

รายงานประกอบโครงงาน

รายวิชา 2603279 Object-Oriented Programming

เรื่อง

โปรแกรมการจัดการเอกสารและจดหมายเหตุ

สำหรับกิจการการจัดการเอกสารและจดหมายเหตุ First Archives

เสนอ

อาจารย์อัมภาพร ทรัพย์สมบูรณ์

โดย

สหัสวรรษ คำสุข 6142088026

28 พฤศจิกายน 2562

ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย

ภาคการศึกษาตอนต้น ปีการศึกษา 2562

คำนำ

รายงานเล่มนี้เป็นส่วนหนึ่งของโครงการ “โปรแกรมการจัดการเอกสารและจดหมายเหตุ สำหรับกิจการจัดการเอกสารและจดหมายเหตุ **First Archives**” ซึ่งเป็นงานมอบหมายในรายวิชา 2603279 Object-Oriented Programming การจัดทำโครงงานนี้มีจุดประสงค์เพื่อนำความรู้ที่ได้จากการศึกษาในรายวิชามาประยุกต์ใช้ในการทำความเข้าใจกระบวนการทำงานของธุรกิจในยุคปัจจุบัน และจัดทำโปรแกรมที่แสดงถึงขั้นตอนการทำงานในธุรกิจดังกล่าวด้วยวิธีการโปรแกรมเชิงวัตถุ

หัวข้อและธุรกิจที่นิสิตเลือกนำมาใช้เป็นกรณีศึกษาคือธุรกิจการจัดการเอกสารและจดหมายเหตุ ซึ่งเป็นกิจการที่ให้บริการในการรับฝาก จัดเก็บ และดูแลเอกสารจากหน่วยงานหรือบุคคลทั่วไปรวมถึงให้บริการอื่นที่เกี่ยวข้อง เนื่องจากว่านิสิตมีความสนใจในเรื่องจดหมายเหตุศึกษา (archives studies) จึงเลือกศึกษาธุรกิจดังกล่าว

ความรู้และทักษะที่ได้จากการศึกษาในรายวิชานี้ (หรือศาสตร์ด้านเทคโนโลยีสารสนเทศ) นิสิตประสงค์ที่จะนำไปประยุกต์ใช้ร่วมกับความรู้และทักษะจากศาสตร์ด้านมนุษยศาสตร์ สังคมศาสตร์ เพื่อที่จะสามารถบูรณาการความรู้และแสดงความเชื่อมโยงในศาสตร์ที่นิสิตสนใจ อันเป็นจุดมุ่งหมายในการศึกษาของนิสิต

การทำโครงงานนี้คงไม่อาจสามารถล่วงได้หากปราศจากความช่วยเหลือและคำปรึกษาจากอาจารย์อัมภพร ทรัพย์สมบูรณ์ อาจารย์ผู้สอนและอาจารย์ที่ปรึกษาประจำรายวิชา นอกจากนี้ นิสิตต้องขอขอบคุณอาจารย์นยา สุจนายา อาจารย์ประจำภาควิชาบรรณารักษศาสตร์ คณะอักษรศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้แนวทางและคำปรึกษาในการจัดการจดหมายเหตุ นิสิตขอขอบคุณอาจารย์ทั้งสอนท่านเป็นอย่างยิ่ง และหากอาจารย์ยินดี นิสิตหวังว่าจะได้ขอรับคำปรึกษาจากอาจารย์อีกในภายภาคหน้า

สหัสวรรษ คำสุข

สารบัญ

คำนำ	ก
1. บทนำ	1
1.1. การศึกษาจากกระบวนการดำเนินงานธุรกิจ (Business Process)	1
1.2. การวิเคราะห์รูปแบบการใช้งาน (Use Case Analysis)	2
1.3. แผนภาพแสดงคลาส (Class Diagram)	3
2. วิธีการทดสอบโปรแกรมและผลการทดสอบโปรแกรม	4
3. บทสรุป	6
บรรณานุกรม	7
ภาคผนวก	8

1. บทนำ

ธุรกิจการจัดการ / การจัดเก็บ / คลังเอกสาร เป็นธุรกิจรูปแบบหนึ่งที่ให้บริการรับฝาก จัดเก็บ และดูแลเอกสารจากหน่วยงานหรือบุคคลทั่วไป นอกจากนี้ยังอาจให้บริการอื่นที่เกี่ยวข้อง เช่น การทำลายเอกสาร การแปลงให้เป็นดิจิทัล (digitalization) ธุรกิจเหล่านี้เกิดขึ้นมาเพื่อรองรับเอกสารจากกิจการหรือหน่วยงานอื่นที่มีจำนวนมากเกินหรือซับซ้อนเกินกว่าที่หน่วยงานนั้นจะจัดการไหว ทั้งช่วยจัดจำแนกหมวดหมู่เอกสารให้เป็นระบบเพื่อให้ง่ายต่อการเข้าถึง และป้องกันดูแลไม่ให้เอกสารเกิดการชำรุดเสียหายหรือสูญหายไป

1.1. กรณีศึกษาจากกระบวนการดำเนินธุรกิจ (Business Process)

บริษัท First Archives ให้บริการจัดเก็บเอกสาร (documents management) แก่ลูกค้าซึ่งเป็นบุคคลทั่วไป หน่วยงานหรือองค์กรต่างๆ ทั้งนี้บริษัทจะรับจัดเก็บเฉพาะแค่เอกสารที่เป็นกระดาษเท่านั้นการให้บริการจัดเก็บเอกสารนั้นจะเป็นการให้เช่ากล่อง ซึ่งแบ่งเป็นสองประเภท คือ กล่องสำหรับเอกสารทั่วไป และกล่องสำหรับเอกสารทางกฎหมาย (เอกสารซึ่งเป็นหลักฐานแสดงสิทธิและความรับผิดชอบระหว่างคู่กรณีสองฝ่าย เช่น สัญญาคู่ค้า, พันัยกรรม, สัญญาการค้า, โฉนด ฯลฯ)

เมื่อลูกค้ามาใช้บริการ พนักงานจะรับแจ้งชื่อลูกค้า (หรือชื่อหน่วยงาน) เบอร์โทรศัพท์สำหรับติดต่อ ที่อยู่ แล้วจึงกำหนด หมายเลขประจำตัวให้ลูกค้า จากนั้นจึงรับแจ้งระยะเวลา (จำนวนเดือน) ที่ต้องการจัดเก็บ พนักงานจะรับเอกสารที่ลูกค้าต้องการจัดเก็บ แล้วจึงแยกหมวดหมู่เอกสารและบันทึกจำนวนเอกสาร ถัดมาพนักงานจึงจัดเตรียมกล่องซึ่งระบุหมายเลขกล่อง ประเภทกล่อง โซนที่จัดเก็บ ระยะเวลาจัดเก็บ วันที่เริ่มจัดเก็บ และวันสิ้นสุดการจัดเก็บ จากนั้นจึงบันทึกข้อมูลเอกสารซึ่งประกอบด้วย หมายเลขอ้างอิงเอกสาร ชื่อเอกสาร และจำนวนแผ่นกระดาษในเอกสารนั้นแล้วจึงบรรจุใส่กล่อง เมื่อจัดเก็บเอกสารครบแล้ว โปรแกรมจะคำนวณแล้วแจ้งค่าบริการทั้งหมด ถัดมาพนักงานจะออกไปส่งเอกสารการจัดเก็บซึ่งระบุหมายเลขสัญญา ชื่อลูกค้า เบอร์โทรศัพท์ ที่อยู่ ระยะเวลาจัดเก็บ วันที่เริ่มจัดเก็บ และวันสิ้นสุดการจัดเก็บ

เอกสารทั่วไป จัดเก็บในโซน A-D		เอกสารทางกฎหมาย จัดเก็บในโซน E-H	
จำนวนเอกสารที่เก็บ	ราคา (บาท) / เดือน	จำนวนเอกสารที่เก็บ	ราคา (บาท) / เดือน
น้อยกว่า 4	125	น้อยกว่า 4	250
4-5	250	4-5	500
มากกว่า 6	500	มากกว่า 6	1000
การคิดค่าบริการจะคำนวณจากอัตราค่าบริการ และระยะเวลาที่ฝาก			
* สำหรับเอกสารทางกฎหมาย จะมีการคิดค่าธรรมเนียมความปลอดภัยแบบพิเศษ เดือนละ 1500 บาท			
ลูกค้าคนเดียวกันเมื่อมาใช้บริการจัดเก็บเอกสารในครั้งถัดไป ระบบจะถือว่าเป็นลูกค้าใหม่			

อัตราและนโยบายการคิดค่าบริการ(บาท)

นอกจากนี้ บริษัทยังให้บริการคั่นคืนและจัดส่งเอกสาร เมื่อลูกค้าต้องการรับเอกสารคืน ลูกค้าจะต้องแจ้งเลขที่ใบสัญญา จากนั้นพนักงานจะรับรายชื่อของเอกสารที่ลูกค้าต้องการคั่นคืน เมื่อคั่นคืนเอกสารครบแล้ว จึงจัดส่งเอกสารไปตามที่อยู่ของลูกค้าที่ได้แจ้งไว้

1.2. การวิเคราะห์รูปแบบการใช้งาน (Use Case Analysis)

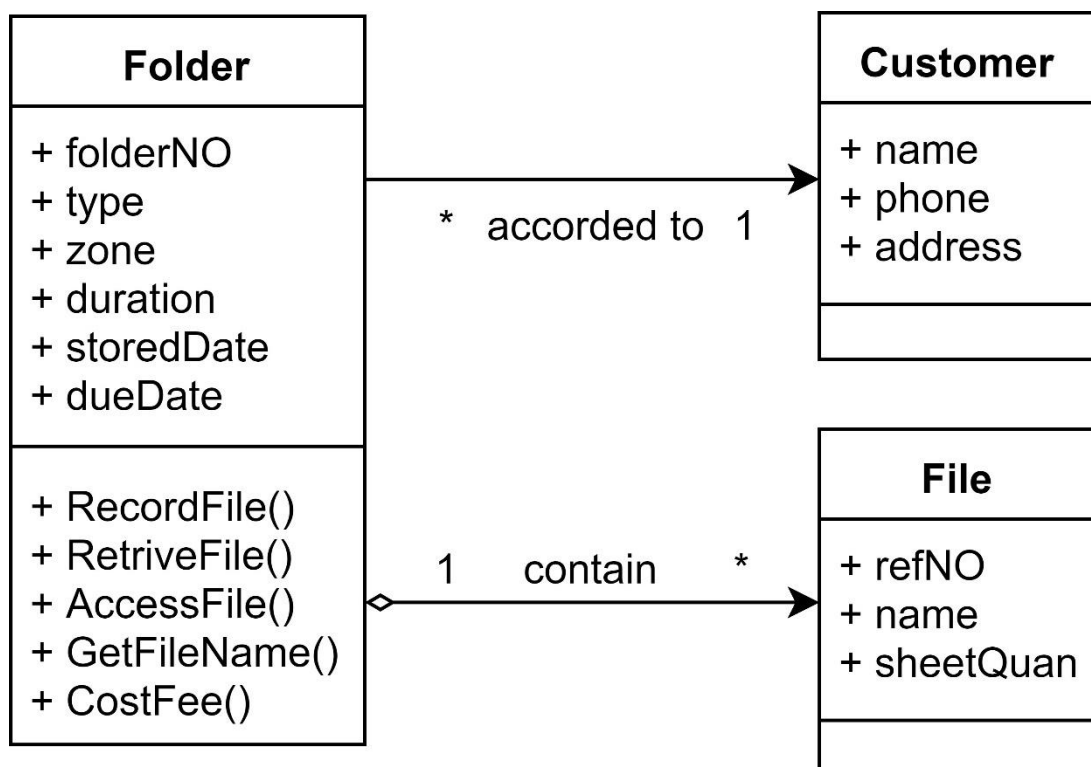
Use Case : รับจัดเก็บเอกสาร

- Primary Actor : พนักงาน
- Overview : ลูกค้าแจ้งข้อมูลของลูกค้าและระยะเวลาในการจัดเก็บ พนักงานรับเอกสารจากลูกค้า จัดเตรียมกล่องแล้วบรรจุเอกสารลงในกล่องจนครบ จากนั้นพนักงานจึงแจ้งค่าบริการทั้งหมด และออกไปส่งเอกสารการจัดเก็บเอกสารให้ลูกค้า
- Main Case Scenario : การรับจัดเก็บเอกสารมีขั้นตอนดังนี้
 1. ลูกค้าแจ้งชื่อลูกค้า (หรือชื่อหน่วยงาน) เบอร์โทรศัพท์ ที่อยู่
 2. พนักงานบันทึกข้อมูลลูกค้า และกำหนดหมายเลขประจำตัวลูกค้า
 3. ไปเก็บบันทึกข้อมูลลูกค้า
 4. พนักงานบันทึกจำนวนเอกสารแต่ละประเภท
 5. พนักงานบันทึกข้อมูลเกี่ยวกับกล่องซึ่งระบุหมายเลขกล่อง ประเภทกล่อง โชนที่จัดเก็บ ระยะเวลาจัดเก็บ วันที่เริ่มจัดเก็บ และวันสิ้นสุดการจัดเก็บ
 6. โปรแกรมบันทึกข้อมูลกล่องที่จัดเก็บเอกสาร
 7. พนักงานบันทึกข้อมูลเอกสารซึ่งประกอบด้วย หมายเลขอ้างอิงเอกสาร ชื่อเอกสาร และจำนวนแผ่นกระดาษ แล้วนำบรรจุลงในกล่อง
 8. โปรแกรมบันทึกข้อมูลเอกสาร
 9. ทำข้อ 7.-8. ซ้ำจนครบจำนวนเอกสารที่ระบุไว้
 10. โปรแกรมคำนวณค่าบริการตามระยะเวลาที่จัดเก็บและจำนวนกล่องที่ใช้ แล้วจึงแจ้งให้ลูกค้าทราบ
 11. โปรแกรมพิมพ์ใบสัญญาการจัดเก็บเอกสาร ซึ่งระบุหมายเลขสัญญา ชื่อลูกค้า เบอร์โทรศัพท์ ที่อยู่ ระยะเวลาจัดเก็บ วันที่เริ่มจัดเก็บ และวันสิ้นสุดการจัดเก็บ
 12. พนักงานมอบใบสัญญาการจัดเก็บเอกสารให้ลูกค้า
- Extensions Scenario : กรณีอื่นที่อาจเกิดขึ้นได้
 - (i) ข้อ 11. ถ้าเอกสารนั้นเป็นเอกสารทางกฎหมาย ให้คิดค่าบริการการรักษาความปลอดภัยแบบพิเศษเข้าไปด้วย

Use Case : การเรียกคืนเอกสาร

- Primary Actor : พนักงาน
- Overview : ลูกค้าแจ้งความประสงค์ขอเรียกคืนเอกสาร และแจ้งเอกสารที่ต้องการคืนคืนพนักงาน พนักงานคืนคืนเอกสารและจัดส่งให้ตามที่อยู่
- Main Case Scenario : การรับจัดเก็บเอกสารมีขั้นตอนดังนี้
 1. ลูกค้าแจ้งเลขที่ใบสัญญา
 2. พนักงานบันทึกเลขที่ใบสัญญาลงโปรแกรม
 3. โปรแกรมแสดงกล่องที่ลูกค้าเป็นเจ้าของ
 4. พนักงานรับแจ้งและบันทึกหมายเลขของกล่องที่ต้องการคืนคืนเอกสาร และจำนวนเอกสารที่ต้องการคืนคืนในกล่องนั้นจากลูกค้าลงโปรแกรม
 5. พนักงานรับแจ้งหมายเลขอ้างอิงเอกสารที่ต้องการคืนคืนจากลูกค้า และบันทึกข้อมูลดังกล่าว
 6. โปรแกรมคืนคืนเอกสารตามหมายเลขอ้างอิงเอกสาร
 7. ทำข้อ 5.-6. ซ้ำจนครบจำนวนเอกสารที่ลูกค้าต้องการคืนคืน
 8. ทำข้อ 4.-7. ซ้ำจนครบจำนวนกล่องที่ลูกค้าต้องการคืนคืน
 9. พนักงานจัดส่งเอกสารทั้งหมดที่คืนคืนไปตามที่อยู่ของลูกค้า

1.3. แผนภาพแสดงคลาส (Class Diagram)



2. วิธีการทดสอบโปรแกรมและผลการทดสอบโปรแกรม

กรณีทดสอบสำหรับการจัดเก็บเอกสาร

กรณีที่	คำอธิบาย	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ปรากฏ	ผลลัพธ์
1	ฝากแค่เอกสารทั่วไป	Quantity of General File <i>Input n > 0</i>	อนุญาตให้มีการฝากแค่เอกสารทั่วไปเท่านั้น	ผลลัพธ์เป็นไปตามที่คาดหวัง	ผ่าน
		Quantity of General File <i>Input n = 0</i>			
2	ฝากแค่เอกสารทางกฎหมาย	Quantity of General Files <i>Input n = 0</i>	อนุญาตให้มีการฝากแค่เอกสารทางกฎหมายเท่านั้น	ผลลัพธ์เป็นไปตามที่คาดหวัง	ผ่าน
		Quantity of Legal Files <i>Input n > 0</i>			
3	ฝากเอกสารทั้งสองประเภท	Quantity of General File <i>Input n > 0</i>	ให้มีการฝากเอกสารทั้งสองประเภท	ผลลัพธ์เป็นไปตามที่คาดหวัง	ผ่าน
		Quantity of Legal Files <i>Input n > 0</i>			

กรณีทดสอบสำหรับการค้นคืนเอกสาร

กรณีที่	คำอธิบาย	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ปรากฏ	ผลลัพธ์
1	ค้นคืนเอกสารบางชิ้นจากกล่องเอกสารที่ถูกค้าเป็นเจ้าของ	Folder NO. <i>Input n n = a, a is folder no. the customer own.</i>	โปรแกรมแจ้งเอกสารที่ค้นคืนและนำเอกสารนั้นออกจากระบบ	ผลลัพธ์เป็นไปตามที่คาดหวัง	ผ่าน
		Number of Files Retrieved <i>Input n n < m, m is total files.</i>			

กรณีที	คำอธิบาย	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่ปรากฏ	ผลลัพธ์
2	ค้นคืนเอกสารทั้งหมดจากกล่องเอกสารที่ถูกค้าเป็นเจ้าของ	Folder NO. <i>Input n n = a</i> , a is folder no. the customer own. Number of Files Retrieved <i>Input n n = m</i> , m is total files.	โปรแกรมแจ้งเอกสารที่ค้นคืนและนำเอกสารนั้นออกจากระบบ	ผลลัพธ์เป็นไปตามที่คาดหวัง	ผ่าน
3	ค้นคืนเอกสารที่ไม่ได้ถูกจัดเก็บจากกล่องที่ไม่ได้บรรจุเอกสาร	Folder NO. <i>Input n n is any integer.</i> File NO. <i>Input n n is not in File NO.</i>	โปรแกรมแจ้งเอกสารที่ค้นคืนและนำเอกสารนั้นออกจากระบบ	โปรแกรมหยุดทำงาน	ไม่ผ่าน
4	ค้นคืนเอกสารจากกล่องเอกสารที่ถูกค้นคืนหมดแล้ว	ใช้ข้อมูลทดสอบเดียวกันกับกรณีทดสอบการค้นคืนเอกสารข้อ 2 Continue? <i>Input "Yes"</i> Number of Files Retrieved <i>Input n n is any integer.</i>	โปรแกรมไม่อนุญาตให้ค้นคืนเอกสารจากกล่องเอกสารที่ไม่มีเอกสารเหลือแล้วและกลับไปหน้าจอเมนู	โปรแกรมหยุดทำงาน	ไม่ผ่าน
5	ค้นคืนเอกสารบางส่วนหรือทั้งหมดจากกล่องเอกสารที่ถูกค้าไม่ได้เป็นเจ้าของ	Folder NO. <i>Input n n != a</i> , a is folder no. the customer own. Number of Files Retrieved <i>Input n n is any integer.</i>	ไม่อนุญาตให้มีการค้นคืนกล่องเอกสารที่ถูกค้าไม่ได้เป็นเจ้าของ	โปรแกรมให้มีการค้นคืนเอกสารจากกล่องเอกสารที่ถูกค้าไม่ได้เป็นเจ้าของ	ไม่ผ่าน

3. บทสรุป

โครงการนี้เป็นการศึกษา ทำความเข้าใจ และจำลองขั้นตอนการทำงานของกิจการที่สนใจออกมาในรูปของโปรแกรม โดยมีจุดประสงค์เพื่อนำความรู้และทักษะที่ได้จากการศึกษาในรายวิชามาประยุกต์ใช้และแก้ไขปัญหาที่พบในธุรกิจ หัวข้อกรณีศึกษาที่นิสิตเลือกทำคือ กิจการการจัดเก็บเอกสารสำหรับโครงการนี้ นิสิตได้กำหนดขอบเขตการทำงานของโปรแกรมให้สามารถทำการใช้งานได้สองอย่าง คือ การจัดเก็บเอกสาร และการค้นคืนเอกสาร

จากการทดสอบโปรแกรม พบว่าโปรแกรมสามารถทำขั้นตอนของการจัดเก็บเอกสารได้อย่างเสร็จสิ้นลุล่วงและสมบูรณ์ แต่สำหรับกระบวนการของการค้นคืนเอกสาร จะพบว่ามีปัญหาเกิดขึ้นจำนวนมากทำให้กระบวนการนี้ไม่สามารถดำเนินการได้ลุล่วงได้ โดยรวมแล้ว นิสิตประเมินว่า โปรแกรมนี้สามารถทำงานตามขอบเขตหลักที่หวังไว้ได้ แต่ก็ยังมีข้อจำกัดในการใช้งานบางส่วนที่ควรปรับปรุงและพัฒนาเพื่อให้โปรแกรมมีประสิทธิภาพมากขึ้น

อย่างไรก็ตาม การทำโครงการนี้ทำให้นิสิตได้ฝึกใช้ ประยุกต์ความรู้และทักษะที่ได้จากการศึกษาในรายวิชามาทำความเข้าใจกับปัญหาทางธุรกิจ และนิสิตหวังว่าความรู้ที่ได้จะถูกนำไปใช้ประโยชน์ในคราวต่อไป

บรรณานุกรม

Access Records Management. (2019, November 28). Retrieved from

<https://www.accessrecordsmanagement.co.uk/>

Cube Records Management. (2019, November 28). Retrieved from <http://www.cuberms.com/>

JWD Group. (2019, November 28). Retrieved from <http://www.jwd-group.com/th/home>

Thai Assets Storage. (2019, November 28). Retrieved from <http://thaiassetsstorage.co.th/>

ภาคผนวก

Class Folder

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Collections;

namespace Don_t_Give_Up_You_Can_Do_it //:)
{
    class Folder
    {
        private int duration;
        private DateTime storedDate;
        private DateTime dueDate;

        private string type;
        private string zone;
        private int folderNO;

        private ArrayList files;

        private Customer customer;

        public Folder(int months, string t, string z, int fNO, Customer c)
        {
            duration = months;
            type = t;
            storedDate = DateTime.Today;
            dueDate = storedDate.AddMonths(months);
            zone = z; folderNO = fNO;
            files = new ArrayList();
            customer = c;
        }

        public void RecordFile(File f)
        {
            files.Add(f);
        }

        public void RetriveFile(int fno)
        {
            int index = 100000;
            for (int i = 0; i < files.Count; i++)
            {
                if (fno == ((File)files[i]).RefNO)
                {
                    File f = ((File)files[i]);
                    index = files.IndexOf(f);
                } break;
            }
            files.RemoveAt(index);
        }

        public void AccessFiles()
        {
            for (int i = 0; i < files.Count; i++)
            {
                Console.WriteLine("File NO.: " + ((File)files[i]).RefNO
                    + " Name: " + ((File)files[i]).Name);
            }
        }
    }
}
```

```

public string GetFileName(int no)
{
    for (int i = 0; i < files.Count; i++)
    {
        if (no == ((File)files[i]).RefNO)
        {
            return ((File)files[i]).Name;
        }
    }
    return "none";
}

public double CostFee(string t, int duration)
{
    double cost = 0;
    if (t == "General")
    {
        if (files.Count > 6) { cost = 500 * duration; }
        else if (files.Count >= 4) { cost = 250 * duration; }
        else { cost = 125 * duration; }
        return cost;
    }
    else if (t == "Legal")
    {
        if (files.Count > 6) { cost = 1000 * duration; }
        else if (files.Count >= 4) { cost = 500 * duration; }
        else { cost = 250 * duration; }
        return cost;
    }
    else { return cost; }
}

public int Duration { get { return duration; } }
public DateTime StoredDate { get { return storedDate; } }
public DateTime DuedDate { get { return dueDate; } }

public string Zone { get { return zone; } }
public int FolderNO { get { return folderNO; } }

public int CustID { get { return customer.CustID; } }
}
}

```

Class File

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Don_t_Give_Up_You_Can_Do_it
{
    class File
    {
        private int refNO;
        private int sheetQuan;
        private string name;

        public File(int NO, int quan, string n)
        {
            refNO = NO;
            sheetQuan = quan;
            name = n;
        }

        public int RefNO { get { return refNO; } }
        public int SheetQuan { get { return sheetQuan; } }
        public string Name { get { return name; } }
    }
}
```

Class Customer

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Collections;

namespace Don_t_Give_Up_You_Can_Do_it
{
    class Customer
    {
        private int custID;
        private string custName;
        private string phoneNum;
        private string address;

        private ArrayList folder = new ArrayList();

        public Customer(int id, string name, string phone, string ads)
        {
            custID = id; custName = name; phoneNum = phone; address = ads;
        }

        public int CustID { get { return custID; } }
        public string CustName { get { return custName; } }
        public string PhoneNum { get { return phoneNum; } }
        public string Address { get { return address; } }
    }
}
```

Main Program

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Collections;

namespace Don_t_Give_Up_You_Can_Do_it
{
    class Program
    {
        static void Main(string[] args)
        {
            SortedList folder = new SortedList();
            int folderNO = 1;

            SortedList customer = new SortedList();
            int custID = 1;

            int contractNO = 1;

            Console.WriteLine("Menu\n1. Collect Files\n2. Access and Retrive\n3. Quit");
            Console.Write("\nChoose -> "); int choice = int.Parse(Console.ReadLine());

            while (choice != 3)
            {
                if (choice == 1)
                {
                    double totalFee = 0;

                    Console.WriteLine("\n\t\t\tCustomer 's Details");
                    Console.Write("Name -> "); string name = Console.ReadLine();
                    Console.Write("Phone -> "); string phone = Console.ReadLine();
                    Console.Write("Address -> "); string address = Console.ReadLine();
                    customer.Add(custID, new Customer(custID, name, phone, address));

                    Console.Write("\nDuration [ in months e.g. 12, 24, etc. ] -> "); int duration =
int.Parse(Console.ReadLine());

                    Console.Write("\nQuantity of General File -> "); int gQuan =
int.Parse(Console.ReadLine());
                    Console.Write("Quantity of Legal File -> "); int lQuan =
int.Parse(Console.ReadLine());

                    if (gQuan > 0)
                    {
                        Console.WriteLine("\n\t\t\tGeneral Files");
                        Console.Write("Zone [ A - D ] -> "); string gZone = Console.ReadLine();
                        Console.WriteLine();
                        Folder GeneralFolder = new Folder(duration, "General", gZone, folderNO,
((Customer)customer[custID]));
                        folder.Add(folderNO, GeneralFolder);
                        for (int i = 1; i <= gQuan; i++)
                        {
                            Console.Write("Sheet Quantity -> "); int sheetQuan =
int.Parse(Console.ReadLine());
```

```

        Console.WriteLine("Name -> "); string n = Console.ReadLine();
        ((Folder)folder[folderNO]).RecordFile(new File(i, sheetQuan, n));
    }
    totalFee += ((Folder)folder[folderNO]).CostFee("General", duration);
    folderNO++;
}

if (lQuan > 0)
{
    Console.WriteLine("\n\t\t\tLegal Files");
    Console.WriteLine("Zone [ E - H ] -> "); string lZone = Console.ReadLine();
    Console.WriteLine();
    Folder LegalFolder = new Folder(duration, "Legal", lZone, folderNO,
    ((Customer)customer[custID]));
    folder.Add(folderNO, LegalFolder);
    for (int i = 1; i <= lQuan; i++)
    {
        Console.WriteLine("Sheet Quantity -> "); int sheetQuan =
        int.Parse(Console.ReadLine());
        Console.WriteLine("Name -> "); string n = Console.ReadLine();
        ((Folder)folder[folderNO]).RecordFile(new File(i, sheetQuan, n));
    }
    totalFee += ((Folder)folder[folderNO]).CostFee("Legal", duration);
    totalFee += 1500;
    folderNO++;
}

Console.WriteLine("\n\t\t\tService Fee");
Console.WriteLine("Total Fee Service: " + totalFee);

Console.WriteLine("\n\t\t\tContract");
Console.WriteLine("Contract NO.: " + contractNO);
Console.WriteLine("Customer Name: " + ((Customer)customer[custID]).CustName);
Console.WriteLine("Customer Phone NO.: " + ((Customer)customer[custID]).PhoneNum);
Console.WriteLine("Customer Address: " + ((Customer)customer[custID]).Address);
Console.WriteLine("\nDuration of Storage: " + ((Folder)folder[folderNO-
1]).Duration + " Months");
Console.WriteLine("Stored Date: " + ((Folder)folder[folderNO-1]).StoredDate);
Console.WriteLine("Dued Date: " + ((Folder)folder[folderNO-1]).DuedDate);
Console.WriteLine();

    custID++;
    contractNO++;
}

else if (choice == 2)
{
    Console.WriteLine("\nContract NO. -> "); int cno = int.Parse(Console.ReadLine());
    Console.WriteLine();
    for (int i = 1; i <= folder.Count; i++)
    {
        if (cno == ((Folder)(folder[i])).CustID)
        {
            Console.WriteLine("Folder NO.: " + ((Folder)(folder[i])).FolderNO);
        }
    }

    string key = "Yes";
    while (key == "Yes")
    {
        Console.WriteLine("\nFolder NO. -> "); int no = int.Parse(Console.ReadLine());

        if (((Folder)folder[no]).FolderNO == no)
        {
            Console.WriteLine(); ((Folder)folder[no]).AccessFiles();

            Console.WriteLine("\nNumber of Files Retrived -> "); int num =
            int.Parse(Console.ReadLine());
            Console.WriteLine("\nInput File NO. "); Console.WriteLine();

```

```

        for (int i = 1; i <= num; i++)
        {
            Console.Write("File NO. "); int fno = int.Parse(Console.ReadLine());
            Console.WriteLine("Name: " + ((Folder)folder[no]).GetFileName(fno));
            ((Folder)folder[no]).RetriveFile(fno);
        }

        Console.WriteLine("Send to " + "\n\t" +
            ((Customer)customer[cno]).Address);
    }

    Console.Write("Continue ? [ Yes | NO ] -> "); key = Console.ReadLine();
}
}
Console.WriteLine("\nMenu\n1. Collect Files\n2. Access and Retrive\n3. Quit");
Console.Write("\nChoose -> "); choice = int.Parse(Console.ReadLine());
}
}
}
}
}

```

