

Title: DVD application

Explanation: This application is to simulate a DVD store. It has following functions:

1. Rent a DVD; that is, check out a DVD.
2. Return, or check in, a DVD.
3. Create a list of DVDs owned by the store.
4. Show the details of a particular DVD.
5. Print a list of all of the DVDs in the store.
6. Check whether a particular DVD is in the store.

Source code:

DVD.h:

```
#ifndef DVD_H
#define DVD_H

#pragma once
#include <string>
#include <iostream>
#include <vector>
using namespace std;

class DVD
{
public:
    DVD(string n,vector<string> s, string p, string d, string c, int num);
    void setTitle(string t);
    void setProducer(string p);
    void setStars(vector<string> s);
    void setDirector(string d);
    void setCompany(string c);
    void setNumber(int n);
    string getTitle();
    string getProducer();
    vector<string> getStars();
    string getCompany();
    int getNumber();
    int getRental();
    string getDirector();
    void rent();
    void returnDVD();
    void printInfo();
    ~DVD();

private:
```

```
    string nameMovie;
    vector<string> stars;
    string nameProducer;
    string director;
    string company;
    int number;
    int rental = 0;
};

#endif
```

DVD.cpp:

```
#include "DVD.h"
#include <vector>
#include <string>
DVD::DVD(string n,vector<string> s, string p, string d, string c, int num)
{
    nameMovie = n;
    stars = s;
    nameProducer = p;
    director = d;
    company = c;
    number = num;
}
string DVD::getCompany(){
    return company;
}
string DVD::getDirector(){
    return director;
}
int DVD::getNumber(){
    return number;
}
int DVD::getRental(){
    return rental;
}
vector<string> DVD::getStars(){
    return stars;
}
string DVD::getProducer(){
    return nameProducer;
}
string DVD::getTitle(){
    return nameMovie;
}
```

```

}
void DVD::setTitle(string t){
    this->nameMovie = t;
}
void DVD::setProducer(string p){
    this->nameProducer = p;
}
void DVD::setStars(vector<string> s){
    this->stars = s;
}
void DVD::setDirector(string d){
    this->director = d;
}
void DVD::setCompany(string c){
    this->company = c;
}
void DVD::setNumber(int n){
    this->number = n;
}
void DVD::rent(){
    if(rental < number) rental++;
    else cout << "This DVD is not available now" << endl;
}
void DVD::returnDVD(){
    if(rental > 0) rental--;
    else cout << "This DVD is not needed to return " << endl;
}
void DVD::printInfo(){
    cout << "Title: " << nameMovie << endl;
    cout << "Stars: ";
    for (string star : stars) {
        cout << star << ", ";
    }
    cout << endl;
    cout << "Producer: " << nameProducer << endl;
    cout << "Director: " << director << endl;
    cout << "Production Company: " << company << endl;
    cout << "Number of copies: " << number << endl;
    cout << "Number of rentals: " << rental << endl;
}
DVD::~~DVD()
{
}

```

Main.cpp:

```
#include "DVD.cpp"
#include <iostream>
#include <string>
#include <vector>

using namespace std;
void printAll(vector<DVD> &l);
DVD* search(vector<DVD> &l, string title);
void add(vector<DVD> &l);
void deleteDVD(vector<DVD> &l, string title, string producer);
void listAll(vector<DVD> &l);
void update(vector<DVD> &l, string title);
bool check(vector<DVD> &l, string title);
bool find(vector<DVD> &l, string title);
bool checkOutDVD(vector<DVD> &l, string title);
bool checkInDVD(vector<DVD> &l, string title);
void printMenu();
int main(){
    vector<DVD> list;
    printMenu();
    while(true){
        int choice;
        cout << "input your choice" << endl;
        cin >> choice;
        switch(choice){
            case 1: {
                string title;
                cout << "Enter the title of the DVD: ";
                cin >> title;
                if (find(list, title)) {
                    cout << "DVD is in the store." << endl;
                } else {
                    cout << "DVD is not in the store." << endl;
                }
                break;
            }
            case 2: {
                string title;
                cout << "Enter the title of the DVD to check-out: ";
                cin >> title;
                checkOutDVD(list, title);
                break;
            }
        }
    }
}
```

```

    case 3: {
        string title;
        cout << "Enter the title of the DVD to check in: ";
        cin >> title;
        checkInDVD(list, title);
        break;
    }
    case 4: {
        string title;
        cout << "Enter the title of the DVD to check availability: ";
        cin >> title;
        if (check(list, title)) {
            cout << "DVD is available for borrowing." << endl;
        } else {
            cout << "DVD is not available for borrowing." << endl;
        }
        break;
    }
    case 5: {
        listAll(list);
        break;
    }
    case 6: {
        printAll(list);
        break;
    }
    case 7: {
        string title;
        cout << "Enter the title of the DVD to update details: ";
        cin >> title;
        update(list, title);
        break;
    }
    case 8: {
        add(list);
        break;
    }
    case 9: {
        string title;
        string producer;
        cout << "Enter the title of the DVD to remove from the list: ";
        cin >> title;
        cout << "Enter the producer of the DVD to remove from the list:
";

```

```

        cin >> producer;
        deleteDVD(list, title, producer);
        break;
    }
    case 10: {
        string title;
        cout << "Enter the title of the DVD to search for: ";
        cin >> title;
        DVD* d = search(list, title);
        d->printInfo();
        break;
    }
    case 11: {
        cout << "Exiting the program." << endl;
        return 0;
    }
    default: {
        cout << "Invalid choice. Please try again." << endl;
        break;
    }
}

return 0;
}

void printAll(vector<DVD> &l){
    for(DVD d : l ){
        d.printInfo();
        cout << "-----" << endl;
    }
}

DVD* search(vector<DVD> &l, string title){
    for(DVD &d : l){
        if(d.getTitle() == title){
            cout << "This is the info of the movie: " << endl;
            d.printInfo();
            return &d;
        }
    }
}

void add(vector<DVD> &l){
    cin.ignore();
    cout << "Enter DVD title: ";
    string title;
    getline(cin, title);

```

```

    cout << "Enter names of the stars (comma-separated): ";
    string starsInput;
    getline(cin, starsInput);
    vector<string> stars;
    string starName = "";
    for (char c : starsInput) {
        if (c == ',') {
            stars.push_back(starName);
            starName = "";
        } else {
            starName += c;
        }
    }
    stars.push_back(starName);
    cout << "Enter producer name: ";
    string producer;
    getline(cin, producer);
    cout << "Enter director name: ";
    string director;
    getline(cin, director);
    cout << "Enter production company name: ";
    string productionCompany;
    getline(cin, productionCompany);
    cout << "Enter number of copies: ";
    int numCopies;
    cin >> numCopies;
    cin.ignore();
    DVD newDVD(title, stars, producer, director, productionCompany,
numCopies);
    l.push_back(newDVD);
    cout << "DVD added successfully." << endl;
}

void deleteDVD(vector<DVD> &l, string title, string producer){
    for (int i = 0; i < l.size(); i++) {
        if (l[i].getTitle() == title && l[i].getProducer() == producer) {
            l.erase(l.begin() + i);
            cout << "DVD removed successfully!" << endl;
            return;
        }
    }
    cout << "DVD not found." << endl;
}

void listAll(vector<DVD> &l){
    for(DVD d : l){

```

```

        string name = d.getTitle();
        cout << name << endl;
    }
}

void update(vector<DVD> &l, string title){
    DVD* d = search(l, title);
    if (d == nullptr) {
        cout << "DVD not found." << endl;
        return;
    }
    cout << "Enter the new details of the DVD: " << endl;

    // Get new details of the DVD from the user
    string newTitle, newStars, newProducer, newDirector,
    newProductionCompany;
    int newCopies;
    cin.ignore();
    cout << "Enter the title: ";
    getline(cin, newTitle);

    cout << "Enter names of the stars (comma-separated): ";
    string starsInput;
    getline(cin, starsInput);
    vector<string> stars;
    string starName = "";
    for (char c : starsInput) {
        if (c == ',') {
            stars.push_back(starName);
            starName = "";
        } else {
            starName += c;
        }
    }
    stars.push_back(starName);

    cout << "Enter the producer: ";
    getline(cin, newProducer);

    cout << "Enter the director: ";
    getline(cin, newDirector);

    cout << "Enter the production company: ";
    getline(cin, newProductionCompany);

```



```

    cout << "Enter the number of copies: ";
    cin >> newCopies;
    cin.ignore();
    // Update the details of the DVD
    d->setTitle(newTitle);
    d->setStars(stars);
    d->setProducer(newProducer);
    d->setDirector(newDirector);
    d->setCompany(newProductionCompany);
    d->setNumber(newCopies);

    cout << "DVD details updated." << endl;
    return;
    cout << "Not found" << endl;
}

bool check(vector<DVD> &l, string title){
    for(DVD d : l){
        if(d.getTitle() == title){
            if(d.getNumber() > d.getRental()){
                cout << "Still available" << endl;
                return true;
            }
            else cout << "Not available" << endl;
            return false;
        }
    }
    cout << "DVD not found" << endl;
    return false;
}

void printMenu() {
    cout << "Menu Options: " << endl;
    cout << "1. Check if a DVD exists in the store" << endl;
    cout << "2. Check-out a DVD" << endl;
    cout << "3. Check in a DVD" << endl;
    cout << "4. Check whether a particular DVD is available for borrowing" <<
endl;
    cout << "5. List all the titles of the DVDs in the store" << endl;
    cout << "6. Print all details of the DVDs in the store" << endl;
    cout << "7. Update details of a particular DVD" << endl;
    cout << "8. Add a new DVD to the list of DVDs" << endl;
    cout << "9. Remove a DVD from the list of DVDs" << endl;
    cout << "10. Search for a DVD, given the title of the DVD and print
details" << endl;
    cout << "11. Exit" << endl;
}

```

```

}
bool find(vector<DVD> &l, string title){
    for(DVD &d : l){
        if(d.getTitle() == title){
            return true;
        }
    }
    return false;
}
bool checkOutDVD(vector<DVD> &l, string title){
    for(DVD &d : l){
        if(d.getTitle() == title){
            d.rent();
            return true;
        }
    }
    return false;
}
bool checkInDVD(vector<DVD> &l, string title){
    for(DVD &d : l){
        if(d.getTitle() == title){
            d.returnDVD();
            return true;
        }
    }
    return false;
}
}

```

Test:

```

Menu Options:
1. Check if a DVD exists in the store
2. Check-out a DVD
3. Check in a DVD
4. Check whether a particular DVD is available for borrowing
5. List all the titles of the DVDs in the store
6. Print all details of the DVDs in the store
7. Update details of a particular DVD
8. Add a new DVD to the list of DVDs
9. Remove a DVD from the list of DVDs
10. Search for a DVD, given the title of the DVD and print details
11. Exit
input your choice
8
Enter DVD title: q
Enter names of the stars (comma-separated): w,e,r
Enter producer name: t
Enter director name: y
Enter production company name: u
Enter number of copies: 12
DVD added successfully.
input your choice
8
Enter DVD title: a
Enter names of the stars (comma-separated): s
Enter producer name: d
Enter director name: f
Enter production company name: g
Enter number of copies: 1
DVD added successfully.
input your choice
6
Title: q
Stars: w, e, r,
Producer: t
Director: y
Production Company: u
Number of copies: 12
Number of rentals: 0
-----
Title: a
Stars: s,
Producer: d
Director: f
Production Company: g
Number of copies: 1
Number of rentals: 0
-----
input your choice
1
Enter the title of the DVD: a
Enter the producer of the DVD: d
DVD is in the store.

```

```
input your choice
2
Enter the title of the DVD to check-out: a
Enter the producer of the DVD to check-out: d
input your choice
6
Title: q
Stars: w, e, r,
Producer: t
Director: y
Production Company: u
Number of copies: 12
Number of rentals: 0
-----
Title: a
Stars: s,
Producer: d
Director: f
Production Company: g
Number of copies: 1
Number of rentals: 1
-----
input your choice
2
Enter the title of the DVD to check-out: a
Enter the producer of the DVD to check-out: d
This DVD is not available now
input your choice
3
Enter the title of the DVD to check in: a
Enter the producer of the DVD to check in: d
input your choice
3
Enter the title of the DVD to check in: q
Enter the producer of the DVD to check in: t
This DVD is not needed to return
input your choice
4
Enter the title of the DVD to check availability: a
Enter the producer of the DVD to check availability: d
Still available
DVD is available for borrowing.
input your choice
5
q
a
input your choice
7
Enter the title of the DVD to update from the list: a
Enter the producer of the DVD to update from the list: d
This is the info of the movie:
Title: a
Stars: s,
```

```
Producer: d
Director: f
Production Company: g
Number of copies: 1
Number of rentals: 0
Enter the new details of the DVD:
Enter the title: asd
Enter names of the stars (comma-separated): zxc
Enter the producer: tyu
Enter the director: fds
Enter the production company: c
Enter the number of copies: 6
DVD details updated.
input your choice
6
Title: q
Stars: w, e, r,
Producer: t
Director: y
Production Company: u
Number of copies: 12
Number of rentals: 0
-----
Title: asd
Stars: zxc,
Producer: tyu
Director: fds
Production Company: c
Number of copies: 6
Number of rentals: 0
-----
input your choice
9
Enter the title of the DVD to remove from the list: asd
Enter the producer of the DVD to remove from the list: tyu
DVD removed successfully!
input your choice
6
Title: q
Stars: w, e, r,
Producer: t
Director: y
Production Company: u
Number of copies: 12
Number of rentals: 0
-----
```

```
input your choice
10
Enter the title of the DVD to search from the list: a
Enter the producer of the DVD to search from the list: s
not found
input your choice
11
Exiting the program.
```

Further explanation:

I used vector to store DVD instance. For search algorithm, I used for loop to transverse all DVD in the DVD vector, and return the DVD instance when the title and producer are corresponding to a specific instance. For the menu part, I used a while loop and a switch to select the option for user, and the program will continue running until the user select to exit;

Summary:

I think my program implements all the required function.