuoi[i + 1]} xepLoai = HocLuc[i]; break;{**C4**, **E5**}

{**C4**, E5, !E6: tongDiem < diemCanDuoi[i] || diemTBMonNhoNhat < diemCanDuoi[i + 1]} {**C4**, **E5**}

**Vậy {C4, E5} là bất biến của Q4. Kết thúc Q4, ta có mệnh đề {C4, !E5}**

* **Chứng minh {C4, !E5}P3{C41}** với P3 là đoạn trình như sau:

****

**Ta cần chứng tỏ:**

{**C4, !E5:** i = [0..4] ∈ int; i = 0; XepHocLuc(0, n) = DaXepHocLuc (0, i-1) U XepHocLuc (i, n); i >= 4**}** if (xepLoai == "") xepLoai = HocLuc[4].ToString();**{C41}**

**+ Dễ dàng ta có**

{**C4, !E5**, E7: xepLoai == ""} xepLoai = HocLuc[4].ToString();{**C4**, !E5}

{**C4**, !E5, !E7: xepLoai != ""}{**C4**, **!E5**}

* {**C4 , !E5}**lstKQ.Add(new KetQuaHocLucInfo(hs.MaHocSinh, hs.HoTen, tongDiem, xepLoai));**{C1}**

**Vậy** **{C1}** là bất biến của **Q1**. Nên kết thúc Q1, ta có mệnh đề **{C1, !E1}.**

**Với {C1, !E1:** lstHS ∈ List<HocSinhInfo>, HS0..HSn ∈ HocSinhInfo, i, n ∈ int; n = lstHS.count, HSi= lstHS[i]; TinhHocLucVaXepLoai(HS0, HSn) = DaTinhHocLucVaXepLoai (HS0, HSi-1) U TinhHocLucVaXepLoai(HSi, HSn); i >= n;}  
+ Dễ dàng chứng tỏ: **{C1, !E1}****{B}**  
Vậy ta có **{A}P{B},** hay chương trình trên là đúng

## ***2. Phương pháp kiểm thử hộp đen:***

### *2.1. Xác định bài toán:*

Đầu vào: một danh sách học sinh (gồm các thông tin liên quan và quan trọng có các cột điểm) được lấy từ mã lớp và mã năm học.

Đầu ra: một danh sách học sinh đã được xếp loại học lực cả năm theo lớp.

Thuật toán:

- Bước 1: đầu tiên khởi tạo các đối tượng mã năm học, mã lớp và đặt giá trị cho các đối tượng học lực và điểm cận dưới.

- Bước 2: khởi tạo giá trị các biến điểm cho học sinh (điểm TB các môn và hệ số các môn, điểm TB).

- Bước 3: tính điểm TB.

- Bước 4: dùng vòng lặp để tính Điểm TB môn nhỏ nhất.

- Bước 5: xét điểm TB >= từng điểm cận dưới && Điểm TB môn nhỏ nhất >= từng điểm cận dưới 🡺 xác định giá trị Xếp loại (gán giá trị Học lực cho Xếp loại).

- Bước 6: Hiển thị danh sách.

**Tiêu chuẩn phân loại học lực:**

+ Loại GIỎI: ĐTB các môn từ 8.0 trở lên, không có môn nào dưới 6.5

+ Loại KHÁ: ĐTB các môn từ 6.5 đến 7.9, không có môn nào dưới 5.0

+ Loại TB: ĐTB các môn từ 5.0 đến 6.4, không có môn nào dưới 3.5

+ Loại YẾU: ĐTB các môn từ 3.5 đến 4.9, không có môn nào dưới 2.0

+ Loại KÉM: Những trường hợp còn lại.

### *2.2. Xác định các TestCase dựa vào phân lớp tương đương:*

|  |  |  |
| --- | --- | --- |
| Các giá trị đầu vào | Lớp tương đương hợp lệ | Lớp tương đương không hợp lệ |
| Năm học: chuỗi ký tự  Lớp: chuỗi ký tự | Đã chọn Năm học tồn tại (đã hoặc đang diễn ra) và Lớp tồn tại trong Năm học, Lớp tồn tại học sinh (1) | - Năm học được chọn không tồn tại (chưa diễn ra) (2).  - Lớp được chọn không tồn tại (3).  - Cả Năm học và Lớp được chọn không tồn tại (4). |
| - Năm học được chọn không tồn tại Lớp được chọn (5).  - Lớp được chọn không tồn tại học sinh (6). |

### *2.3. Xác định các TestCase:*

* Các ca kiểm thử bao phủ các lớp tương đương hợp lệ là: (1)
  + 2007 – 2008, 10A1
* Các ca kiểm thử tương ứng với từng ca kiểm thử không hợp lệ:
  + (2) 2019 – 2020, 10A1
  + (3) 2007 – 2008, 13A1
  + (4) 2019 – 2020, 13A1
  + (5) 2007 – 2008, 10A3 (năm học này trường không phân lớp 10A3)
  + (6) 2007 – 2008, 11A2 (lớp 11A2 chưa có danh sách học sinh)

### *2.4. Kết quả mong muốn khi sử dụng TestCase:*

|  |  |
| --- | --- |
| Giá trị đầu vào kiểm thử | Kết quả mong muốn |
| Đã chọn Năm học tồn tại (đã hoặc đang diễn ra) và Lớp tồn tại trong Năm học đó, Lớp tồn tại học sinh (1) | Hiển thị danh sách học sinh theo Lớp trong Năm học đó và kết quả xếp loại học lực của từng học sinh. |
| Năm học được chọn không tồn tại (chưa diễn ra) (2). | Hiển thị danh sách rỗng. |
| Lớp được chọn không tồn tại (3). | Hiển thị danh sách rỗng. |
| Cả Năm học và Lớp được chọn không tồn tại (4). | Hiển thị danh sách rỗng. |
| Năm học được chọn không tồn tại Lớp được chọn (5). | Hiển thị danh sách rỗng. |
| Lớp được chọn không tồn tại học sinh (6). | Hiển thị danh sách rỗng. |

### *2.5. Biểu đồ nguyên nhân – kết quả:*

* **Nguyên nhân là:**

1. Đã chọn Năm học.

2. Đã chọn Lớp.

3. Đã chọn Năm học tồn tại (đã hoặc đang diễn ra) và Lớp tồn tại trong Năm học đó, Lớp tồn tại học sinh.

4. Năm học được chọn không tồn tại (chưa diễn ra).

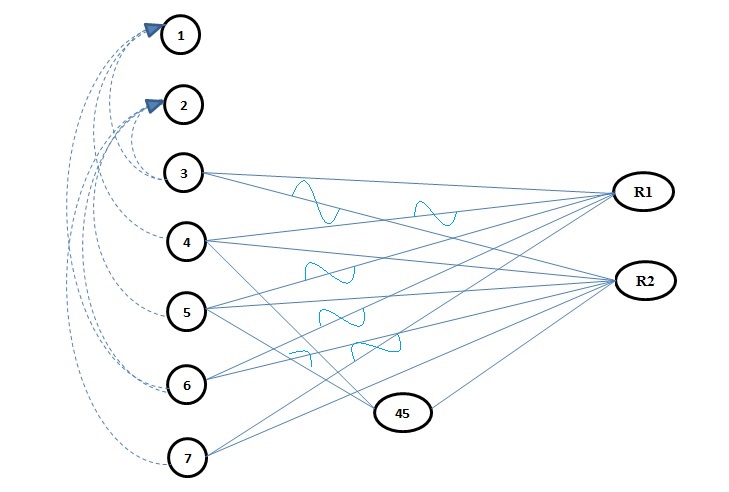
5. Lớp được chọn không tồn tại.

6. Năm học được chọn không tồn tại Lớp được chọn.

7. Lớp được chọn không tồn tại học sinh.

* **Kết quả là:**

1. Hiển thị danh sách học sinh theo Lớp trong Năm học đó và kết quả xếp loại học lực của từng học sinh.
2. Hiển thị danh sách rỗng.



## ***3. Phương pháp kiểm thử hộp trắng:***

*3.1 Xác định các nút:*

private void DanhGiaHocLucTheoNam()

{

List<KetQuaHocLucInfo> lstKQ = new List<KetQuaHocLucInfo>();

//Khởi tạo các đối tượng get dữ liệu

var maLop = cmbLop.SelectedValue.ToString();

var namHoc = cmbNamHoc.SelectedValue.ToString();

String[] HocLuc = { "Giỏi", "Khá", "Trung Bình", "Yếu", "Kém" };

float[] diemCanDuoi = { 8, 6.5f, 5, 3.5f, 0};

HocSinhData m\_HocSinhData = new HocSinhData();

MonHocData m\_MonHocData = new MonHocData();

DiemCtrl m\_DiemCtrl = new DiemCtrl();

DataTable m\_DT = m\_MonHocData.LayDsMonHoc(namHoc, maLop);

List<HocSinhInfo> lstHS = m\_HocSinhData.LayDSHocSinhTheoLop(namHoc, maLop);

//Tính điểm TB và xếp loại học lực cho mỗi học sinh trong lớp

foreach (HocSinhInfo hs in lstHS)

{

//Khởi tạo các biến điểm cho mỗi học sinh

float tongDiem = 0;

float tongDiemCacMon = 0;

float diemTBTungMon = 0;

int tongHeSoCacMon = 0;

String xepLoai = "";

float[] arrayDiemTBTungMon = new float[m\_DT.Rows.Count];

int soMonHoc = 0;

//Tính điểm trung bình

foreach (DataRow row in m\_DT.Rows)

{

diemTBTungMon = m\_DiemCtrl.DiemTrungBinhMonCaNam(hs.MaHocSinh, row["MaMonHoc"].ToString(), namHoc, maLop);

arrayDiemTBTungMon[soMonHoc++] = diemTBTungMon;

tongDiemCacMon+=diemTBTungMon\*Convert.ToInt32(row["HeSo"].ToString());

tongHeSoCacMon += Convert.ToInt32(row["HeSo"].ToString());

}

if (tongHeSoCacMon > 0)

tongDiem = tongDiemCacMon / tongHeSoCacMon;

else tongDiem = 0;

//Xếp loại học lực

//Tìm điểm TB Môn nhỏ nhất

float diemTBMonNhoNhat = arrayDiemTBTungMon[0];

for (int i = 1; i < arrayDiemTBTungMon.Length - 1; i++)

{

diemTBMonNhoNhat = Math.Min(diemTBMonNhoNhat, arrayDiemTBTungMon[i]);

}

//Xét học lực theo điểm TB và điểm TB môn nhỏ nhất

for (int i = 0; i < 4; i++)

{

if(tongDiem >= diemCanDuoi[i] && diemTBMonNhoNhat>=diemCanDuoi[i+1]){

xepLoai = HocLuc[i];

break;

}

}

if (xepLoai == "")

xepLoai = HocLuc[4].ToString();

lstKQ.Add(new KetQuaHocLucInfo(hs.MaHocSinh, hs.HoTen,tongDiem,xepLoai));

}

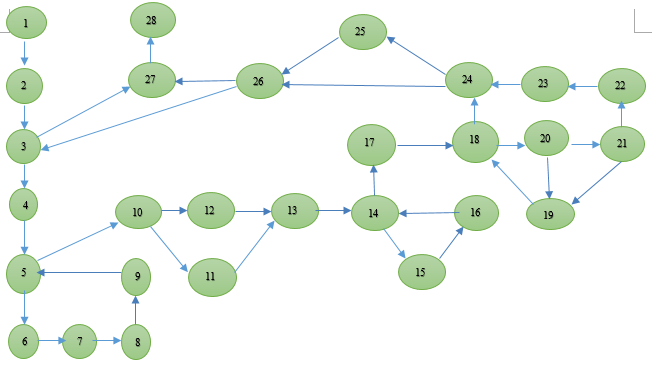
//Hiển thị ra gridview

bSKQCNTH.DataSource = lstKQ;

gvHocLuc.DataSource = bSKQCNTH;

}

*3.2 Vẽ đồ thị thể hiện đường diễn tiến của chương trình*



*3.3 Xác định số đường kiểm thử V(G):*

V(G) = 36 – 28 + 2 = 10

* Các đường kiểm thử là:

1/ 1.2.3.27

2/ 1.2.3.4.5.10…28

3/ 1.2.3.4.5…10.11.13.14.17….28

4/ 1.2.3.4.5…10.12.13.14.17….28

5/ 1.2.3.4.5...10.11.13.14....17.18.20.21.22.23.24.25.26(.3...26.3).27.28

6/ 1.2.3.4.5...10.12.13.14....17.18.20.21.22.23.24.25.26(.3...26).27.28

7/ 1.2.3.4.5...10.11.13.14....17.18.20.21.22.23.24.26(.3...26.3).27.28

8/ 1.2.3.4.5...10.12.13.14....17.18.20.21.22.23.24.26(.3...26.3).27.28

9/ 1.2.3.4.5...10.11.13.14....17.18.20.19(.18...19.18).20.24.25.26(.3...26).27.28

10/ 1.2.3.4.5...10.11.13.14...17.18.20.21.19(.18...19.18).20.24.25.26(.3...26).27.28

Ba chấm sau những đường 5, 14, 26 cho biết rằng 1 đường đi bất kì qua phần còn lại của cấu trúc điều khiển đều chấp nhận được.

*3.4 Xác định các TestCase:*

Bảng 1: Các trường hợp kiểm thử theo phương pháp kiểm thử đường thi hành cơ bản

|  |  |  |
| --- | --- | --- |
| Đường kiểm thử | Giá trị đầu vào | Kết quả mong đợi |
| 1 | ArrayDiemTBTungMon =null, tongDiem = null | Danh sách rỗng |
| 2 | ArrayDiemTBTungMon =0, tongDiem = 0 | Danh sách tất cả đều Kém |
| 3 | ArrayDiemTBTungMon = arrayValue , tongDiem = value | Danh sách đã được xếp loại |
| 4 | ArrayDiemTBTungMon = arrayValue , tongDiem = 0 | Danh sách tất cả đều Kém |
| 5 | ArrayDiemTBTungMon = arrayValue , tongDiem = value | Danh sách đã được xếp loại |
| 6 | ArrayDiemTBTungMon = arrayValue , tongDiem = 0 | Danh sách tất cả đều Kém |
| 7 | ArrayDiemTBTungMon = arrayValue , tongDiem = value | Danh sách không được xếp loại |
| 8 | ArrayDiemTBTungMon = arrayValue , tongDiem = value | Danh sách không được xếp loại |
| 9 | ArrayDiemTBTungMon = arrayValue , tongDiem = value | Danh sách đã được xếp loại |
| 10 | ArrayDiemTBTungMon = arrayValue , tongDiem = value | Danh sách đã được xếp loại |