Homework 4
Write up

Brian Kennedy
Pete Koehn
Theodore Lindsey

November 19, 2014

0.1 hw04.py

```
from cController import Controller

Controller()
```

0.2 cController.py

```
class Controller:
    def __init__(self):
        from cView import View
        from cModel import Model

        model = Model()
        view = View()

        while True:
            action = view.menu()

            if action == "quit":
                quit()
            elif action[0] == "r":
                self.RemoveFromOrder(model, int(action[1:]))
            elif action[0] == "a":
                self.AddToOrder(model, int(action[1:]))
            elif action[0] == "d":
                self.DisplayBook(model, view, action[1:])
            elif action == "order":
                self.GetOrderList(model, view)
            elif action == "cost":
                self.GetOrderCost(model,view)


    def DisplayBook(self, model, view, genre):
        BookList = model.GetBookList(genre)
        view.GeneralDisplay(BookList)

    def AddToOrder(self, model, BookID):
        model.AddToOrder(BookID)
        return "Added book #" + str(BookID) + " to the order."

    def RemoveFromOrder(self, model, BookID):
        model.RemoveFromOrder(BookID)
```

```
        return "Removed book #" + str(BookID) + " from the order."

    def GetOrderCost(self, model, view):
        cost = model.CalculateOrderCost()
        view.GeneralDisplay(cost)

    def GetOrderList(self, model, view):
        OrderList = model.GetOrder()
        view.GeneralDisplay(OrderList)
```

0.3 cView.py

```
class View:
    #--Display Books--
    def DisplayBooks(self):
        print("Select genre:")
        print("1.     Science Fiction")
        print("2.     Travel")
        print("3.     Software Engineering")
        uchoice = input("Please input the number of your choice: "
            )
        print(uchoice)
        print()

        if uchoice == "1":
            return "dSciFi"
        elif uchoice == "2":
            return "dTravel"
        elif uchoice == "3":
            return "dSoftware"

    #--Add Book to Order--
    def AddToOrder(self):
        uchoice = input("Please input BookID: ")
        if (1 <= int(uchoice) <= 60):
            return "a" + str(uchoice)
        else:
            print("Invalid BookID.")
            return self.AddToOrder()

        #^ this will pass bookid to other thing

    #--Remove Book from Order--
```

```
    def RemoveFromOrder(self):
        uchoice = input("Please input BookID: ")
        if (1 <= int(uchoice) <= 60):
            return "r" + str(uchoice)
        else:
            print("Invalid BookID.")
            return self.RemoveFromOrder()

    def GeneralDisplay(self, array):
        for entry in array:
            print(entry)

    #--menu---
    def menu(self):
        print()
        print ("Avalible actions:")
        print ("1.     Display Books")
        print ("2.     Display Current Order")
        print ("3.     Add Book to Order")
        print ("4.     Remove Book from Order")
        print ("5.     Calculate Order Cost")
        print ("6.     Quit")
        uchoice = input("Please input the number of your choice: "
            )
        print("")
        if uchoice == '1':
            return self.DisplayBooks()
        elif uchoice == '2':
            return "order"
        elif uchoice == '3':
            return self.AddToOrder()
        elif uchoice == '4':
            return self.RemoveFromOrder()
        elif uchoice == '5':
            return "cost"#self.DisplayOrderCost()
        elif uchoice == '6':
            return "quit"
        else:
            print("Invalid input.")
```

0.4 cModel.py

```
#! /usr/bin/python
```

```
class Model:
    def __init__(self):
        self.CurrentOrder = [];
        self.cost = 0;

        self.titles = [0] * 61
        self.authors = [0] * 61

        # Science Fiction titles, $50
        self.titles[1] = 'Dune [S1]'
        self.titles[2] = 'Ender\'s Game [S1]'
        self.titles[3] = 'The Foundation Trilogy'
        self.titles[4] = 'Hitch Hiker\'s Guide to the Galaxy [S1]'
        self.titles[5] = '1984'
        self.titles[6] = 'Stranger in a Strange Land'
        self.titles[7] = 'Fahrenheit 451'
        self.titles[8] = '2001: A Space Odyssey'
        self.titles[9] = 'Do Androids Dream of Electric Sheep?'
        self.titles[10] = 'Neuromancer [S1]'
        self.titles[11] = '[C] I, Robot'
        self.titles[12] = 'Starship Troopers'
        self.titles[13] = 'Ringworld [S1]'
        self.titles[14] = 'Rendezvous With Rama'
        self.titles[15] = 'Hyperion [S1]'
        self.titles[16] = 'Brave New World'
        self.titles[17] = 'The Forever War'
        self.titles[18] = 'The Time Machine'
        self.titles[19] = 'Childhood\'s End'
        self.titles[20] = 'The Moon is a Harsh Mistress'

        # Travel titles, $40
        self.titles[21] = 'A Dragon Apparent'
        self.titles[22] = 'A House in Bali'
        self.titles[23] = 'A Moveable Feast'
        self.titles[24] = 'A Short Walk in the Hindu Kush'
        self.titles[25] = 'A Time of Gifts'
        self.titles[26] = 'A Turn in the South'
        self.titles[27] = 'A Walk in the Woods'
        self.titles[28] = 'A Winter in Arabia'
        self.titles[29] = 'Among the Russians'
        self.titles[30] = 'An Area of Darkness'
        self.titles[31] = 'Arabian Sands'
        self.titles[32] = 'Arctic Dreams'
        self.titles[33] = 'The Art of Travel'
        self.titles[34] = 'As I Walked Out One Midsummer Morning'
```

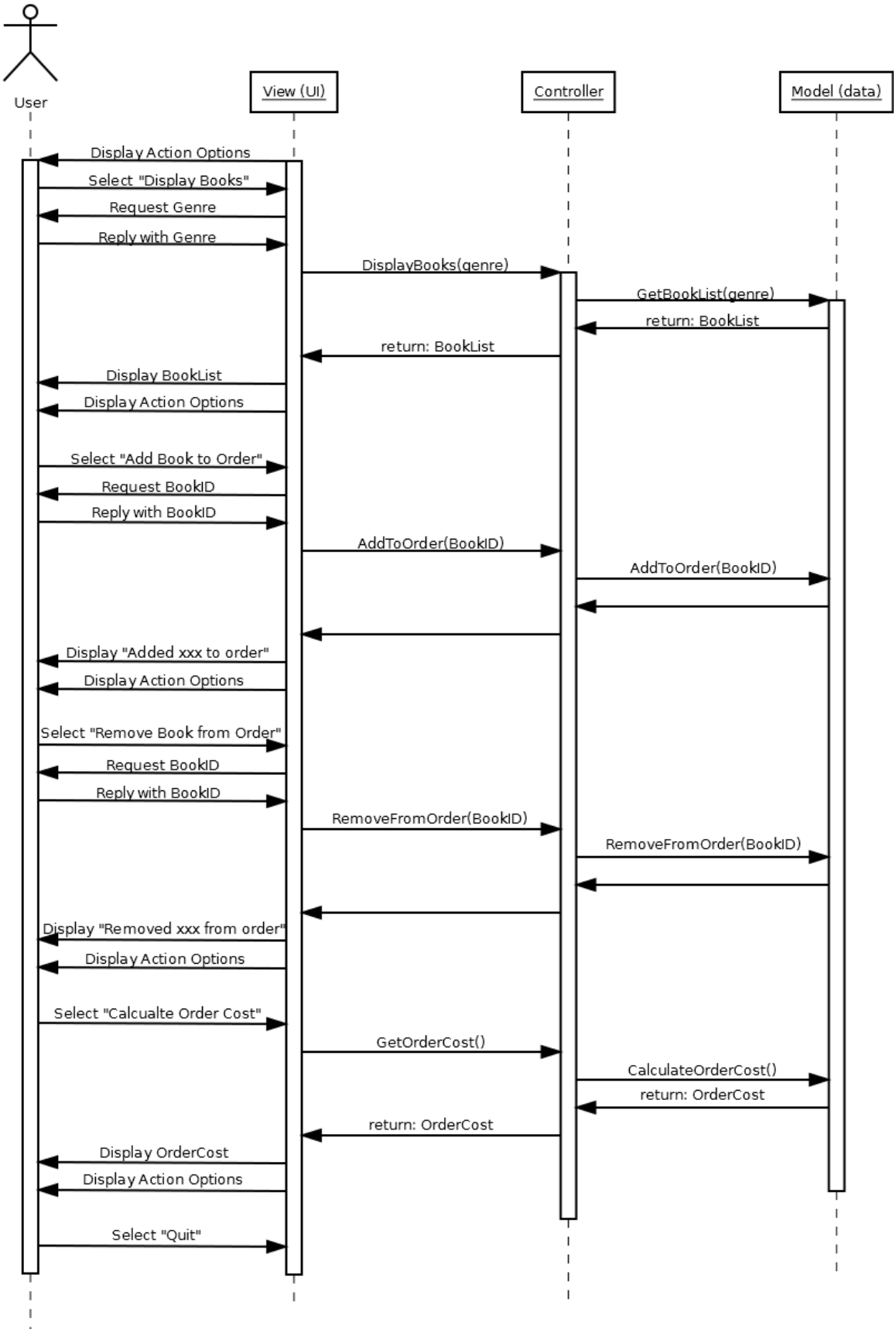
```
        self.titles[35] = 'Baghdad Without a Map'
        self.titles[36] = 'Balkan Ghosts'
        self.titles[37] = 'Beyond Euphrates'
        self.titles[38] = 'The Bird Man and the Lap Dancer'
        self.titles[39] = 'Bitter Lemons of Cyprus'
        self.titles[40] = 'Black Lamb and Grey Falcon'

        # Software Engineering titles, $100
        self.titles[41] = 'Code Complete: A Handbook of Software
            Construction'
        self.titles[42] = 'Head First Design Patterns'
        self.titles[43] = 'Rapid Development'
        self.titles[44] = 'Design Patterns: Elements of Reusable
            Object-Oriented Software'
        self.titles[45] = 'Cryptography: Protocols, Algorithms,
            and Source Code'
        self.titles[46] = 'Agile Software Development: Principles,
            Patterns and Practices'
        self.titles[47] = 'Joel on Software'
        self.titles[48] = 'Peopleware: Productive Projects and
            Teams'
        self.titles[49] = 'The Mythical Man-Month, Anniversary
            Edition'
        self.titles[50] = 'Refactoring: Improving the Design of
            Existing Code'
        self.titles[51] = 'Agile Estimating and Planning'
        self.titles[52] = 'Writing Effective Use Cases'
        self.titles[53] = 'Object-Oriented Software Construction'
        self.titles[54] = 'Software Estimation: Demystifying the
            Black Art'
        self.titles[55] = 'User Stories Applied: For Agile
            Software Development'
        self.titles[56] = 'The Art of Computer Programming'
        self.titles[57] = 'Patterns of Enterprise Application
            Architecture'
        self.titles[58] = 'Mastering Regular Expressions'
        self.titles[59] = 'The Pragmatic Programmer'
        self.titles[60] = 'Software Requirements'

        # Science Fiction titles, $50
        self.authors[1] = 'Frank Herbert'
        self.authors[2] = 'Orson Scott Card'
        self.authors[3] = 'Isaac Asimov'
        self.authors[4] = 'Douglas Adams'
        self.authors[5] = 'George Orwell'
```

```
        self.authors[6] = 'Robert A Heinlein'
        self.authors[7] = 'Ray Bradbury'
        self.authors[8] = 'Arthur C Clarke'
        self.authors[9] = 'Philip K Dick'
        self.authors[10] = 'William Gibson'
        self.authors[11] = 'Isaac Asimov'
        self.authors[12] = 'Robert A Heinlein'
        self.authors[13] = 'Larry Niven'
        self.authors[14] = 'Arthur C Clarke'
        self.authors[15] = 'Dan Simmons'
        self.authors[16] = 'Aldous Huxley'
        self.authors[17] = 'Joe Haldeman'
        self.authors[18] = 'H G Wells'
        self.authors[19] = 'Arthur C Clarke'
        self.authors[20] = 'Robert A Heinlein'

        # Travel titles, $40
        self.authors[21] = 'Norman Lewis'
        self.authors[22] = 'Colin McPhee'
        self.authors[23] = 'Ernest Hemingway'
        self.authors[24] = 'Eric Newby'
        self.authors[25] = 'Patrick Leigh Fermor'
        self.authors[26] = 'V.S. Naipaul'
        self.authors[27] = 'Bill Bryson'
        self.authors[28] = 'Freya Stark'
        self.authors[29] = 'Colin Thubron'
        self.authors[30] = 'V.S. Naipaul'
        self.authors[31] = 'Wilfred Thesiger'
        self.authors[32] = 'Barry Lopez'
        self.authors[33] = 'Alain de Botton'
        self.authors[34] = 'Laurie Lee'
        self.authors[35] = 'Tony Horwitz'
        self.authors[36] = 'Robert D. Kaplan'
        self.authors[37] = 'Freya Stark'
        self.authors[38] = 'Eric Hansen'
        self.authors[39] = 'Lawrence Durrell'
        self.authors[40] = 'Rebecca West'

        # Software Engineering titles, $100
        self.authors[41] = 'Steve McConnell'
        self.authors[42] = 'Elisabeth Freeman'
        self.authors[43] = 'Steve McConnell'
        self.authors[44] = 'Erich Gamma'
        self.authors[45] = 'Bruce Schneier'
        self.authors[46] = 'Robert C. Martin'
```

```
        self.authors[47] = 'Joel Spolsky'
        self.authors[48] = 'Tom DeMarco'
        self.authors[49] = 'Frederick P. Brooks'
        self.authors[50] = 'Martin Fowler'
        self.authors[51] = 'Mike Cohn'
        self.authors[52] = 'Alistair Cockburn'
        self.authors[53] = 'Bertrand Meyer'
        self.authors[54] = 'Steve McConnell'
        self.authors[55] = 'Mike Cohn'
        self.authors[56] = 'Donald E. Knuth'
        self.authors[57] = 'Martin Fowler'
        self.authors[58] = 'Jeffrey Friedl'
        self.authors[59] = 'Andrew Hunt'
        self.authors[60] = 'Karl E. Wiegers'

    def GetBookList(self, genre):
        BookList = ["ID\tAuthor\t\t\tTitle"]
        if genre == "SciFi":
            for BookID in range(1,20+1):
                BookList.append(str(BookID) + "\t" + self.authors[
                    BookID] + "\t\t\t" + self.titles[BookID])
        elif genre == "Travel":
            for BookID in range(21,40+1):
                BookList.append(str(BookID) + "\t" + self.authors[
                    BookID] + "\t\t\t" + self.titles[BookID])
        elif genre == "Software":
            for BookID in range(41,60+1):
                BookList.append(str(BookID) + "\t" + self.authors[
                    BookID] + "\t\t\t" + self.titles[BookID])
        return BookList

    def GetOrder(self):
        OrderList = ["Current order contents:"]
        OrderList.append("ID\tAuthor\t\t\tTitle")
        for BookID in self.CurrentOrder:
            OrderList.append(str(BookID) + "\t" + self.authors[
                BookID] + "\t\t\t" + self.titles[BookID])
        return OrderList


    def AddToOrder(self, BookID):
        self.CurrentOrder.append(BookID)
        self.CurrentOrder.sort()

    def RemoveFromOrder(self, BookID):
```

```
        try:
            self.CurrentOrder.remove(BookID)
        except:
            print("Book ID " + str(BookID) + " not in current
                order")

    def CalculateOrderCost(self):
        self.cost = 0
        for i in self.CurrentOrder:

            #print("Cost:    " + str(self.cost))
            #print("BookID: " + str(i))
            #print()
            if i < 21:
                self.cost += 50
            elif i < 41:
                self.cost += 40
            else:
                self.cost += 100
        return ["Order cost is $" + str(self.cost)]
```

Figure 1: Sequence Diagram



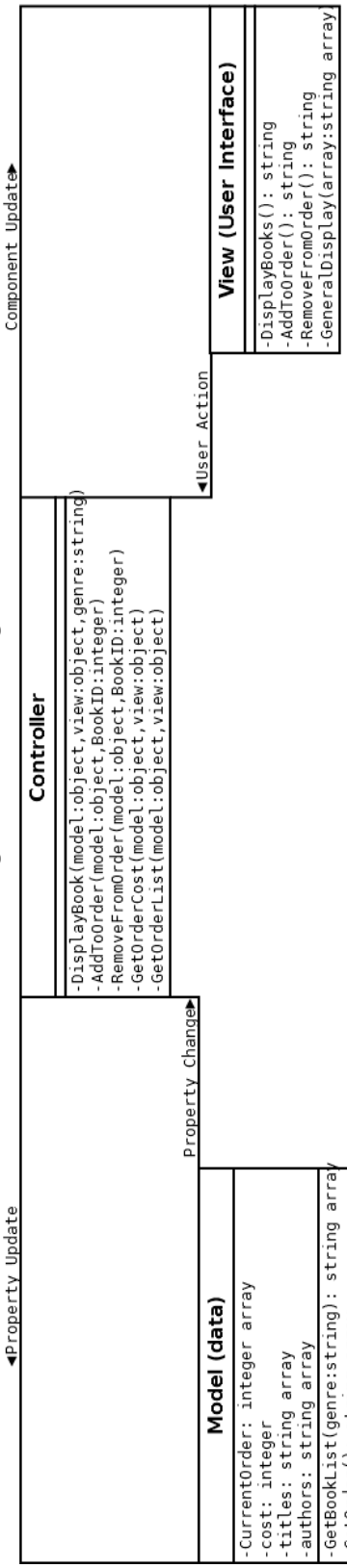


Figure 2: Class Diagram