Composition

CSE 5910

www.cse.yorku.ca/course/5910

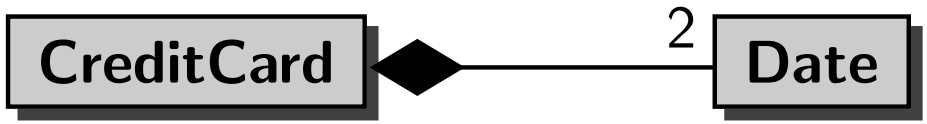
Composition

Composition is a special type of aggregation. The aggregate A and its part P form a composition if "A owns P", that is, each object of type A has exclusive access to its attribute of type P.

The designer and the implementer of a class determine whether an aggregation is a composition.

Java does not provide any special language constructs for implementing compositions. The constructors, accessors and mutators are implemented in a particular way.

UML Diagrams



`CreditCard` Object

Question

Create a `CreditCard` object with number 123456 and name Virginia Kaarthouer.

CreditCard Object

Question

Create a CreditCard object with number 123456 and name Virginia Kaarthouer.

Answer

```
int number = 123456;
String name = "Virginia Kaarthouer";
CreditCard card = new CreditCard(number, name);
```

CreditCard Object

Question

Create a `CreditCard` object with number 123456 and name Virginia Kaarthouer.

Answer

```
int number = 123456;
String name = "Virginia Kaarthouer";
CreditCard card = new CreditCard(number, name);
```

Question

Draw the memory diagram depicting memory at the end of the second line. (To save space, do not include the attributes `balance` and `limit`.)

CreditCard Object

100	main invocation	
	123456	number
	200	name
	500	card
200	String object	
	"Virginia Kaarthouer"	value
300	Date object	
	1415637359054	time
400	Date object	
	1478795881318	time
500	CreditCard object	
	123456	number
	200	name
	300	issueDate
	400	expiryDate

Accessors

Question

Create a `CreditCard` object with number 123456 and name Virginia Kaarthouer and print its expiry date. (To save space, do not include the attributes `balance` and `limit`.)

Accessors

Question

Create a `CreditCard` object with number 123456 and name Virginia Kaarthouer and print its expiry date. (To save space, do not include the attributes `balance` and `limit`.)

Answer

```
int number = 123456;
String name = "Virginia Kaarthouer";
CreditCard card = new CreditCard(number, name);
Date expiryDate = card.getExpiryDate();
output.println(expiryDate);
```

Accessors

Question

Create a `CreditCard` object with number 123456 and name Virginia Kaarthouer and print its expiry date. (To save space, do not include the attributes `balance` and `limit`.)

Answer

```
int number = 123456;
String name = "Virginia Kaarthouer";
CreditCard card = new CreditCard(number, name);
Date expiryDate = card.getExpiryDate();
output.println(expiryDate);
```

Question

Draw the memory diagram depicting memory at the end of the fourth line.

Accessors

100	main invocation	
	123456	number
	200	name
	500	card
	600	expiryDate
200	String object	
	"Virginia Kaarthouer"	value
300	Date object	
	1415637359054	time
400	Date object	
	1478795881318	time
500	CreditCard object	
	123456	number
	200	name
	300	issueDate
	400	expiryDate
600	Date object	
	1478795881318	time

Accessors

Question

Why can't `card.ExpiryDate()` return a reference to the `Date` object on address 400?

Accessors

Question

Why can't `card.ExpiryDate()` return a reference to the `Date` object on address 400?

Answer

If `card.ExpiryDate()` were to return a reference to the `Date` object on address 400, then both the `main` invocation and the `CreditCard` object would have access to that `Date` object. But the `CreditCard` object "owns" that `Date` object, because `CreditCard` and `Date` form a composition. Hence, `CreditCard` should have exclusive access to that `Date` object.

Mutators

Question

Create a `CreditCard` object with number 123456 and name Virginia Kaarthouer and set its expiry date to five years from now.

Mutators

Question

Create a `CreditCard` object with number 123456 and name Virginia Kaarthouer and set its expiry date to five years from now.

Answer

```
int number = 123456;
String name = "Virginia Kaarthouer";
CreditCard card = new CreditCard(number, name);
Calendar calendar = Calendar.getInstance();
calendar.add(Calendar.YEAR, 5);
Date expiryDate = calendar.getTime();
card.setExpiryDate();
```

Mutators

Question

Create a `CreditCard` object with number 123456 and name Virginia Kaarthouer and set its expiry date to five years from now.

Answer

```
int number = 123456;
String name = "Virginia Kaarthouer";
CreditCard card = new CreditCard(number, name);
Calendar calendar = Calendar.getInstance();
calendar.add(Calendar.YEAR, 5);
Date expiryDate = calendar.getTime();
card.setExpiryDate();
```

Question

Draw the memory diagram depicting memory at the end of the sixth line. (To save space, do not include the attributes `number`, `name`, `balance` and `limit`.)

Mutators

100	main invocation	
	123456	number
	200	name
	500	card
	600	calendar
	700	expiryDate
200	String object	
	"Virginia Kaarthouer"	value
300	Date object	
	1415637359054	time
400	Date object	
	1478795881318	time
500	CreditCard object	
	300	issueDate
	400	expiryDate
600	Calendar object	
	1415637372347	time
700	Date object	
	1415637372347	time

Mutators

Question

Draw the memory diagram depicting memory at the end of the seventh line. Draw only those objects that are relevant to the changes.

Mutators

100	main invocation	
	123456	number
	200	name
	500	card
	600	calendar
	700	expiryDate
300	Date object	
	1415637359054	time
400	Date object	
	1478795881318	time
500	CreditCard object	
	300	issueDate
	800	expiryDate
700	Date object	
	1415637372347	time
800	Date object	
	1415637372347	time

Mutators

Question

Why can't we set the `expiryDate` attribute to refer to the `Date` object on address 700?

Mutators

Question

Why can't we set the `expiryDate` attribute to refer to the `Date` object on address 700?

Answer

If the `expiryDate` attribute were to refer to the `Date` object on address 700, then both the `main` invocation and the `CreditCard` object would have access to that `Date` object. But the `CreditCard` object "owns" that `Date` object, because `CreditCard` and `Date` form a composition. Hence, `CreditCard` should have exclusive access to that `Date` object.

Composition

Question

Write some code that shows that `CreditCard` and `Date` form a composition.

Composition

Question

Write some code that shows that `CreditCard` and `Date` form a composition.

Answer

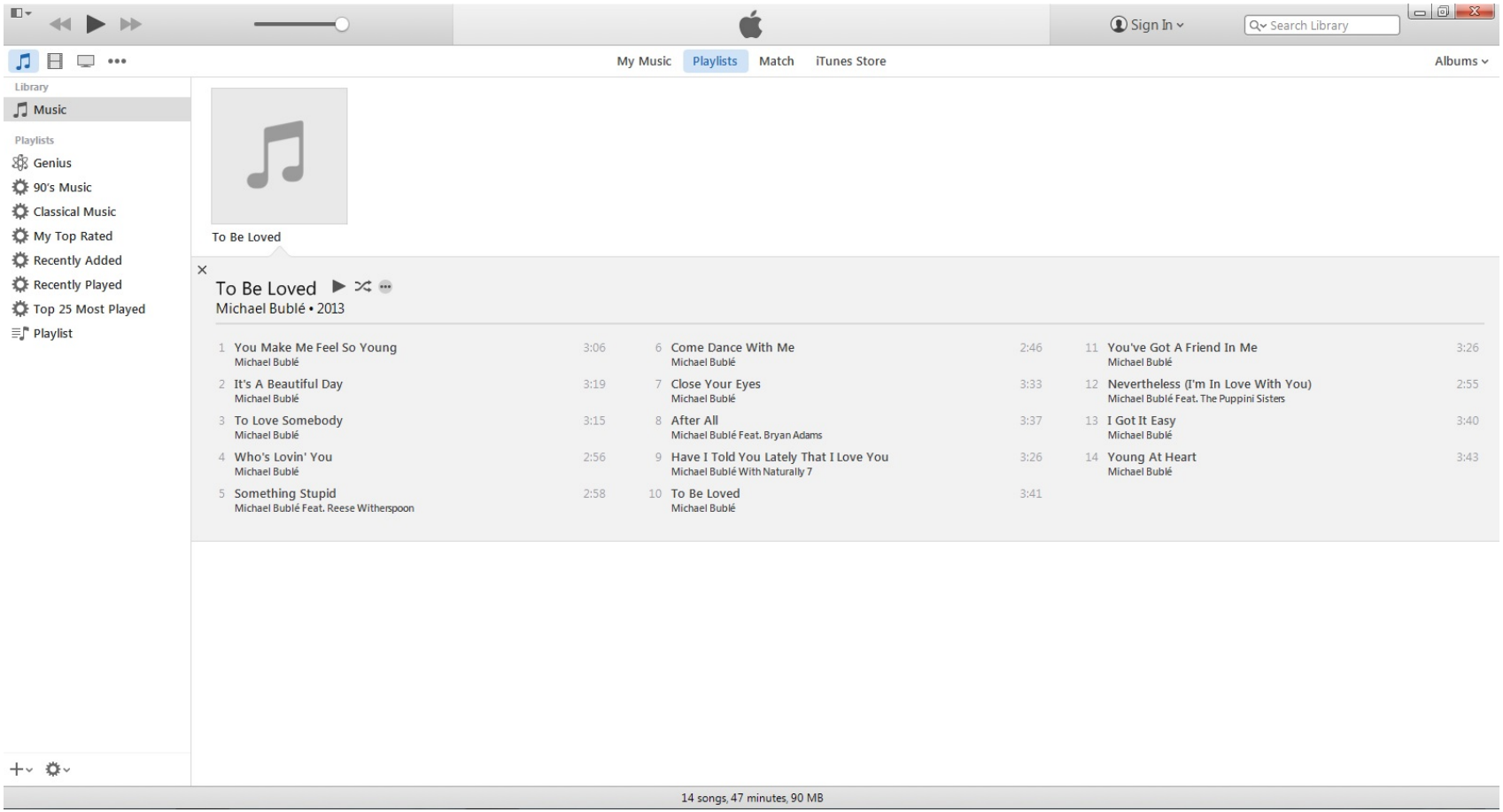
```
int number = 123456;
String name = "Virginia Kaarthouer";
Creditcard card = new Creditcard(number, name);
Date now = new Date();
card.setIssueDate(now);
Date issueDate = card.getIssueDate();
output.println(now == issueDate);
// prints false when a composition
```

Collections

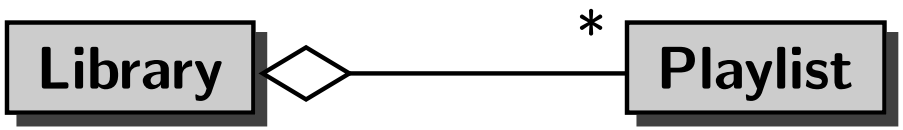
CSE 5910

www.cse.yorku.ca/course/5910

ITunes


Sign In
Search Library
My Music
Playlists
Match
iTunes Store
Albums
Library
Music
Playlists
Genius
90's Music
Classical Music
My Top Rated
Recently Added
Recently Played
Top 25 Most Played
Playlist
To Be Loved
To Be Loved
Michael Bublé • 2013
1 You Make Me Feel So Young 3:06
Michael Bublé
2 It's A Beautiful Day 3:19
Michael Bublé
3 To Love Somebody 3:15
Michael Bublé
4 Who's Lovin' You 2:56
Michael Bublé
5 Something Stupid 2:58
Michael Bublé Feat. Reese Witherspoon
6 Come Dance With Me 2:46
Michael Bublé
7 Close Your Eyes 3:33
Michael Bublé
8 After All 3:37
Michael Bublé Feat. Bryan Adams
9 Have I Told You Lately That I Love You 3:26
Michael Bublé With Naturally 7
10 To Be Loved 3:41
Michael Bublé
11 You've Got A Friend In Me 3:26
Michael Bublé
12 Nevertheless (I'm In Love With You) 2:55
Michael Bublé Feat. The Puppini Sisters
13 I Got It Easy 3:40
Michael Bublé
14 Young At Heart 3:43
Michael Bublé
14 songs, 47 minutes, 90 MB

UML Diagrams

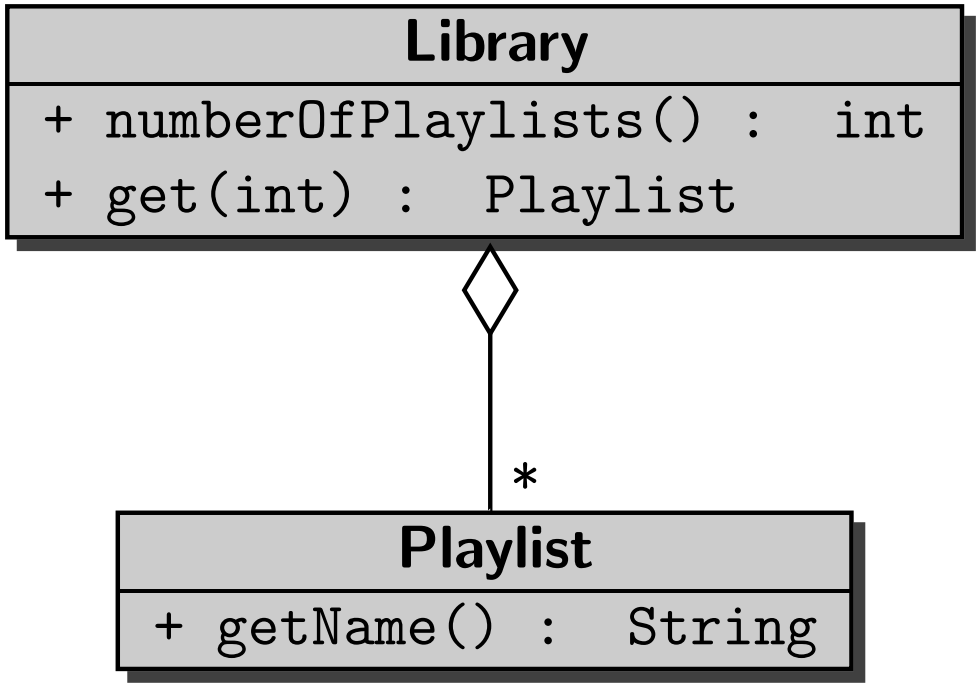


ITunes

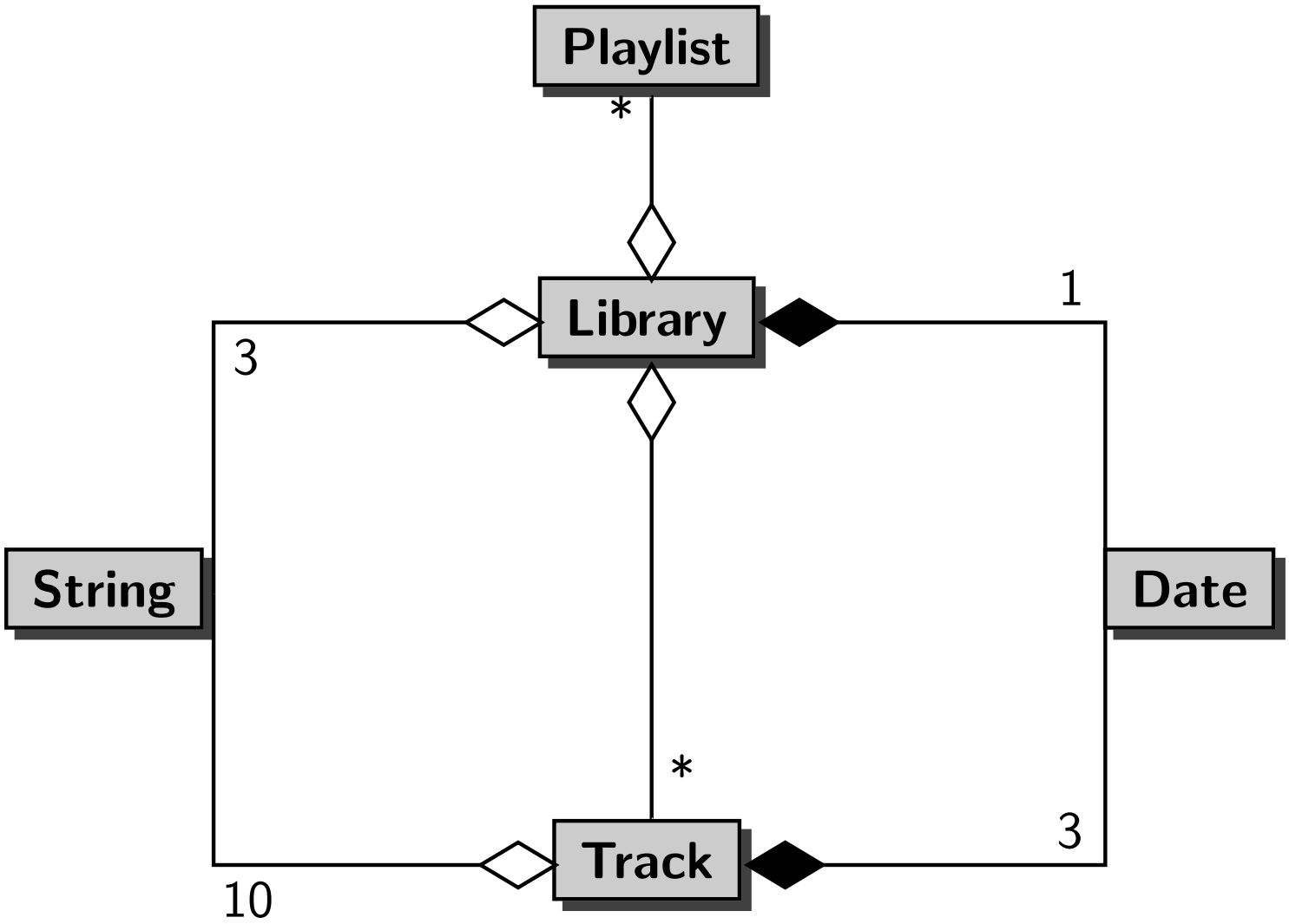
Question

Create a `Library` object from an `iTunes Library.xml` file. For each `Playlist` of the `Library` print its name on a separate line.

UML Diagrams



UML Diagrams



ITunes

Question

Create a `Library` object from an `iTunes Library.xml` file. For each `Playlist` of the `Library` print its name on a separate line and print the names of all its `Tracks` on a separate line followed by an empty line.

ITunes

Question

Create a `Library` object from an `iTunes Library.xml` file. Print the total amount of time the `Library` has played in milliseconds. You may assume that each track occurs in one playlist.