

پروژه پیشرفته
مدیریت مهد کودک

نویسنده گزارشکار
اسماء چگنی

شماره دانشجویی
۹۹۱۲۳۵۸۰۱۵

در این پروژه قرار است یک مهدکودک مدیریت شود. ساختار برنامه به گونه ای است که هم توسط بچه ها و هم توسط مدیر مهدکودک قابل استفاده می باشد.

بدین صورت که در ابتدای برنامه کاربر باید یوزرنیم و پسورد خود را وارد کند و پس از بررسی و تشخیص نوع کاربر وارد برنامه شده و امکانات مورد استفاده ی کاربر نمایش داده میشود.

در ادامه به توضیح کلاس های پیاده سازی در این فاز می پردازم.

کلاس Login

این کلاس برای بررسی یوزرنیم وپسورد وارد شده توسط کاربر ایجاد شده و بدین صورت عمل می کند که اطلاعات ورودی با اطلاعات فایل (userData.txt) چک میشود و در صورت وجود کاربر یک شی از نوع همان کاربر ساخته میشود و در ادامه شی های مربوط به کاربر ساخته کهدر فاز های بعدی تکمیل میگردد.

در این کلاس یک استراکت برای خواندن اطلاعات درون فایل تعریف شده در واقع بر اساس ساختار همین استراکت و منیپولیتور های استفاده شده در توابع همین کلاس اطلاعات درون فایل نوشته شده است.

یک داده شمارشی هم برای برگرداندن نوع کاربر تعریف شده است.

```
#ifndef LOGIN_HPP
#define LOGIN_HPP
#include <string>
//struct for read information'S users
struct info
{
    std::string firstName;
    std::string lastName;
    std::string username;
    std::string password;
    int age;
};
//enum to specify the type of user
enum state
{
    MANAGER,
    CHILD,
    UNDEFINED_USER
};
class Login
{
public:
    Login();
    ~Login();
    state searchUser(std::string, std::string);
    std::string getFirstName() const;
    std::string getLastName() const;
    int getAge() const;

private:
    info user;
};
#endif //LOGIN_HPP
```

کانستراکتور و دیستراکتور این کلاس در حال حاضر دارای بدنه ی خالی میباشند.

تابع searchUser

این تابع اصلی ترین تابع این کلاس است. که در آن فایل با مود in باز شده و اطلاعات آن به ترتیب و ساختار مناسب تا انتهای فایل خوانده میشود و در صورت تطابق یوزرنیم و پسورد داده شده به چک کردن دو حرف اول یوزرین می پردازد. این کار برای تشخیص نوع کاربر انجام میشود. و بعد نوع کاربر را توسط یک داده ی شمارشی بر میگرداند.

تابع getFirstName

تابع getLastName

تابع getAge

این سه تابع برای برگرداندن اطلاعات کاربر تعریف شده اند و برای تعریف شی هایی از نوع همان کاربر مورد استفاده قرار خواهند گرفت.

در این کلاس یک استرکات از همان نوع تعریف شده قرار گرفته شده برای ذخیره ی اطلاعات کاربر و مورد استفاده قرار گرفتن در توابع گفته شده.

کلاس Manager

این کلاس برای مدیر مهدکودک تعریف شده است. متغیر های این کلاس نام و نام خانوادگی مدیر می باشد.

کانستراکتور این کلاس توابع `setFirstName` و `setLastName` را فراخوانی میکند.

توابع `get` و `set` برای نام و نام خانوادگی مدیر تعریف شده است در توابع `set` اعتبارسنجی صورت گرفته است که نام و نام خانوادگی تماما از حروف الفبا تشکیل شده باشند.

در غیر این صورت پیغام خطا به کاربر نشان داده میشود.

توابع `get` هم که مقدار نام و نام خانوادگی را بر میگردانند.

```
#ifndef MANAGER_HPP
#define MANAGER_HPP
#include <string>
class Manager
{
public:
    Manager(std::string, std::string);
    void setFirstName(std::string);
    void setLastName(std::string);
    std::string getFirstName() const;
    std::string getLastName() const;

private:
    std::string FirstName;
    std::string LastName;
};
#endif //MANAGER_HPP
```

کلاس Child

این کلاس برای بچه های مهدکودک تعریف شده است و متغیر های ی کلاس نام و سن کاربر است. سن برای این تعریف شده احتمالا در فاز های بعدی و در ادامه توسعه برنامه نیاز به چنین اطلاعاتی باشد.

ساختار این کلاس مشابه کلاس Manager است. کانستراکتور توابع `setAge` و `setFirstName` را فراخوانی میکند.

تابع `setAge`

در این تابع اعتبار سنجی صورت گرفته است که سن کاربر بین ۳ تا ۷ سال باشد در غیر این صورت پیغام خطا نمایش داده میشود.

تابع `setFirstName`

در این تابع هم اعتبار سنجی صورت گرفته است که نام تماما از حروف الفبا تشکیل شده باشد.

توابع `get` هم مقادیر نام و سن را بر میگردانند.

```
#ifndef CHILD_HPP
#define CHILD_HPP
#include <string>
class Child
{
public:
    Child(std::string, int);
    void setFirstName(std::string);
    void setAge(int);
    std::string getFirstName() const;
    int getAge() const;

private:
    std::string FirstName;
    int age;
};
#endif //MANAGER_HPP
```


کلاس ManageKids

این کلاس برای اجرای دستورات مدیر مهدکودک در نظر گرفته شده است. این کلاس دستورات حذف، اضافه و ویرایش اطلاعات بچه ها را در بر میگیرد. تمامی این تغییرات در فایلی که اطلاعات در آن قرار دارد انجام می شود. دو متغیر از همان استراکتی که در کلاس لاگین تعریف شده بود در نظر گرفته شده است.

تابع searchUser

برای اجرای تغییرات در اطلاعات برخی توابع نیاز به بررسی وجود کاربر دارند برای همین این تابع تعریف شده که در آن با توجه به یوزرنیم جستجو میکند و اگر کاربر مورد نظر پیدا شد true و در غیر این صورت false بر میگرداند.

توابع

ValidationAge

,ValidationNmae,

ValidationPassword,

ValidationUsername

در این توابع اطلاعاتی که قرار است به فایل اضافه شود اعتبار سنجی میشوند. سن باید بین ۳ تا ۷ باشد.

نام باید فقط از حروف الفبای تشکیل شده باشد.

پسورد باید بین ۴ تا ۱۰ کاراکتر باشد.

یوزرنیم دو حرف اول آن باید CH باشد.

تابع printChild

این تابع تمامی اطلاعات فایل را خوانده و در خروجی چاپ میکند.

تابع addChild

این تابع تمام اطلاعات مورد نیاز یک بچه را از مدیر دریافت و سپس اعتبار سنجی میکند در صورت معتبر بودن اطلاعات فایل با مود app باز شده و اطلاعات به انتهای فایل اضافه میشود.

تابع deleteChild

این تابع یوزرنیم مورد نظر مدیر را دریافت میکند و پس از اطمینان از وجود آن کاربر تمامی اطلاعات فایل به جز اطلاعات آن کاربر را در فایلی دیگر می نویسد و فایل قدیم حذف و فایل جدید ایجاد شده نامش تغییر میکند.

تابع editChild

این تابع نیز همانند تابع دیلیت عمل میکند با این تفاوت که از مدیر سوال میکند میخواهد چه اطلاعاتی را تغییر دهد و سپس مقدار جدید را دریافت و اعتبار سنجی میکند و اطلاعات جدید را در فایل جدید می رنویسد و فایل قبلی حذف می شود.

```
#ifndef MANAGEKIDS_HPP
#define MANAGEKIDS_HPP
#include <string>
#include "Child.hpp"
#include "Login.hpp"
class ManageKids
{
public:
    void addChild();
    void deleteChild(std::string);
    void editChild(std::string);
    void printChild();
    bool searchUser(std::string);
    bool ValidationName(std::string);
    bool ValidationAge(int);
    bool ValidationPassword(std::string);
    bool ValidationUsername(std::string);

private:
    info user;
    info temp;
};
#endif //MANAGEKIDS_HPP
```