****

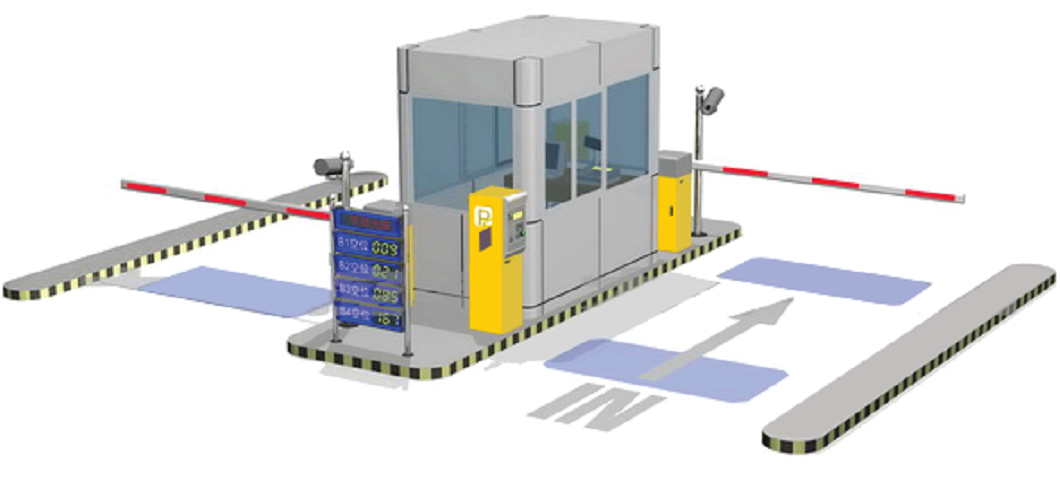
**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN TỬ - VIỄN THÔNG**

**------🙢⯎🙠*------***

**BÁO CÁO PBL1: LẬP TRÌNH**

****

**ĐỀ TÀI:**

**NHÀ XE THÔNG MINH BÁCH KHOA**

Giảng viên hướng dẫn: TS Nguyễn Duy Nhật Viễn

Sinh viên thực hiện: Nguyễn Văn Quý 22KTMT1

Nguyễn Thị Uyên Phương 22KTMT1

*Đà Nẵng, tháng 5 năm 2023*

**MỞ ĐẦU**

Với sự phát triển của khoa học trong thời kì 4.0, công nghệ ngày càng đóng vai trò quan trọng. Hiện nay, hầu hết tất cả các lĩnh vực trong cuộc sống đều có sự can thiệp của công nghệ.

Trường Đại Học Bách Khoa – Đại Học Đà Nẵng, nơi mà em đang theo học, là ngôi trường đào tạo kĩ thuật hàng đầu khu vực miền Trung – Tây Nguyên, với nhiều giảng viên giỏi, tận tâm, trang thiết bị dạy học hiện đại. Tuy nhiên, qua trải nghiệm thực tế, tụi em cảm thấy việc gửi xe ở trường hiện tại khá mất thời gian và bất tiện, vì mọi khâu trong nhà xe hoàn toàn thủ công, nên tụi em mong muốn xây dựng một hệ thống Nhà Xe Thông Minh Bách Khoa bằng những kiến thức được học từ môn Lập Trình Hướng Đối Tượng, dưới sự hướng dẫn của thầy Nguyễn Duy Nhật Viễn.

\*Bảng phân công nhiệm vụ:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Nhiệm vụ | Khối lượng |
| 1 | Nguyễn Văn Quý | Xây dựng bài toán, code class Xe, TheXe, camarara, baidoxe1chieu, maycheckthe. Viết hàm vận hành. Vẽ lưu đồ thuật toán bãi đỗ 1 chiều | 50% |
| 2 | Nguyễn Thị Uyên Phương | Xây dựng bài toán, code class NhanVien, Thethongminh, cameravao, baidoxe2chieu. Vẽ lưu đồ thuật toán đăng nhập, xây dựng hàm đăng nhập | 50% |

- Các file Word, Power Point, README, ref: cùng thảo luận làm.

**MỤC LỤC**

[**1.** **Giới thiệu:** 4](#_Toc136599931)

[*1.1.* *Tổng quan về dự án:* 4](#_Toc136599932)

[*1.2.* *Cơ sở lý thuyết:* 4](#_Toc136599933)

[*1.3.* *Mục tiêu:* 5](#_Toc136599934)

[**2.** **Ý tưởng sơ đồ nguyên lý hoạt động của hệ thống:** 5](#_Toc136599935)

[**3.** **Phân tích:** 6](#_Toc136599936)

[*3.1.* *Xác định các đối tượng:* 6](#_Toc136599937)

[*3.1.1.* *Class NhanVien:* 7](#_Toc136599938)

[*3.1.2.* *Class Xe* 9](#_Toc136599939)

[*3.1.3.* *Class TheXe* 11](#_Toc136599940)

[*3.1.4.* *Class Thethongminh* 12](#_Toc136599941)

[*3.1.5.* *Class cameravao:* 14](#_Toc136599942)

[*3.1.6.* *Class Maycheckthe:* 15](#_Toc136599943)

[*3.1.7.* *Class camerara:* 17](#_Toc136599944)

[*3.1.8.* *Class baidoxe1chieu:* 18](#_Toc136599945)

[3.1.9. Các hàm khác (giao diện, đăng nhập, gotoxy,... ): 19](#_Toc136599946)

[*3.2.* Thuật toán: 21](#_Toc136599947)

[*3.2.1.* *Thuật toán đăng nhập:* 21](#_Toc136599948)

[*3.1.2. Thuật toán bãi đỗ xe 1 chiều:* 22](#_Toc136599949)

[**4.** **Chạy chương trình, kết quả:** 23](#_Toc136599950)

[**5.** **Nhận xét, đánh giá:** 29](#_Toc136599951)

[**6.** **Định hướng phát triển:** 29](#_Toc136599952)

[**7.** **Nguồn tham khảo:** 30](#_Toc136599953)

1. ***Giới thiệu:***
   1. *Tổng quan về dự án:*

Để hệ thống nhà xe của Trường Đại Học Bách Khoa hiện đại hơn, mỗi lúc đi học, sinh viên sẽ đỡ mất thời gian trong việc gửi xe, vị trí để xe được bố trí hợp lí, tránh lãng phí diện tích nhà xe, tụi em đề xuất dự án kiểm soát nhà xe bách khoa bằng công nghệ. Đây không phải là 1 ý tưởng mới, nhưng tụi em sẽ cải tiến từ những cái đã có, để tạo nên 1 hệ thống hoàn thiện, khắc phục được các nhược điểm.

* 1. *Cơ sở lý thuyết:*

- Biết được cơ bản về lập trình hướng đối tượng và ngôn ngữ C++.

- Các khái niệm quan trọng trong lập trình hướng đối tượng như lớp, đối tượng, kế thừa, đa hình và đóng gói.

- Các đặc điểm và ưu điểm của lập trình hướng đối tượng trong ngôn ngữ C++.

- Các thuật toán cơ bản như tính toán, sắp xếp, ....

- Lưu đồ lớp, lưu đồ thuật toán.

- Tìm hiểu và vận dụng cái thư viện (bài báo cáo này sử dụng các thư viện như iostream, cmath (thư viện toán học), vector, string, ctime (thư viện dùng để in ra thời gian), fstream (đọc và ghi file), Windows.h (giao diện màn hình).)

- Trình tự và các đặc điểm của nhà xe.

* 1. *Mục tiêu:*

- Quản lí danh sách nhân viên nhà xe, ca trực, tổng thu mỗi ca.

- Tạo thẻ thông minh, thẻ xe cho sinh viên, lưu thông tin cá nhân, thông tin xe của mỗi sinh viên.

- Thanh toán phí gửi xe tự động.

- Vị trí để xe được hiển thị cụ thể.

- Lưu lại lịch sử vào ra của mỗi lượt xe.

- Kiểm tra thẻ, xe chặt chẽ.

1. **Ý tưởng sơ đồ nguyên lý hoạt động của hệ thống:**

**A picture containing text, screenshot, diagram

Description automatically generated**

• Mô tả sơ đồ:

Trước khi vào ca trực, nhân viên sẽ đăng nhập vào màn hình hệ thống bằng số điện thoại và mật khẩu đã được cấp.

→ Vào:

+ Lúc sinh viên đi vào đến cổng nhà xe và quét thẻ, nhân viên sẽ nhập mã số thẻ thông minh, loại xe, biển số xe, sau đó hệ thống sẽ hiển thị thông tin của sinh viên, biển số xe, thời điểm đi vào. Khi nhân viên nhấn chọn hành động đỗ xe, hệ thống sẽ hiển thị vị trí đỗ xe cho sinh viên.

→ Ra:

+ Khi sinh viên đi ra, nhân viên sẽ nhập lại mã số thẻ thông minh, biển số xe. Lúc này, hệ thống sẽ kiểm tra xem có trùng khớp với biển số xe lúc đi vào không, để tránh trường hợp dắt trộm xe ra. Sau khi kiểm tra, hệ thống sẽ tự động trừ tiền tài khoản.

Hệ thống sẽ ghi lại thông tin vào ra của từng lượt xe, thông tin nhân viên trong từng ca trực để dễ dàng quản lí, truy cứu khi cần thiết. Đồng thời, sau mỗi ca trực, khi nhân viên chọn đăng xuất, hệ thống cũng sẽ ghi lại tổng doanh thu của ca đó trong file.

1. **Phân tích:**
   1. *Xác định các đối tượng:*

|  |  |
| --- | --- |
| Name | Class |
| Nhân viên | NhanVien |
| Xe | Xe |
| Thẻ xe | TheXe |
| Thẻ Thông Minh | Thethongminh |
| Camera vào | cameravao |
| Camera ra | camerara |
| Máy check thẻ | Maycheckthe |
| Bãi đỗ xe một chiều | Baidoxe1chieu |

* + 1. *Class NhanVien:*

A screenshot of a computer

Description automatically generated with medium confidence

* + - 1. Giới thiệu:

Lớp NhanVien đại diện cho thông tin của một nhân viên nhà xe.

* + - 1. Thuộc tính:

• ten: Tên nhân viên

• ca: Ca trực (sáng/chiều)

• sdt: Số điện thoại nhân viên

• soluongnhanvien: biến static để đếm số nhân viên

• stt: Số thứ tự của nhân viên

* + - 1. Các phương thức:

- Hàm tạo có đối số mặc định

- Hàm setNhanVien(string name, string catruc, string sodt): khởi tạo biến cho các hàm setten, setca, setsdt.

- Hàm setten, setca, setsdt: Thiết lập giá trị cho các biến ten, ca, sdt của đối tượng nhân viên.

- Hàm getten(): Là phương thức của lớp NhanVien trả về biến ten để lưu tên nhân viên.

- Hàm getca(): Là phương thức của lớp NhanVien trả về biến ca để lưu ca trực của nhân viên.

- Hàm getsdt(): Là phương thức của lớp NhanVien trả về biến sdt để lưu số điện thoại của nhân viên.

- Hàm friend istream &operator >> (istream &is, NhanVien &n): Quá tải toán tử nhập nhân viên, cho phép nhập thông tin về tên, ca, số điện thoại của đối tượng nhân viên thông qua luồng đầu vào (istream& is).

- Hàm friend ostream &operator << (ostream &os, NhanVien &n): Quá tải toán tử xuất nhân viên, cho phép xuất thông tin về tên, ca và số điện thoại của đối tượng nhân viên ra luồng đầu ra (ostream& os).

* + - 1. Mối quan hệ:

Lớp NhanVien có mối quan hệ hàm friend với istream và ostream, cho phép toán tử >> và << truy cập và thao tác trực tiếp với dữ liệu của đối tượng nhân viên.

* + - 1. Các tính chất:

- Tính đóng gói: Sử dụng getter và setter để truy cập và cập nhật giá trị của các thuộc tính private của lớp NhanVien.

* + 1. *Class Xe*

*A screenshot of a computer

Description automatically generated with low confidence*

3.1.2.1. Giới Thiệu:

- Lớp Xe đại diện cho một chiếc xe trong hệ thống nhà xe thông minh.

3.1.2.2. Các thuộc tính:

* biensoxe: biển số xe.
* loaixe: loại xe.
* stt: số thứ tự của xe.
* soluongxe: biến static để đếm số lượng xe. ( stt của thẻ xe sẽ tương ứng với stt của số lượng xe)
* dt: biến để lưu thời điểm xe vào.

3.1.2.3. Các phương thức:

* Hàm Xe (string biensoxe, string loaixe): Đây là hàm khởi tạo của lớp Xe. Nó nhận vào hai tham số là biensoxe và loaixe để thiết lập giá trị cho biển số xe và loại xe của đối tượng Xe.
* Hàm friend istream& operator >> (istream& is, Xe& x): Đây là toán tử nhập dữ liệu được quá tải cho lớp Xe. Nó cho phép nhập thông tin về loại xe và biển số xe của đối tượng Xe từ người dùng thông qua luồng đầu vào (istream& is).
* Hàm void tgianxevao(): Đây là phương thức của lớp Xe để lấy thời điểm xe vào. Nó sử dụng hàm ctime để lấy thời gian hiện tại và lưu vào biến dt.
* Hàm friend ostream& operator << (ostream& os, Xe& x): Đây là toán tử xuất dữ liệu được quá tải cho lớp Xe. Nó cho phép xuất thông tin về loại xe, biển số xe và số thứ tự của đối tượng Xe ra luồng đầu ra (ostream& os).
* Hàm string getdtvao(): Đây là phương thức của lớp Xe để trả về thời điểm xe vào. Nó trả về giá trị của biến dt lưu trữ thời điểm xe vào.
* Hàm string getBienSoXe(): Đây là phương thức của lớp Xe để trả về biển số xe. Nó trả về giá trị của biến biensoxe.
* Hàm int getstt(): Đây là phương thức của lớp Xe để trả về số thứ tự của xe. Nó trả về giá trị của biến stt.

3.1.2.4. Mối quan hệ:

* Lớp Xe có mối quan hệ hàm friend với istream và ostream, cho phép toán tử >> và << truy cập và thao tác trực tiếp với dữ liệu của đối tượng Xe.

3.1.2.5. Tính kế thừa, đa hình và đóng gói:

* Tính đóng gói: Sử dụng các từ khóa private và public để định nghĩa phạm vi truy cập của các thành viên trong lớp Xe và sử dụng getter và setter để truy cập và cập nhật giá trị của các thuộc tính riêng tư của lớp Xe.
* Tính kế thừa: Hiện tại thì vẫn chưa có kế thừa nhưng tương lai thì sẽ có thể tạo lớp con kế thừa từ lớp Xe, như XeMay, XeDap, XeOto, để mô phỏng các loại xe cụ thể trong hệ thống nhà xe thông minh.
* Tính Đa Hình: Khi có các lớp con như XeMay, XeDap, XeOto sẽ định nghĩa các phương thức ảo trong lớp Xe và ghi đè chúng trong lớp con để thực hiện các hành vi riêng biệt cho từng loại xe.
  + 1. *Class TheXe*

*A picture containing text, font, screenshot, white

Description automatically generated*

3.1.3.1. Giới Thiệu:

Lớp TheXe đại diện cho thông tin của một thẻ xe. Nó có một thuộc tính sothe để lưu trữ số thẻ xe và một hàm khởi tạo để thiết lập giá trị cho sothe.

3.1.3.2. Đối tượng:

Một đối tượng của lớp TheXe được tạo ra khi chúng ta khởi tạo một thẻ xe với số thẻ xe cụ thể.

3.1.3.3. Các phương thức:

* Hàm tạo TheXe để lưu số thẻ vào
* Toán tử nhập dữ liệu (friend istream &operator >> (istream &is, TheXe &tx)) để cho phép người dùng nhập thông tin về số thẻ xe từ luồng đầu vào. Trong hàm này, số thẻ xe được nhập và lưu trữ vào biến sothe.
* getSoThe() để truy cập và trả về giá trị của thuộc tính sothe.

3.1.3.4. Mối quan hệ:

- Lớp TheXe có mối quan hệ hàm friend với istream cho phép toán tử >> truy cập và thao tác trực tiếp với dữ liệu của đối tượng TheXe.

3.1.3.5. Tính kế thừa, đa hình và đóng gói:

* Đóng gói: phương thức getSoThe() được sử dụng để truy cập giá trị của thuộc tính sothe từ bên ngoài lớp TheXe.
* Đa hình và kế thừa: Không có.
  + 1. *Class Thethongminh*

*A screenshot of a computer code

Description automatically generated with low confidence*

3.1.4.1. Giới Thiệu:

* Lớp Thethongminh là một lớp con của lớp TheXe, được kế thừa công khai (public) từ lớp cha là lớp TheXe.

3.1.4.2. Đối tượng:

* tenhocsinh: biến kiểu string để lưu tên học sinh.
* loaithenganhang: biến kiểu string để lưu loại thẻ ngân hàng.
* sothesinhvien: biến kiểu int để lưu số thẻ sinh viên.
* sothenganhang: biến kiểu long long int để lưu số thẻ ngân hàng.
* tientaikhoan: biến kiểu int để lưu số tiền trong tài khoản.
* soluongcard: biến tĩnh (static) kiểu int để đếm số lượng thẻ thông minh đã được tạo.
* stt: biến kiểu int để lưu số thứ tự của thẻ thông minh.

3.1.4.3. Các phương thức:

* Hàm khởi tạo Thethongminh nhận vào các tham số sothe, tenhocsinh, loaithenganhang, sothesinhvien, sothenganhang, tientaikhoan. Trong hàm khởi tạo, hàm cha TheXe cũng được gọi với tham số sothe. Các thành viên dữ liệu tenhocsinh, loaithenganhang, sothesinhvien, sothenganhang, tientaikhoan được gán bằng các giá trị tương ứng. Biến stt được gán giá trị bằng ++soluongcard.
* Các phương thức settenhocsinh, setloaithenganhang, setsothesinhvien, setsothenganhang, settientaikhoan được sử dụng để thiết lập giá trị cho các thành viên dữ liệu tương ứng.
* Các phương thức gettenhocsinh, getloaithenganhang, getsothesinhvien, getsothenganhang, gettientaikhoan được sử dụng để truy xuất giá trị của các thành viên dữ liệu tương ứng.
* Phương thức getsoluongcard trả về giá trị của biến tĩnh soluongcard, đếm số lượng thẻ thông minh đã được tạo.
* Phương thức getstt trả về giá trị của biến stt, số thứ tự của thẻ thông minh.
* Toán tử << (được định nghĩa là bạn của lớp) được sử dụng để in thông tin của thẻ thông minh ra màn hình. Nội dung in được định dạng và định vị bằng cách sử dụng hàm gotoxy và cout. Hàm Sleep được sử dụng để tạm dừng chương trình trong 3 giây.
* Phương thức kiemTraTien kiểm tra xem số tiền trong tài khoản (tientaikhoan) có đủ để trừ 1000 đơn vị hay không. Nếu có, phương thức trả về true, ngược lại trả về false. Nếu số tiền không đủ, một thông báo sẽ được in ra màn hình.
* Phương thức trutientaikhoan trừ 1000 đơn vị từ số tiền trong tài khoản (tientaikhoan) nếu kiểm tra số tiền đủ thực hiện trừ thành công.
* Phương thức GhiThongTinCard nhận vào một đối tượng ofstream và ghi thông tin của thẻ thông minh vào file. Các thông tin bao gồm số lượng thẻ thông minh (stt), số thẻ (sothe), tên học sinh (tenhocsinh), loại thẻ ngân hàng (loaithenganhang), số thẻ sinh viên (sothesinhvien), và số thẻ ngân hàng (sothenganhang).

3.1.4.4. Mối quan hệ:

* Lớp Thethongminh và lớp TheXe là một quan hệ kế thừa, trong đó lớp Thethongminh được xem là lớp con và lớp TheXe được xem là lớp cha. Mối quan hệ kế thừa cho phép lớp con (ở đây là Thethongminh) kế thừa tất cả các thành viên dữ liệu và phương thức từ lớp cha (ở đây là TheXe), và có thể mở rộng hoặc ghi đè các phương thức của lớp cha theo nhu cầu.
* Lớp Thethongminh có mối quan hệ hàm friend với ostream, cho phép toán tử << truy cập và thao tác trực tiếp với dữ liệu của đối tượng Thethongminh.

3.1.3.5. Tính kế thừa, đa hình và đóng gói:

* Đóng gói: getter/setter của các thành viên dữ liệu tenhocsinh, loaithenganhang, sothesinhvien, sothenganhang, tientaikhoan, soluongcard, stt.
* Kế thừa: lớp Thethongminh và lớp TheXe là một quan hệ kế thừa, chi tiết là kế thừa sothe của Class TheXe vào Class Thethongminh.
  + 1. *Class cameravao:*

A picture containing text, screenshot, font, line

Description automatically generated

* + - 1. Giới thiệu:

Lớp cameravao đại diện cho dữ liệu đầu vào khi một lượt xe đi vào nhà xe.

* + - 1. Đối tượng:

- biensoxe, loaixe: là biển số, loại xe mà camera vào check được khi có lượt xe vào, sẽ lưu theo thông tin của thẻ thông minh đã tích hợp cho sinh viên đó.

* + - 1. Các phương thức:

- Hàm checkXe: Làm nhiệm vụ kiểm tra xem các thông tin có hợp lệ không, biển số xe, loại xe nhập vào có đúng với đã đăng kí trong thẻ tích hợp không. Nếu qua hết các kiểm tra, xe được cho vào trong nhà xe.

- Hàm checkxechinhchu(): Kiểm tra xe chính chủ, nếu xe không chính chủ sẽ không được quét thẻ.

* + - 1. Mối quan hệ:
      2. Tính kế thừa, đa hình và đóng gói:

- Tính đóng gói: Các setter/getter của dữ liệu biensoxe và loaixe.

* + 1. *Class Maycheckthe:*

A close-up of a card

Description automatically generated with medium confidence

* + - 1. Giới thiệu:

Lớp Maycheckthe đại diện cho một thiết bị kiểm tra thẻ trong hệ thống quản lý bãi giữ xe. Nó chứa các phương thức để kiểm tra thông tin thẻ, kiểm tra tài khoản và xử lý quá trình ra vào của xe.

* + - 1. Đối tượng:

- soluong: Một biến static kiểu số nguyên lưu trữ số lượng máy check thẻ đã được khởi tạo.

- sott: Một biến kiểu số nguyên lưu trữ số thứ tự máy check thẻ.

* + - 1. Các phương thức:

- Maycheckthe(): Hàm khởi tạo mặc định. Tăng giá trị của soluong và gán giá trị cho sott khi khởi tạo một đối tượng máy check thẻ.

- checkCard(Thethongminh card): Phương thức static để kiểm tra thông tin trên thẻ thông minh. Kiểm tra thông tin học sinh và số tiền trong tài khoản. Trả về true nếu thông tin hợp lệ và đủ tiền, ngược lại trả về false.

- checkhocsinhdivao(Thethongminh card1): Phương thức để kiểm tra học sinh khi đi vào bãi đỗ xe. Sử dụng checkCard để kiểm tra thông tin trên thẻ và in thông báo ra màn hình.

- checkhocsinhdira(Thethongminh card1, Camerara cmrr1, Xe xe1): Phương thức để kiểm tra học sinh khi đi ra khỏi bãi đỗ xe. Sử dụng checkCard để kiểm tra thông tin trên thẻ và cmrr1.getbiensoxe() để kiểm tra biển số xe trên camera. In thông báo và trừ tiền trong tài khoản nếu hợp lệ.

* + - 1. Mối quan hệ:

Lớp MayCheckThe không có mối quan hệ.

* + - 1. Tính kế thừa, đa hình và đóng gói:

- Tính đóng gói: Sử dụng các từ khóa private và public để định nghĩa phạm vi truy cập của các thành viên trong lớp MayCheckThe và sử dụng getter để cập nhật giá trị của các thuộc tính riêng tư của lớp MayCheckThe.

- Tính kế thừa, đa hình: không có.

* + 1. *Class camerara:*

A screenshot of a computer

Description automatically generated with medium confidence

* + - 1. Giới thiệu:

- Lớp Camerara là một lớp đại diện cho hệ thống camera giám sát xe đi ra từ bãi đổ . Nó có các thuộc tính và phương thức để quản lý việc giám sát các phương tiện giao thông và kiểm tra thông tin về xe cộ.

* + - 1. Đối tượng:

- biensoxe: Một chuỗi lưu trữ biển số xe.

- soluotxera: Một biến static kiểu số nguyên lưu trữ số lượt xe ra khỏi camera.

- sttbichanlai: Một biến kiểu số nguyên lưu trữ số lượng xe bị cấm.

- sttra: Một biến kiểu số nguyên lưu trữ số lượng xe đã được giám sát qua camera.

- dt1: Một con trỏ char để lưu trữ thời gian xe ra khỏi camera.

- dt2: Một con trỏ char để lưu trữ thời gian xe bị cấm.

* + - 1. Các phương thức:

- Camerara(): Hàm khởi tạo mặc định. Tăng giá trị của soluotxera mỗi khi khởi tạo một đối tượng.

- getbiensoxe(): Phương thức trả về giá trị của biến biensoxe.

- tgianxera(Xe xe): Phương thức để lấy thời gian xe ra khỏi camera và in thông tin ra màn hình.

- tgianxebichanlai(Xe xe): Phương thức để lấy thời gian xe bị cấm và in thông tin ra màn hình.

- nhapxera(Xe xe1): Phương thức để nhập biển số xe của người mang thẻ xe để ra khỏi camera.

- kiemtracamra(Xe xe1): Phương thức để kiểm tra thông tin xe qua camera và in kết quả ra màn hình.

- getdtra(): Phương thức trả về giá trị của con trỏ dt1.

- getdtchan(): Phương thức trả về giá trị của con trỏ dt2.

* + - 1. Mối quan hệ:
      2. Tính kế thừa, đa hình và đóng gói:

- Tính đóng gói: Sử dụng các từ khóa private và public để định nghĩa phạm vi truy cập của các thành viên trong lớp camerara và sử dụng getter để cập nhật giá trị của các thuộc tính riêng tư của lớp camerara.

- Tính kế thừa, đa hình: không có.

* + 1. *Class baidoxe1chieu:*

A screenshot of a computer code

Description automatically generated with low confidence

* + - 1. Giới thiệu:

baidoxe1chieu đại diện cho bãi đỗ xe một chiều trong hệ thống quản lý bãi đỗ xe. Nó chứa các phương thức để nhập số lượng chỗ đỗ xe, chọn hành động (đỗ xe hoặc lấy xe), và cách vận hành của bãi đỗ xe.

* + - 1. Đối tượng:

- n: Số lượng chỗ đỗ xe trong bãi.

- hanhdong: Biến lưu trữ hành động được chọn.

- doxe: Một vector lưu trữ trạng thái của các chỗ đỗ xe (0: trống, 1: đã đỗ).

- chodoxecontrong: Vị trí chỗ đỗ xe trống để đỗ xe.

- layxe: Vị trí chỗ đỗ xe cần lấy xe.

* + - 1. Các phương thức:

- nhapsoluongchodexe(): Cho phép người dùng nhập số lượng chỗ đỗ xe còn trống.

- chonhanhdong(): Hiển thị menu cho phép người dùng chọn hành động.

- cachvanhanh(): Thực hiện vòng lặp để chọn và thực hiện các hành động trong bãi đỗ xe. Hành động bao gồm đỗ xe và lấy xe. Nếu không còn chỗ đỗ xe trống, thông báo sẽ được hiển thị. Nếu vị trí chỗ đỗ xe không hợp lệ hoặc không có xe tại vị trí đó, thông báo tương ứng sẽ được hiển thị.

* + - 1. Mối quan hệ:

Lớp Baidoxe1chieu không có mối quan hệ.

* + - 1. Tính kế thừa, đa hình và đóng gói:

Lớp Baidoxe1chieu không có tính kế thừa, đa hình và đóng gói.

### 3.1.9. Các hàm khác (giao diện, đăng nhập, gotoxy,... ):

3.1.9.1. Hàm gotoxy(int x, int y):

* Hàm gotoxy được định nghĩa để di chuyển con trỏ đến tọa độ (x, y) trên cửa sổ dòng. Đây là một phần của thư viện Windows.h và sử dụng các hàm của Windows API để thực hiện việc di chuyển con trỏ.

3.1.9.2. Hàm giaodien():

* dùng để tạo giao diện ban đầu cho chương trình và cho phép người dùng chờ trước khi bắt đầu thao tác.

3.1.9.3. Hàm dangnhap():

- Hàm DangNhap() bắt đầu bằng việc khai báo và khởi tạo các biến cần thiết như FileOut và FileIn, cùng với các biến lưu trữ thông tin như sdt, ten, ca, mk, pass và x.

- Gọi hàm giaodien() để hiển thị giao diện đăng nhập lên màn hình console.

- Sử dụng vòng lặp do-while để yêu cầu người dùng nhập lựa chọn. Nếu người dùng chọn "2", hàm sẽ kết thúc bằng cách gọi hàm exit(0) để thoát khỏi chương trình. Nếu người dùng nhập một lựa chọn không hợp lệ, thông báo lỗi sẽ được hiển thị và người dùng sẽ được yêu cầu nhập lại.

- Sau khi người dùng chọn "1" để đăng nhập, hàm sẽ tiếp tục vào vòng lặp while để yêu cầu người dùng nhập số điện thoại và mật khẩu.

- Trong quá trình nhập mật khẩu, các ký tự được hiển thị dưới dạng '\*' bằng cách sử dụng hàm getch() để lấy ký tự ngay khi người dùng nhập vào, sau đó ký tự đó sẽ được thêm vào chuỗi pass. Quá trình này diễn ra cho đến khi người dùng nhấn Enter.

- Sau khi nhập mật khẩu xong, hàm sẽ kiểm tra xem mật khẩu nhập vào có khớp với mật khẩu mặc định (mk) hay không. Nếu khớp, biến xacthucthongtin sẽ được đặt thành true và quá trình đăng nhập được tiếp tục. Nếu không khớp, thông báo lỗi sẽ được hiển thị và người dùng sẽ được yêu cầu nhập lại mật khẩu.

- Hàm tiếp tục bằng việc mở file có tên là số điện thoại (được nhập bởi người dùng) và đọc thông tin tương ứng của người dùng từ file đó.

- Nếu file tồn tại, hàm sẽ đọc thông tin người dùng và kiểm tra xem mật khẩu mặc định có khớp (mk == "0000") hay không. Nếu khớp, biến taptintontai sẽ được đặt thành true và thông báo đăng nhập

* 1. Thuật toán:
     1. *Thuật toán đăng nhập:*

A diagram of a flowchart

Description automatically generated with medium confidence

### *3.1.2. Thuật toán bãi đỗ xe 1 chiều:*

A diagram of a flowchart

Description automatically generated with low confidence

1. **Chạy chương trình, kết quả:**
   1. Giao diện mở đầu của chương trình:

A picture containing text, screenshot, font, line

Description automatically generated

* 1. Bắt đầu project, đầu tiên nhập lựa chọn (Đăng nhập/Thoát khỏi chương trình):

A screen shot of a computer

Description automatically generated with medium confidence

* + 1. Nếu lựa chọn đăng nhập, màn hình sẽ hiển thị:

A screen shot of a computer

Description automatically generated with medium confidence

* + 1. Màn hình khi đăng nhập thành công:

A screenshot of a computer

Description automatically generated with medium confidence

* + 1. Màn hình khi nhập mật khẩu không đúng:

A screenshot of a computer

Description automatically generated with medium confidence

* + 1. Khi nhập số điện thoại không đúng:

A picture containing text, screenshot, font

Description automatically generated

* + 1. Sau khi đăng nhập thành công, thông tin nhân viên sẽ được hiển thị:

A screen shot of a computer

Description automatically generated with low confidence

* + 1. Lúc này, nhân viên sẽ nhập số lượng chỗ để xe còn trống:

A black screen with white text

Description automatically generated with low confidence

* + 1. Giao diện chọn chức năng, nếu dữ liệu nhập vào không phải là 1, 2, 3 thì sẽ nhập lại đến khi nhập đúng:

A screen shot of a computer

Description automatically generated with low confidence

1. Nếu lựa chọn chức năng đi vào, thì tiếp tục nhập vào số thẻ xe:

A screen shot of a computer

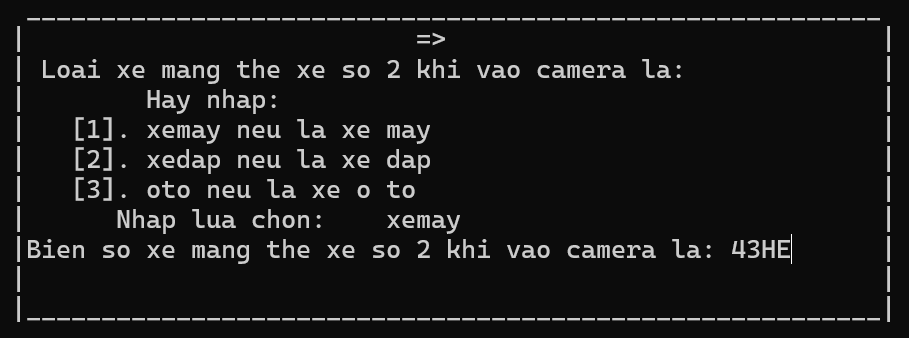
Description automatically generated with low confidence

- Nhập vào loại xe, nếu nhập sai thì nhập lại theo mẫu:

A screen shot of a computer

Description automatically generated with medium confidence

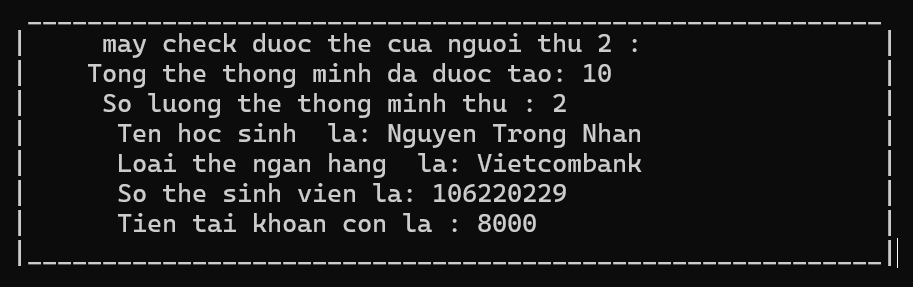
- Sau đó tiếp tục nhập biển số xe:

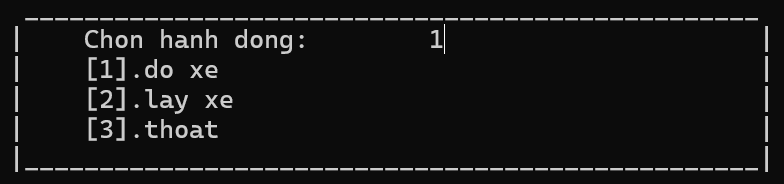


- Lúc này màn hình console sẽ hiển thị thời điểm vào của lượt xe, và thông tin sinh viên:

A picture containing text, screenshot, font, line

Description automatically generated



- Giao diện chọn hành động đỗ xe, lấy xe, hoặc thoát:

- Màn hình khi lựa chọn là đỗ xe:

A black screen with white text

Description automatically generated with low confidence

1. Nếu lựa chọn chức năng đi ra:

A picture containing text, font, screenshot, line

Description automatically generated

A black screen with white text

Description automatically generated with low confidence- Nhập số thẻ xe, tiếp tục chọn hành động lấy xe:

- Sau khi nhập vị trí chỗ đỗ xe cần lấy và biển số xe, màn hình sẽ hiển thị thông tin thời điểm xe ra khỏi bãi:

A black screen with white text

Description automatically generated with low confidence

- Nếu biển số xe không giống với lúc đi vào, màn hình sẽ hiển thị:

A black screen with white text

Description automatically generated with low confidence

* + 1. File ghi lại thông tin lượt xe vào ra, và khi nhân viên chọn đăng xuất, tổng doanh thu mỗi ca trực cũng sẽ được ghi:

A screenshot of a computer

Description automatically generated

1. **Nhận xét, đánh giá:**

- Hoàn thành dự án giúp tụi em hoàn thiện hơn kĩ năng lập trình, vận dụng được phần lý thuyết được học từ môn Lập Trình Hướng Đối Tượng để thực hiện hóa các ý tưởng trong chương trình. Từ những góp ý của thầy Nguyễn Duy Nhật Viễn trong những lần tụi em trình bày tiến trình dự án, tụi em học hỏi thêm được cách tổ chức chương trình sao cho hợp lí, gọn gàng. Bên cạnh đó, quá trình làm dự án cũng giúp tụi em cái thiện kĩ năng làm việc nhóm, để tìm ra được giải pháp tối ưu cho chương trình.

- Từng class được xây dựng chỉnh chu từ khâu lên ý tưởng, vẽ lưu đồ đến khâu viết chương trình.

- Tuy nhiên, chương trình vẫn còn một số nhược điểm. Chưa có chức năng nạp tiền vào thẻ thông minh, một vài chỗ vẫn còn lãng phí bộ nhớ, và một vài điểm chưa hay khác nữa, tụi em rất mong được thầy góp ý thêm.

1. **Định hướng phát triển:**

- Sắp tới khi được học về hệ thống nhúng, trí tuệ nhân tạo, trang bị đầy đủ kiến thức hơn thì tụi em mong muốn mỗi thẻ quét vào sẽ được máy check tự động lấy thông tin.

- Camera cũng sẽ có hệ thống tự nhận diện biển số để tăng tốc độ xử lí, và nhà xe cũng không cần nhân viên.

- Ở cổng sẽ có cần gạt, tự động đóng mở khi có lượt xe vào, ra nhà xe.

- Tích hợp chức năng nạp tiền cho thẻ thông minh.

- Phát triển chức năng tìm kiếm sinh viên theo tên.

1. **Nguồn tham khảo:**

[1]. Thao tác vào ra file cơ bản từ CodeLearn:

<https://codelearn.io/sharing/su-dung-file-sieu-co-ban-voi-cpp>

[2]. Hàm thời gian:

<https://codelearn.io/sharing/5-phut-tim-hieu-date-va-time-trong-cpp>

[3]. Hàm gotoxy:

<https://cachhoc.net/2013/08/22/cc-gotoxy-trong-dev-c-gotoxy-in-dev-c/?fbclid=IwAR0TpaIlIZNlpRNvFOai2fxydGyHd0YTk-W7PgoKXzQbZilIN-eZsC3yd1g>

[4]. Mảng, vòng lặp for:

<https://codelearn.io/learning/cpp-advanced?fbclid=IwAR1RK17C9YYunMZ7ZKwKVR4oToPU52NTPQ9LmgePETOxPUpWLJT2onAHYjs>

[5]. Kiến thức về lập trình hướng đối tượng:

<https://nhatvien.com/courses/lap-trinh-huong-doi-tuong/>