

รายการอ้างอิง

1. สมใจ บุญศิริ. การออกแบบระบบเครือข่ายคอมพิวเตอร์สำหรับงานรับสมัคร เพื่อสอบคัดเลือกบุคคลเข้าศึกษาในสถาบันอุดมศึกษา. วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ สาขาวิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2535.
2. ดนู สายสุทธิชัย. การพัฒนากราฟิกในตัวประสานผู้ใช้ สำหรับระบบไคลเอนต์ โดยใช้งานรับสมัครของขบวนการสอบคัดเลือกบุคคลเข้าศึกษาในสถาบันอุดมศึกษา. วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ สาขาวิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2537.
3. Coad, P., and Nicola, J. Object-oriented programming. New Jersey: Prentice-Hall. 1991.
4. Pressman, R., S. Software engineering: A practitioner's approach. 3rd ed. Singapore: McGraw Hill. 1992.
5. Satzinger, J., W. The object-oriented approach: Concepts, modeling, and system development. Massachusetts: boyd & fraser publishing company. 1996.
6. Tkach, D., and Puttick R. Object technology: In application development. USA: Benjamin/Cummings Publishing. 1994.
7. Ward, P., T. How to Integrate Object Orientation with Structured Analysis and Design. IEEE Software (March 1989): 74-82
8. Coad, P., North, D., and Mayfield, M. Object Models:Strategies. Patterns. & Applications. New Jersey: McGraw Hill. 1995.
9. Coad, P., and Yourdon, E. Object-Oriented Analysis. New Jersey: Prentice-Hall. 1991.
10. Fichman, R., G., and Kemerer, C., F. Object-Oriented and Convectional Analysis and Design Methodologies. COMPUTER (October 1992): 22-39
11. Bailin, S., C. An Object-Oriented Requirements Specification Method. Comm ACM Vol. 32 No. 5 (May 1989): 608-623
12. Coad, P., and Yourdon, E. Object-Oriented Design. New Jersey: Prentice-Hall. 1991.
13. Norman R., J. Object-Oriented System Analysis And Design. New Jersey: Prentice-Hall. 1996

ภาคผนวก

ภาคผนวก ก

พจนานุกรมข้อมูลของแผนภาพกระแสข้อมูล ส่วนไคลเอนต์

Data Flow Description

| | |
|------------------|---|
| Data Flow Name | TM13 |
| From-To Process | 5.รับข้อมูลผู้สมัครเพิ่มเติม-6.บันทึกข้อมูลผู้สมัคร 6.บันทึกข้อมูลผู้สมัคร-7.พิมพ์บัตรประจำตัวผู้สอบ |
| Data Composition | Exampplace, Subject Type, Seat No Sequence, Register Form No, Seat No, Name, Sex, School Code, Student Status, Grade, Client No, Seq No, Date, Session, Choices, Registration Fee, Subject Code, Status |

| | |
|------------------|--|
| Data Flow Name | TM3 |
| From-To Process | 7.พิมพ์บัตรประจำตัวผู้สอบ-ผู้สมัคร |
| Data Composition | Exampplace, Exampplace Name, Subject Type, Seat No, Name, Sex, Client No, Seq No, Date, Session, Choices, Subject Code |

| | |
|------------------|---|
| Data Flow Name | Error Message |
| From-To Process | 3.ตรวจสอบวิชาที่มีวันสอบตรงกัน-ผู้สมัคร |
| Data Composition | ข้อความแสดงรหัสวิชาที่มีวันที่สอบตรงกัน |

Data Store Description

| | |
|------------------|--|
| Data Store Name | Faculty-Subject Table |
| Purpose | เก็บข้อมูลรหัสคณะและวิชาที่สอบ |
| Data Composition | Faculty Code, Sex, Number of Subject, Subject Code |

| | |
|------------------|--|
| Data Store Name | Crash Table |
| Purpose | เก็บข้อมูลรหัสวิชาที่มีวันสอบวันเดียวกัน |
| Data Composition | Subject1, Subject2 |

| | |
|------------------|--|
| Data Store Name | TM13 Table |
| Purpose | เก็บข้อมูลผู้สมัคร |
| Data Composition | Examplice, Subject Type, Seat No Sequence, Register Form No, Seat No, Name, Sex, School Code, Student Status, Grade, Client No, Seq No, Date, Session, Choices, Registration Fee, Subject Code, Status |

| | |
|------------------|----------------------------|
| Data Store Name | Examplice Name Table |
| Purpose | เก็บข้อมูลชื่อของเขตการสอบ |
| Data Composition | Examplice, Examplice Name |

Process Description

| | |
|--------------|--------------------------------|
| Process Name | 1.คำนวณวิชาที่สอบ |
| Purpose | หาวิชาที่ต้องสอบจากคณะที่เลือก |
| Description | หาจากตาราง Faculty-Subject |

| | |
|--------------|---|
| Process Name | 2.คำนวณหมวดสอบ |
| Purpose | หาหมวดสอบของผู้สมัคร |
| Description | สอบวิชา 01-06 และ 12-44 เป็นหมวด 1 สอบวิชา 01-06 และ 12, 13, 18, 22 เป็นหมวด 2 สอบวิชา 07-11 และ 12-44 เป็นหมวด 3 สอบวิชา 01-06, 07-11, 12-44 เป็นหมวด 4 |

| | |
|--------------|-------------------------------------|
| Process Name | 3.ตรวจสอบวิชาที่มีวันสอบตรงกัน |
| Purpose | หาวิชาที่มีวันสอบตรงกัน |
| Description | ข้อมูลวิชาสอบตรงกันมาจากตาราง Crash |

| | |
|--------------|---|
| Process Name | 4.คำนวณเงินที่ใช้ในการสมัคร |
| Purpose | หาจำนวนเงินที่ผู้สมัครต้องจ่าย |
| Description | คำนวณจากคณะที่เลือกคณะละ 50 บาท และวิชาที่ต้องสอบ วิชาละ 50 บาท |

| | |
|--------------|--|
| Process Name | 5.รับข้อมูลผู้สมัครเพิ่มเติม |
| Purpose | รับข้อมูลที่ยังไม่ได้กรอกในช่วงแรกให้ครบถ้วน |
| Description | |

| | |
|--------------|--------------------------------|
| Process Name | 6.บันทึกข้อมูลผู้สมัคร |
| Purpose | นำผลการสมัครบันทึกลงแฟ้มข้อมูล |
| Description | |

| | |
|--------------|----------------------------|
| Process Name | 7. พิมพ์บัตรประจำตัวผู้สอบ |
| Purpose | พิมพ์ข้อมูลออกเครื่องพิมพ์ |
| Description | |

ภาคผนวก ข

พจนานุกรมข้อมูลของแผนภาพกระแสข้อมูล ส่วนเวิร์ฟเวอร์

Data Flow Description

| | |
|------------------|---|
| Data Flow Name | TM13 |
| From-To Process | เครื่องรับสมัคร-5.ตรวจสอบสถานะของการสมัคร 5.ตรวจสอบสถานะของการสมัคร-7.ปรับปรุงสถานะของการสมัคร 7.ปรับปรุงสถานะของการสมัคร-8.บันทึกข้อมูลผู้สมัคร |
| Data Composition | Exampplace, Subject Type, Seat No Sequence, Register Form No, Seat No, Name, Sex, School Code, Student Status, Grade, Client No, Seq No, Date, Session, Choices, Registration Fee, Subject Code, Status |

Data Store Description

| | |
|------------------|--|
| Data Store Name | Seat No Table |
| Purpose | เก็บข้อมูลเลขที่นั่งสอบที่ |
| Data Composition | Exampplace, Subject Type, Seat No Sequence, Register Form No, Seat No, Client No, Seq No, Status |

| | |
|------------------|---|
| Data Store Name | Index Table |
| Purpose | เก็บข้อมูลครรหณีของเลขที่นั่งสอบที่ว่างเป็นเลขแรก |
| Data Composition | Exampplace, Subject Type, Index |

| | |
|------------------|---|
| Data Store Name | Seat Count Table |
| Purpose | เก็บข้อมูลจำนวนผู้สมัครในแต่ละเขตการสอบ |
| Data Composition | Exampplace, Subject Type, Number of Assigned Seat |

| | |
|------------------|--|
| Data Store Name | Count Limit Table |
| Purpose | เก็บข้อมูลจำนวนจำกัดผู้สมัครในแต่ละเขตการสอบ |
| Data Composition | Examplice, Count Limit |

| | |
|------------------|--|
| Data Store Name | TM13 Table |
| Purpose | เก็บข้อมูลผู้สมัคร |
| Data Composition | Examplice, Subject Type, Seat No Sequence, Register Form No, Seat No, Name, Sex, School Code, Student Status, Grade, Client No, Seq No, Date, Session, Choices, Registration Fee, Subject Code, Status |

Process Description

| | |
|--------------|---|
| Process Name | 1.แก้ไขและปรับปรุงเพิ่มจำนวนผู้สมัครในแต่ละเขตการสอบ |
| Purpose | เลือกสนามสอบที่ยังมีที่ว่างให้แก่ผู้สมัคร |
| Description | สนามสอบจะว่างเมื่อตรวจสอบจากตาราง Seat Count และตาราง Count Limit ได้ผลดังนี้ สำหรับแต่ละเขตการสอบ หมวดสอบ 1+2+4 น้อยกว่า Count Limit หมวดสอบ 3+4 น้อยกว่า Count Limit |

| | |
|--------------|--|
| Process Name | 2. ค้นหาครรชนีของเลขที่นั่งสอบที่ว่าง |
| Purpose | ค้นหาครรชนีของแฟ้มข้อมูลเลขที่นั่งสอบเพื่อให้ได้ระเบียบของเลขที่นั่งสอบที่ว่าง |
| Description | |

| | |
|--------------|--|
| Process Name | 3. ค้นหาเลขที่นั่งสอบ |
| Purpose | เลือกเลขที่นั่งสอบที่ยังว่างอยู่ |
| Description | ใช้ค่าครรชนีเพื่อช่วยในการค้นหาให้เร็วขึ้น |

| | |
|--------------|--|
| Process Name | 4. ปรับปรุงแฟ้มครรชนี |
| Purpose | แก้ไขแฟ้มครรชนีเพื่อให้ชี้ไปยังระเบียบของแฟ้มเลขที่นั่งสอบที่ว่างถัดไป |
| Description | |

| | |
|--------------|--|
| Process Name | 5. ตรวจสอบสถานะของการสมัคร |
| Purpose | เพื่อให้แน่ใจว่าขั้นตอนการสมัครของผู้สมัครถูกต้องตามที่ควรเป็น ไม่มีการใช้เลขที่นั่งสอบซ้ำซ้อน |
| Description | |

| | |
|--------------|---|
| Process Name | 6. ทำรายการใหม่ |
| Purpose | ให้ผู้สมัครทำการสมัครใหม่ เมื่อเกิดความผิดพลาดในการรับสมัคร |
| Description | |

| | |
|--------------|----------------------------|
| Process Name | 7.ปรับปรุงสถานะของการสมัคร |
| Purpose | แก้ไขข้อมูลสถานะให้ถูกต้อง |
| Description | |

| | |
|--------------|--------------------------------|
| Process Name | 8.บันทึกข้อมูลผู้สมัคร |
| Purpose | นำผลการสมัครบันทึกลงแฟ้มข้อมูล |
| Description | |

ภาคผนวก ค

คำอธิบายคลาสของโปรแกรมไคลเอนต์

คลาส AppendLOGFile สืบทอดมาจาก

หน้าที่ บันทึกข้อมูลผิดพลาดในการทำงานลงแฟ้มข้อมูล

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|------------------------|
| private cDate | Data | วันเวลาที่บันทึกข้อมูล |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|-------------------------|--------|---------------------------------|
| public LOGClose | void | ปิดแฟ้มข้อมูลผิดพลาด |
| public LOGWrite(String) | void | เขียนข้อมูลผิดพลาด พร้อมวันเวลา |
| public write(String) | void | เขียนข้อมูลผิดพลาด พร้อมวันเวลา |
| private getTimeString | String | ให้ค่าวันเวลา |

คลาส Applicant

สืบทอดมาจาก TM13ClientObject

หน้าที่ ดำเนินขั้นตอนการสมัคร

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|---------------------------------|---------|---|
| public findSubjectTypeAndFee | void | คำนวณหาหมวดวิชาและค่าสมัคร |
| public requestSeatNo | boolean | ขอเลขที่นั่งสอบ ให้ค่า true เมื่อขอสำเร็จ |

| | | |
|---------------------------|---------|--|
| public confirmApplication | boolean | ยืนยันการสมัคร ให้ค่า true เมื่อยืนยันสำเร็จ |
| public reset | void | ล้างข้อมูลเดิมเพื่อรับสมัครใหม่ |
| public printIDCard | void | พิมพ์บัตรประจำตัวผู้สมัครสอบ |

คลาส CheckSum สืบทอดมาจาก

หน้าที่ คำนวณ checksum ของข้อความ

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|----------------------------|---------|--|
| public getCheckSum(String) | byte | คืนค่า checksum ของข้อความ |
| public isEqual(byte) | boolean | ตรวจสอบค่า checksum คืนค่า true เมื่อเท่ากัน |
| public calSum(byte[],int) | byte | คำนวณค่า checksum |

คลาส CheckSum2 สืบทอดมาจาก CheckSum, IEncode

หน้าที่ คำนวณ checksum ของข้อความ

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|-----------------------|--------|---|
| public encode(String) | String | คำนวณและรวมค่า checksum เข้าไปใน String |

| | | |
|-------------------------|---------|---|
| public decode(String) | String | ตรวจสอบและนำค่า checksum ออกจาก String |
| public validate(String) | boolean | ตรวจสอบความถูกต้องของ String ที่มี checksum |

คลาส Client4

สืบทอดมาจาก

หน้าที่ เริ่มการทำงานของโปรแกรมไคลเอนต์

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|-----------------------|--------|--------------------------------|
| public main(String[]) | void | สร้างวัตถุแล้วแสดงหน้าจอ login |

คลาส CommProxy

สืบทอดมาจาก

หน้าที่ รับส่งข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|------------------|-----------|--------------------------|
| private _encoder | IEncode | ใช้เข้ารหัสข้อความที่ส่ง |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|-------------------------------|--------|---|
| public setEncodeClass(String) | void | ตั้งค่าของคลาสที่สืบทอดจาก IEncode เพื่อที่จะเป็นวัตถุสำหรับการเข้ารหัส |
| public close | void | ปิดช่องทางการติดต่อ |

| | | |
|---------------------|--------|------------|
| public send(String) | void | ส่งข้อความ |
| public receive | String | รับข้อความ |

คลาส CrashObject สืบทอดมาจาก

หน้าที่ ควบคุมเพิ่มข้อมูลวิชาที่มีวันสอบตรงกัน

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|------------------------|
| errorMsg | String | ข้อความแสดงความผิดพลาด |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|---------------------------|---------|--|
| public chk2Subj(Vector) | boolean | ตรวจสอบว่ามีวิชาคู่ใดใน Vector สอบตรงกัน คืนค่า true ถ้าไม่มีวิชาตรงกัน |
| public errMsg | String | คืนค่าข้อความแสดงความผิดพลาด |
| private CrashTest(Vector) | boolean | ตรวจสอบว่ามีวิชาคู่ใดใน Vector สอบตรงกัน คืนค่า true ถ้าไม่มีวิชาตรงกัน |

คลาส EnrollProperties สืบทอดมาจาก Properties

หน้าที่ อ่านค่าติดตั้งจากแฟ้ม Enroll.conf

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|------------------------|
| errorMsg | String | ข้อความแสดงความผิดพลาด |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|------------|--------|----------|
|------------|--------|----------|

| | | |
|---------------------------|---------|--|
| public chk2Subj(Vector) | boolean | ตรวจสอบว่ามีวิชาคู่ใดใน Vector สอดตรงกัน คืนค่า true ถ้าไม่มีวิชาตรงกัน |
| public errMsg | String | คืนค่าข้อความแสดงความผิดพลาด |
| private CrashTest(Vector) | boolean | ตรวจสอบว่ามีวิชาคู่ใดใน Vector สอดตรงกัน คืนค่า true ถ้าไม่มีวิชาตรงกัน |

คลาส ExamPlaceObject สืบทอดมาจาก

หน้าที่ ควบคุมเพิ่มข้อมูลชื่อของเขตการสอบ

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|-----------------------------|
| PlaceList | Hashtable | รายการชื่อของแต่ละเขตการสอบ |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|----------------------|--------|---------------------------------------|
| private getName(int) | String | คืนค่าชื่อของเขตการสอบหมายเลขที่กำหนด |

คลาส FacultyChoiceObject สืบทอดมาจาก

หน้าที่ ควบคุมเพิ่มข้อมูลรหัสคณะและวิชาที่สอบ

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|------------------|
| FacCode | String | รหัสคณะ |
| Sex | char | เพศที่สมัครได้ |
| NumSubject | char | จำนวนวิชาที่สอบ |
| Subject | Vector | รายการวิชาที่สอบ |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|---------------------------|--------|------------------------|
| public setFacCode(String) | void | ตั้งค่ารหัสคณะ |
| public getFacCode | String | คืนค่ารหัสคณะ |
| public getSex | char | คืนค่าเพศที่สมัครได้ |
| public getNumSubject | char | คืนค่าจำนวนวิชาที่สอบ |
| public getSubject | Vector | คืนค่ารายการวิชาที่สอบ |

คลาส FrmMain

สืบทอดมาจาก Frame

หน้าที่ หน้าจอรับสมัครหลัก

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|--------------|--------------------------------|
| LoginDlg | dlgLogin | หน้าจอเริ่มโปรแกรม |
| ErrorDlg | dlgInfo | หน้าจอแสดงข้อความผิดพลาด |
| FiSeqCount | SeqCntObject | วัตถุสำหรับนับเลขลำดับการสมัคร |
| myAppl | Applicant | วัตถุผู้สมัคร |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|---------------------------|---------|-----------------------------|
| private processFirstHalf | boolean | ขั้นตอนการขอเลขที่นั่งสอบ |
| private processSecondHalf | boolean | ขั้นตอนการยืนยันการรับสมัคร |
| private newApplicant | void | เริ่มการสมัครคนใหม่ |
| private increaseSeqCount | void | เพิ่มจำนวนนับเลขลำดับ |
| private decreaseSeqCount | void | ลดจำนวนนับเลขลำดับ |

คลาส IEncode

สืบทอดมาจาก

หน้าที่ เป็นคลาสนามธรรม

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|-------------------------|---------|----------------------------------|
| public encode(String) | String | เข้ารหัส |
| public decode(String) | String | ถอดรหัส |
| public validate(String) | boolean | ตรวจสอบความถูกต้องของข้อความรหัส |

คลาส PrintTM13

สืบทอดมาจาก

หน้าที่ พิมพ์ข้อมูล ทม. 13

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------------|
| TM13Record | String | ข้อมูลผู้สมัคร |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|-----------------------|--------|----------------------|
| public PrintToScreen | void | พิมพ์ออกหน้าจอ |
| public PrintToPrinter | void | พิมพ์ออกเครื่องพิมพ์ |

คลาส PrinterStream

สืบทอดมาจาก PrintStream

หน้าที่ พิมพ์ข้อมูล ทม. 13

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|-----------------|--------|--------------------|
| public linefeed | void | ขึ้นบรรทัดใหม่ |
| public formfeed | void | ขึ้นหน้ากระดาษใหม่ |

คลาส SeqCntObject

สืบทอดมาจาก

หน้าที่ ควบคุมแฟ้มข้อมูลลำดับการสมัคร

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------------|
| cDate | Date | วันที่รับสมัคร |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|--------------|--------|----------------------------|
| public add | void | เพิ่มจำนวนเลขลำดับการสมัคร |
| public sub | void | ลดจำนวนเลขลำดับการสมัคร |
| public close | void | ปิดแฟ้มข้อมูล |

คลาส TM13ClientObject สืบทอดมาจาก

หน้าที่ ควบคุมเพิ่มข้อมูลผู้สมัคร

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|------------------|-----------|-----------------------|
| S_ExamPlace | String | สนามสอบที่ได้ |
| C_SubjectType | String | หมวดสอบ |
| S_SequenceNo | String | เลขลำดับในเซิร์ฟเวอร์ |
| C_RegFormNo | String | เลขที่ใบสมัคร |
| S_SeatNo | String | เลขที่นั่งสอบ |
| C_Name | String | ชื่อ-นามสกุลผู้สมัคร |
| C_Sex | String | เพศ |
| C_SchoolCode | String | รหัสโรงเรียน |
| C_StudentStatus | String | สถานะทางการศึกษา |
| C_Grade | String | คะแนนเฉลี่ย |
| C_ClientNo | String | เลขที่เครื่องรับสมัคร |
| C_SeqNo | String | เลขลำดับการสมัคร |
| _Date | String | วันที่รับสมัคร |
| _Session | String | ช่วงของการรับสมัคร |
| C_Choice | String[4] | คณะที่เลือก |
| _RegFee | String | ค่าสมัคร |
| _SubjectCode | Vector | วิชาที่ต้องสอบ |
| _Status | String | สถานะการสมัคร |
| _ExamPlaceChoice | String[3] | เขตการสอบที่เลือก |

| | | |
|--------------|--------|----------------|
| ErrorMessage | String | ข้อความผิดพลาด |
|--------------|--------|----------------|

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|---------------------------|--------|---|
| public getTM13String | String | คืนค่าข้อมูลในรูปของ String เดียว |
| public checkSubjectAndFee | void | คำนวณวิชาและค่าสมัคร |
| public checkSubjectType | void | คำนวณหมวดสอบ |
| public doFirstHalf | void | ทำการคำนวณวิชา ค่าสมัครและหมวดสอบ |
| public writeToFi | void | เขียนข้อมูลลงแฟ้มข้อมูลเมื่อการสมัครเสร็จสิ้น |

คลาส TM3Printer

สืบทอดมาจาก PrintTM13

หน้าที่ พิมพ์ข้อมูล ทม. 13

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|--------------------|--------|---------------------------|
| public printIDCard | void | พิมพ์บัตรประจำตัวผู้สมัคร |

คลาส Util

สืบทอดมาจาก

หน้าที อรรถประโยชน์

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|--|---------|------------------------------------|
| public checkBoundDigit(String, int, Component) | boolean | ตรวจสอบความกว้างข้อมูลใน Component |
| public correctBound(String, int) | String | ตรวจสอบความกว้างข้อมูล |
| public sortV(Vector) | Vector | เรียงลำดับข้อมูลใน Vector |

คลาส dlgInfo

สืบทอดมาจาก Dialog

หน้าที หน้าต่างข้อผิดพลาด

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|------------|--------|----------|
| | | |

คลาส dlgLogin

สืบทอดมาจาก Dialog

หน้าที่ หน้าต่างเริ่มเข้าโปรแกรม

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|--------------------|-----------|-----------------------|
| protected ClientNo | int | เลขที่เครื่องรับสมัคร |
| protected Day | int | วันที่รับสมัคร |
| protected Month | int | เดือนที่รับสมัคร |
| protected Year | int | ปีที่รับสมัคร |
| protected Session | int | ช่วงเวลาที่รับสมัคร |

ตารางบริการ

| ชื่อบริการ | สินค้า | คำอธิบาย |
|------------|--------|----------|
| | | |

ภาคผนวก

คำอธิบายคลาสของโปรแกรมเวิร์ฟเวอร์

คลาส CheckSum **สืบทอดมาจาก**

หน้าที่ คำนวณ checksum ของข้อความ

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|----------------------------|---------|--|
| public getCheckSum(String) | byte | คืนค่า checksum ของข้อความ |
| public isEqual(byte) | boolean | ตรวจสอบค่า checksum คืนค่า true เมื่อเท่ากัน |
| public calSum(byte[],int) | byte | คำนวณค่า checksum |

คลาส CheckSum2 **สืบทอดมาจาก** CheckSum, IEncode

หน้าที่ คำนวณ checksum ของข้อความ

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|-------------------------|---------|---|
| public encode(String) | String | คำนวณและรวมค่า checksum เข้าไปใน String |
| public decode(String) | String | ตรวจสอบและนำค่า checksum ออกจาก String |
| public validate(String) | boolean | ตรวจสอบความถูกต้องของ String ที่มี checksum |

คลาส CommProxy สืบทอดมาจาก

หน้าที่ รับส่งข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|------------------|-----------|--------------------------|
| private _encoder | IEncode | ใช้เข้ารหัสข้อความที่ส่ง |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|----------------------------------|--------|---|
| public setEncodeClass(String) | void | ตั้งค่าของคลาสที่สืบทอดจาก IEncode เพื่อที่จะ เป็นวัตถุสำหรับการเข้ารหัส |
| public close | void | ปิดช่องทางการติดต่อ |
| public send(String) | void | ส่งข้อความ |
| public receive | String | รับข้อความ |

คลาส Confirmation สืบทอดมาจาก

หน้าที่ ให้บริการการยืนยันการรับสมัคร

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|--------------------|------------------|-------------------------------------|
| private NumConfirm | int | จำนวนการยืนยันการรับสมัคร |
| private seatobj | SeatNoObject | วัตถุสำหรับเพิ่มข้อมูลเลขที่นั่งสอบ |
| private tm13 | TM13ServerObject | วัตถุสำหรับเพิ่มข้อมูลผู้สมัคร |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|---------------------------|--------|---------------------------------|
| public confirm(String) | void | ยืนยันการรับสมัคร |
| public getResult | String | คืนค่าผลลัพธ์จากการทำงาน |
| public getNumConfirmation | int | คืนค่าจำนวนการยืนยันการรับสมัคร |

คลาส ConnectionManager สืบทอดมาจาก

หน้าที่ ให้บริการการยืนยันการรับสมัคร

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|--------------------|---------------------------------------|
| my_thread | ServerConnection[] | อะเรย์ของวัตถุสำหรับให้บริการไคลเอนต์ |
| _Addon | IAddonServer | วัตถุสำหรับทำงานก่อนรอการติดต่อ |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|--------------------------------|--------|---------------------------------------|
| getNumConnection | int | คืนค่าจำนวนไคลเอนต์ที่ให้บริการอยู่ |
| getConnectionList | Vector | คืนค่ารายการของวัตถุ ServerConnection |
| notifyRemove(ServerConnection) | void | ลบวัตถุ ServerConnection ออกจากอะเรย์ |

คลาส CountLimitObject สืบทอดมาจาก

หน้าที่ วัตถุของเพิ่มข้อมูลจำนวนจำกัดผู้สมัครในแต่ละสนามสอบ

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|------------------|--------|---|
| getLimit(String) | int | คืนค่าจำนวนที่นั่งที่รับได้ของเขตการสอบที่กำหนด |

คลาส EnrollProperties สืบทอดมาจาก Properties

หน้าที่ อ่านค่าติดตั้งจากแฟ้ม Enroll.conf

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|------------------------|
| errorMsg | String | ข้อความแสดงความผิดพลาด |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|---------------------------|---------|--|
| public chk2Subj(Vector) | boolean | ตรวจสอบว่ามีวิชาคู่ใดใน Vector สอบตรงกัน คืนค่า true ถ้าไม่มีวิชาตรงกัน |
| public errMsg | String | คืนค่าข้อความแสดงความผิดพลาด |
| private CrashTest(Vector) | boolean | ตรวจสอบว่ามีวิชาคู่ใดใน Vector สอบตรงกัน คืนค่า true ถ้าไม่มีวิชาตรงกัน |

คลาส FPAddOn สืบทอดมาจาก IAddOnServer

หน้าที่ อ่านค่า file pointer ของระเบียบเลขที่นั่งสอบที่ว่างเลขแรกก่อนการสมัครแต่ละวัน

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|-----------------------|-----------------|-------------------------------------|
| array_of_file_pointer | long[] | อะเรย์ของ file pointer |
| fp_finder | FindFirstUnused | วัตถุที่ใช้ในการหาเลขที่นั่งสอบว่าง |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|---------------------------------|--------|--|
| public initialize | void | ทำการอ่านค่า file pointer |
| public getFilePointerList | long[] | คืนค่า อะเรย์ของ file pointer ของทุกกรณี |
| public getFilePointerIndex(int) | long | คืนค่า file pointer ของกรณีที่ระบุ |

คลาส FindFirstUnused สืบทอดมาจาก

หน้าที่ อ่านค่า file pointer ของระเบียบเลขที่นั่งสอบที่ว่างเลขแรกก่อนการสมัครแต่ละวัน

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|--------------------------|--------|---------------------------|
| public findFirstUnused() | long[] | ทำการอ่านค่า file pointer |

คลาส IAddOnServer สืบทอดมาจาก

หน้าที่ อ่านค่า file pointer ของระเบียบเลขที่นั่งสอบที่ว่างเลขแรกก่อนการสมัครแต่ละวัน

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|-------------------|--------|---|
| public initialize | void | ถูกเรียกใช้จากวัตถุ ConnectionManager ก่อน รอกการติดต่อจากไคลเอนต์ |

คลาส IEncode

สืบทอดมาจาก

หน้าที่ เป็นคลาสนามธรรม

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|-------------------------|---------|----------------------------------|
| public encode(String) | String | เข้ารหัส |
| public decode(String) | String | ถอดรหัส |
| public validate(String) | boolean | ตรวจสอบความถูกต้องของข้อความรหัส |

คลาส ITaskSel

สืบทอดมาจาก

หน้าที่ เป็นคลาสนามธรรม ถูกใช้โดยวัตถุ ServerConnection

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|--|--------|--|
| public setInput(String) | void | ตั้งค่าข้อมูลเข้าที่มาจากไคลเอนต์ |
| public setAddOnObject (IAddOnServer) | void | ตั้งค่าสำหรับการใช้ประโยชน์ภายหลัง |
| public getOutputString | String | คืนค่าผลลัพธ์ของการทำงาน |
| public getTaskStatus | String | คืนค่าสถานะการทำงาน |
| public setConnectionManager (ConnectionManager) | void | ตั้งค่าวัตถุ ConnectionManager สำหรับการใช้งานที่ต้องการทราบ |

คลาส IndexObject

สืบทอดมาจาก

หน้าที่ เป็นวัตถุของแฟ้มข้อมูลครรรนี้

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------------------------------|
| ExamPlace | String | เขตการสอบ |
| SubjectType | String | หมวดสอบ |
| Index | String | ค่า SequenceNo ของระเบียบที่ว่าง |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|------------|--------|----------|
| | | |

คลาส Reservation

สืบทอดมาจาก

หน้าที่ ให้บริการการขอเลขที่นั่งสอบ

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|--------------------|-----------|--------------------------------------|
| private NumReserve | int | จำนวนการขอเลขที่นั่งสอบ |
| private job | doRequest | วัตถุสำหรับขั้นตอนการขอเลขที่นั่งสอบ |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|--------------------------|--------|-------------------------------|
| public reserve(String) | void | ขอเลขที่นั่งสอบ |
| public getResult | String | คืนค่าผลลัพธ์จากการทำงาน |
| public getNumReservation | int | คืนค่าจำนวนการขอเลขที่นั่งสอบ |

คลาส SeatCountObject สืบทอดมาจาก

หน้าที่ วัตถุของแฟ้มข้อมูลจำนวนผู้สมัครในแต่ละเขตการสอบ

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------------|-----------|-------------------------------|
| private ExamPlace | int | เขตการสอบ |
| private SubjectType | int | หมวดสอบ |
| private NumAssSeat | int | จำนวนที่นั่งที่รับสมัครไปแล้ว |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|-----------------------------------|--------|--|
| public getNumOfAssignSeat124() | int | คืนค่าจำนวนที่นั่งของหมวดสอบ 1,2 และ 4 |
| public getNumOfAssignSeat34 | int | คืนค่าจำนวนที่นั่งของหมวดสอบ 3 และ 4 |
| private numOfAssignSeat(int) | int | คืนค่าจำนวนที่นั่งของหมวดสอบที่กำหนด |

คลาส SeatNoObject สืบทอดมาจาก

หน้าที่ วัตถุของแฟ้มข้อมูลเลขที่นั่งสอบ

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------------|-----------|------------------------------|
| private ExamPlace | String | เขตการสอบ |
| private SubjectType | String | หมวดสอบ |
| private SequenceNo | String | เลขลำดับของแฟ้มเลขที่นั่งสอบ |
| private RegFormNo | String | เลขที่ใบสมัคร |
| private ClientNo | String | เลขที่เครื่องรับสมัคร |
| private SeqNo | String | เลขลำดับการรับสมัคร |
| private SeatNo | String | เลขที่นั่งสอบ |
| private Status | String | สถานะการรับสมัคร |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|------------|--------|----------|
| | | |

คลาส ServerConnection สืบทอดมาจาก

หน้าที่ ให้บริการตามที่ไคลเอนต์

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|------------------|-----------|-------------------------|
| private myClient | CommProxy | วัตถุที่ใช้ในการสื่อสาร |
| private myTask | ITaskSel | วัตถุที่ให้บริการ |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|----------------------|---------|--|
| private doServerWork | boolean | ให้บริการตามที่ร้องขอ คืนค่า true ถ้าทำงานสำเร็จ |
| public getTaskStatus | String | คืนค่าสถานะการทำงาน |

คลาส StartServer สืบทอดมาจาก

หน้าที่ เริ่มการทำงานของโปรแกรมเซิร์ฟเวอร์

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|-----------------------|--------|--------------------------------------|
| public main(String[]) | void | สร้างวัตถุแล้วรอการติดต่อจากไคลเอนต์ |

คลาส StatManager

สืบทอดมาจาก

หน้าที่ วัตถุประสงค์ให้บริการด้านสถิติ

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|--|--------|---|
| public lookfor(String) | void | หาค่าจำนวนไคลเอนต์ที่ต่ออยู่ และชื่อของไคลเอนต์ |
| public lookfor(String,String) | void | หาค่าจำนวนการขอเลขที่นั่งสอบหรือจำนวนการยืนยันการสมัครของไคลเอนต์ที่กำหนด |
| public getResult | String | คืนค่าผลลัพธ์ |
| public setConnectionManager(ConnectionManager) | void | ตั้งค่าวัตถุ ConnectionManager สำหรับการดึงค่าสถิติบางตัวจากวัตถุนี้ |
| private getNCC | void | หาค่าจะนวนไคลเอนต์ที่ติดต่ออยู่ |
| private getNGS(String) | void | หาค่าจำนวนการขอเลขที่นั่งสอบ |
| private getNWT(String) | void | หาค่าจำนวนการยืนยันการรับสมัคร |
| private getNAL | void | หาชื่อแทนไคลเอนต์ที่ติดต่ออยู่ทั้งหมด |

คลาส TM13ServerObject

สืบทอดมาจาก

หน้าที่ วัตถุให้บริการด้านสถิติ

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|---------------------------|
| serverstring | String | ข้อมูล ทม. 13 จากไคลเอนต์ |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|--------------------|--------|---------------------------------|
| public writeToFile | void | เขียนข้อมูลลงแฟ้มข้อมูลผู้สมัคร |

คลาส TaskSelector

สืบทอดมาจาก ITaskSel

หน้าที่ วัตถุที่ให้บริการโดยจะทำหน้าที่แบ่งการทำงานให้กับวัตถุ Reservation และวัตถุ Confirmation

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|------------|--------|----------|
| | | |

คลาส TaskStatusSplitter

สืบทอดมาจาก ITaskSel

หน้าที่ แปลงค่าสถานะของวัตถุ ServerConnection ให้ง่ายขึ้น

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|---------------------------|--------|---------------------------------|
| public getName | String | คืนค่าชื่อของวัตถุ |
| public getNumReservation | int | คืนค่าจำนวนการจองเลขที่นั่งสอบ |
| public getNumConfirmation | int | คืนค่าจำนวนการยืนยันการรับสมัคร |

คลาส Util

สืบทอดมาจาก

หน้าที่ อรรถประโยชน์

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|---------------|-----------|----------|
| | | |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|--|---------|------------------------------------|
| public checkBoundDigit(String, int, Component) | boolean | ตรวจสอบความกว้างข้อมูลใน Component |
| public correctBound(String, int) | String | ตรวจสอบความกว้างข้อมูล |
| public sortV(Vector) | Vector | เรียงลำดับข้อมูลใน Vector |

คลาส doRequest

สืบทอดมาจาก

หน้าที ทำการขอเลขที่นั่งสอบ

ตารางคุณลักษณะ

| ชื่อคุณลักษณะ | แบบข้อมูล | คำอธิบาย |
|-----------------|------------------|--|
| public IndexO | IndexObject | วัตถุสำหรับแฟ้มดรรชนี |
| public seatCntO | SeatCountObject | วัตถุสำหรับข้อมูลจำนวนผู้สมัครในแต่ละเขต การสอบ |
| public CntLimO | CountLimitObject | วัตถุสำหรับข้อมูลจำนวนจำกัดในแต่ละเขต การสอบ |
| public SeatNoO | SeatNoObject | วัตถุสำหรับแฟ้มข้อมูลเลขที่นั่งสอบ |

ตารางบริการ

| ชื่อบริการ | คืนค่า | คำอธิบาย |
|-----------------------------|--------|------------------------------|
| public setREQString(String) | void | ตั้งค่าข้อมูลขอใช้บริการ |
| public getACKString | String | คืนค่าผลลัพธ์จากการดำเนินการ |
| public findExamplace | void | ค้นหาเขตการสอบที่ว่าง |
| public findSeatNo | void | ค้นหาเลขที่นั่งสอบที่ว่าง |

ภาคผนวก ๑

ตัวอย่างคลาสในโปรแกรมไคลเอนต์

คลาส Applicant

```
// Generated by Together
import java.lang.String;
import java.net.Socket;
import java.io.IOException;
import java.net.UnknownHostException;
public class Applicant extends TM13ClientObject {
    Applicant(String clino, String _date, String sess) {
        super(clino,_date,sess);    // send to TM13ClientObject
        // setup communication
        try {
            myServer = new CommProxy(getSocketToServer());
        } catch (IOException e) {
            System.err.println("Applicant:Can't create CommProxy object.");
            e.printStackTrace();
            System.exit(1);
        }
    }

    public void setCommEncode(String enc) {
        myServer.setEncodeClass(enc);
    }

    public void findSubjectTypeAndFee(){
        super.doFirstHalf();    // find subjecttype, fee, list of subject
    }

    public boolean requestSeatNo() {
        String ss = super.getRequestString();
        boolean ret=true;
        String sendstring,serverstring,header;
        int limit=0;
```

```

String ermess = new String();

do {
    sendstring = (new String("REQ")).concat(ss);
    serverstring = new String("");
    try {
        limit++;
        System.out.println("sending REQ:"+limit+"."+ss);
        myServer.send(sendstring); // Send request to Server
        // wait to reply
        sendstring = myServer.receive();
        if (sendstring==null) {
            limit=12; // checksum failed
            //ermess = "Error check incoming packet";
            ermess = "???????? Server ??????,????????????????";
        }
        serverstring = sendstring.substring(3,sendstring.length());
        System.out.println("receiving ACK:"+serverstring.length()+"."+serverstring);
        header = sendstring.substring(0,3);
    } catch(Exception exception) {
        System.err.println("Exception #2:" + exception);
        ermess = exception.toString();
        header = new String("REP");
    }
} while (header.equals("REP")&&(limit<10));

if (limit>=10) { // some error occur
    this.ErrTitle = new String("Error on Connection: Request for Seat no.");
    if (limit==12) { // checksum failed
        this.ErrMessage = new String("CheckSum failed:"+ermess+"\n");
    } else {
        this.ErrMessage = new String("Exception error:"+ermess+"\n");
    }
    ret = false;
} else if (header.equals("IGN")) {
    this.ErrTitle = new String("Error on Connection: Request for Seat no.");

```

```

    this.ErrorMessage = new String("Error update seat:"+Server ??????????,?????????????????");
    ret = false;
    // save error data to error file
    try {
        AppendLOGFile fp = new AppendLOGFile("ErrorC"+super.getClientNo()+".dat", false, false);
        fp.LOGwrite("IGN occur:when request seat");
        fp.LOGclose();
    } catch (IOException e) {
        System.err.println("Error file error2 :"+e);
    }
} else {    // unpack packet
    super.setExamPlaceGet(serverstring.substring(0,1));
    super.setSequenceNoGet(serverstring.substring(1,6));
    super.setSeatNoGet(serverstring.substring(6,13));
}

return ret;
}

```

```

public boolean confirmApplication() {
    boolean ret = true;
    String ss = super.getTM13String();
    String sendstring,serverstring,header;
    int limit=0;
    String ermess = new String();

    do {
        sendstring = (new String("REC")).concat(ss);
        serverstring = new String("");
        try {
            limit++;
            System.out.println("sending REC:"+limit+":"+ss);
            myServer.send(sendstring);
            // wait to reply
            sendstring = myServer.receive();
            if (sendstring==null) {

```

```

        limit=12; // checksum failed

        //errmsg = "Error check incoming packet";
        errmsg = "???????? Server ??????,????????????????";
    }

    serverstring = sendstring.substring(3,sendstring.length());
    System.out.println("receiving ACK:"+serverstring.length()+":"+serverstring);
    header = sendstring.substring(0,3);
} catch(Exception exception) {
    System.err.println("Exception #3:" + exception);
    errmsg = exception.toString();
    header = new String("REP");
}
} while (header.equals("REP")&&(limit<10));

if (limit>=10) { // some error occur
    this.ErrTitle = new String("Error on Connection: Record Tm13");
    if (limit==12) { // checksum failed
        this.ErrMessage = new String("CheckSum failed 2:"+errmsg);
    } else {
        this.ErrMessage = new String("Exception error 2:"+errmsg);
    }
    ret = false;
} else if (header.equals("IGN")) {
    this.ErrTitle = new String("Error on Connection: Record Tm13");
    this.ErrMessage = new String("Error update seat:"+Server ??????????,????????????????");
    ret = false;
    // save error data to error file
    try {
        AppendLOGFile fp = new AppendLOGFile("ErrorC"+super.getClientNo()+".dat", false, false);
        fp.LOGwrite("IGN occur:when write TM13 record below");
        fp.LOGclose();
    } catch (IOException e) {
        System.err.println("Error file error2 :"+e);
    }
}
}
return ret;

```



```
}

public String getErrTitle() {
    return ErrTitle;
}

public String getErrMsg() {
    return ErrMessage;
}

public void reset() { // reset applicant
/**/ System.out.println("Applicant.reset() called.");
    super.clearAttribute();
}

public void printIDCard() { // print TM3
    System.out.println("Applicant.printIDCard");
    myPrinter = new TM3Printer(super.getTM13String());
    myPrinter.PrintToScreen();
    myPrinter.printIDCard();
}

protected void finalize() {
    try {
        myServer.close();
    } catch (IOException e) {
        System.err.println("Applicatn:Can't close CommProxy object.");
        e.printStackTrace();
        System.err.flush();
    }
}

private static Socket getSocketToServer() {
    EnrollProperties a = new EnrollProperties();
    String sport = new String(a.getProperty("ServerPort"));
}
```

```

String sname = new String(a.getProperty("ServerName"));
Socket mysock=null;
try {
    mysock = new Socket(sname,Integer.parseInt(sport));
} catch (UnknownHostException e2) {
    System.err.println("Applicant:Can't create CommProxy object.");
    e2.printStackTrace();
    System.exit(1);
} catch (IOException e) {
    System.err.println("Applicant:Can't create CommProxy object.");
    e.printStackTrace();
    System.exit(1);
}
return mysock;
}

private CommProxy myServer = null;
private TM3Printer myPrinter;
private String ErrTitle = new String("");
private String ErrMessage = new String("");

}

```

คลาส FacultyChoiceObject

```

/**
 * FacultyChoiceObject.java
 * create FacultyChoice object from Faculty Choice & Subject File
 * @version 0.01 11/21/96
 * @author Jakrit Siriwathanakul
 */
import java.lang.*;
import java.io.*;
import java.util.*;
/**
 * This class will read FacSub.dat and find data for selected faculty
 * which come with constructor, this time use sequential search in file

```

```

* (more intelligent 'll be add later)
*/
public class FacultyChoiceObject {
    private String FacCode = new String("0000"); // don't have faculty
    private char Sex = ' '; // not strick in sex
    private char NumSubject = '0'; // default no subject
    private Vector Subject = new Vector(); // array to save each subject
    private RandomAccessFile fp;

    /**
     * Create data object which is selected faculty
     * @param facSelected create this Faculty object
     */
    public FacultyChoiceObject(String facSelected) {
        setFacCode(facSelected);
    }

    /**
     * Re input a new Faculty Code
     * @param faccode a new faculty code of object
     */
    public void setFacCode(String faccode) {
        try {
            // fp = new RandomAccessFile(ConstValue.FACSUBRECS.name(),"r");
            fp = new RandomAccessFile(this.getFacSubFileFromConfig(),"r");
            fp.seek(0); // rewind
            // jump
            long jump = (fp.length()*Integer.parseInt(faccode)/this.getMaxFacultyFromConfig())-
(2*this.getRecordLenFromConfig());
            if (jump < 0) jump=0;
            fp.seek( jump );
            if (jump!=0) fp.readLine();
            String line;
            while ((line = fp.readLine()) != null) {
                // System.out.println("1">"+line);
                String cutFac = line.substring(0,4);

```

```

        // System.out.println("2>"+cutFac);
        if (faccode.equals(cutFac)) {
            parseData(line);
            break;
        }
    }
    fp.close();
} catch (IOException e) {
    System.err.println(e);
}
}

```

```
/**
```

```
 * Parse string into data element
```

```
 * @param l data string
```

```
 */
```

```

private void parseData(String l) {
    // System.out.println("3>"+l);
    String s = new String();
    FacCode = l.substring(0,4);
    Sex = l.charAt(4);
    NumSubject = l.charAt(5);
    int n = Character.digit(NumSubject, 10);
    n = (n==0)? 10:n;
    for (int i=0;i<n;i++) {
        s = l.substring(6+(2*i),8+(2*i));
        // System.out.println("4>"+s);
        Subject.addElement(s);
    }
}
}

```

```
/**
```

```
 * Get Faculty Code which input
```

```
 */
```

```

public String getFacCode() {

```

```

    return FacCode;
}

/**
 * Get a specific code for check sex for faculty (1=male, 2=female)
 */
public char getSex() {
    return Sex;
}

/**
 * Get Number of Subject which 'll test
 */
public char getNumSubject() {
    return NumSubject;
}

/**
 * Get each subject to test
 */
public Vector getSubject() {
    return Subject;
}

/**
 * Return filename of faculty and subject file from Enroll.cfg, keyword "FacultySubjectFile="
 */
private static String getFacSubFileFromConfig() {
    EnrollProperties a = new EnrollProperties();
    String sname = new String(a.getProperty("FacultySubjectFile"));
    return sname;
}

/**
 * Return max records of faculty and subject file from Enroll.cfg, keyword "MAXFaculty="
 */

```

```

private static int getMaxFacultyFromConfig() {
    EnrollProperties a = new EnrollProperties();
    String sname = new String(a.getProperty("MAXFaculty"));
    return Integer.parseInt(sname);
}

/**
 * Return max length of records of faculty and subject file from Enroll.cfg, keyword "MAXFacSubRecord="
 */
private static int getRecordLenFromConfig() {
    EnrollProperties a = new EnrollProperties();
    String sname = new String(a.getProperty("MAXFacSubRecord"));
    return Integer.parseInt(sname);
}
}

```

คลาส EnrollProperties

```

/**
 * EnrollProperties.java
 * use to store program's setting and let others class see them
 *
 * @author Jakrit Siriwanakul
 * @version 1.0 8.4.41
 * @exception
 * @since JDK1.0
 */
import java.util.Hashtable;
import java.util.StringTokenizer;
import java.util.Properties;
import java.lang.StringBuffer;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

public class EnrollProperties extends Properties {
    /**

```

```
* Construct EnrollProperties object.  
*  
*/  
EnrollProperties() {  
    super();  
    try {  
        super.load(new FileInputStream("Enroll.conf"));  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}  
  
/**  
 * Converts to a string representing the data in EnrollProperties.  
 *  
 * @return a string representation of the EnrollProperties.  
 */  
public String toString() {  
    return new String("EnrollProperties@"+java.lang.Integer.toHexString(this.hashCode())  
        +":"+super.toString());  
}  
}
```

ภาคผนวก จ

ตัวอย่างคลาสในโปรแกรมเซิร์ฟเวอร์

คลาส ConnectionManager

```
// File: GenericServer.java
// Author: Thomas Lea (leat@goodnet.com, http://www.goodnet.com/~leat)
// Date: 12/7/95
// Version: 1.0 (for Beta JDK API)
// Copyright (c) 1995 Thomas Lea. All Rights Reserved.
//
// Please feel free to take this program and modify it to your hearts content.
// The only thing I ask is that my name and the above version/copyright information
// stay at the top of this module.
// If you like this example, please send me some mail and let me know
// how it helped!

import java.net.*;
import java.io.*;
import java.util.*;
import java.lang.*;

// This guy waits for a connection then spawns a ServerConnection to deal with it
public class ConnectionManager extends Thread {
    private ServerConnection InUnNamed; // store in _my_thread
    private static int _port;
    private static Vector _my_threads = new Vector(5,2); //size of 5 initially, grow by 2
    private ServerSocket _main_socket = null;
    private IAddOnServer _AddOn = null;
    private String _TS_name = null;
    private String _ENC_name = null;
    /**
     * Create CM with <code>port</code>, then waiting for connection
     * @param port server's port
     */
    public ConnectionManager(int port) {
```



```

        _port = port;
    }

    /**
     * Choose name of class which will be loaded when send/receive message via
     <code>CommProxy</code> object.
     * @param ENC class name which implement <code>IEncode</code> interface.
     */
    public void setEncodeClass(String ENC){
        _ENC_name = ENC;
    }

    /**
     * Choose name of class which will work in each <code>ServerConnecton</code> object.
     * @param TS class name which implement <code>ITaskSel</code> interface.
     */
    public void setTaskClass(String TS){
        _TS_name = TS;
    }

    /**
     * Add object that each connection may use
     * @param addit object that implement <code>IAddOnServer</code> interface
     */
    public void setAddOnObject(IAddOnServer addit) {
        _AddOn = addit;
    }

    /**
     * Initial CM before wait connection
     */
    public void run() {
        if ( _AddOn != null ) {
            _AddOn.initialize();
        }
        System.out.println("CM:Server Ready.");
    }

```

```

        serveRequests();
    }

/**
 * Service any request, spawn <code>ServerConnection</code> object
 */
private void serveRequests() {
    try {
        _main_socket = new ServerSocket(_port);
    } catch(Exception e) {
        System.err.println("CM:Exception\n" + e);
        System.exit(1);
    }
    ServerConnection temp_sc = null;
    while (true) {
        try {
            Socket this_connection = _main_socket.accept();
            temp_sc = new ServerConnection(this_connection,this);
            temp_sc.setAddOnObject(_AddOn);
            temp_sc.setTaskClass(_TS_name);
            temp_sc.setEncodeClass(_ENC_name);
            _my_threads.addElement(temp_sc);    // add this SC to my list
            temp_sc.start();
            System.out.println("CM:get connected: " + temp_sc.getInetAddress()
+ "*" + temp_sc.getPort());

            System.out.println("CM:connected client = " + _my_threads.size());
            System.out.flush();
        } catch(Exception e) {
            System.err.println("CM:Exception\n" + e);
            System.err.flush();
        }
    }
}

/**
 * Called by each <code>ServerConnection</code> object when object terminated

```

```

    * @param sc <code>ServerConnection</code> object
    */
    public void notifyRemove(ServerConnection sc) {
        _my_threads.removeElementAt(_my_threads.indexOf(sc));
        System.out.println("CM:left client = " + _my_threads.size());
        System.out.flush();
    }

    /**
     * Return number of client that connect to this server
     * @return number of connected client.
     */
    public int getNumConnection() {
        return ConnectionManager._my_threads.size();
    }

    /**
     * Return <code>java.util.Vector</code> of ServerConnection object.
     * @return list of each connection.
     */
    public Vector getConnectionList() {
        return _my_threads;
    }
}

```

คลาส ServerConnection

```

// File: GenericServer.java
// Author: Thomas Lea (leat@goodnet.com, http://www.goodnet.com/~leat)
// Date: 12/7/95
// Version: 1.0 (for Beta JDK API)
//
// Copyright (c) 1995 Thomas Lea. All Rights Reserved.
//
// Please feel free to take this program and modify it to your hearts content.
// The only thing I ask is that my name and the above version/copyright information
// stay at the top of this module.

```

```
// If you like this example, please send me some mail and let me know
// how it helped!
```

```
import java.net.*;
import java.io.*;
import java.util.*;
import java.lang.*;
```

```
public class ServerConnection extends Thread {
    private CommProxy myClient;
    private ITaskSel myTask = null;
    private Socket _mysocket;
    private IAddOnServer _AddOn = null;
    private String _TS_name = null;
    private String _ENC_name = null;
    private ConnectionManager _cm = null;

    /**
     * Constructor, create ClientProxy when init
     * @param s <code>java.net.Socket</code>
     * @param cm <code>ConnectionManager</code>
     */
    public ServerConnection(Socket s, ConnectionManager cm) {
        _mysocket = s;
        _cm = cm;
        // init each object that need to connect to this ServerConnection
        try {
            myClient = new CommProxy(_mysocket);
        } catch (IOException e) {
            System.err.println("Can't create ServerConnection object:"+e);
            System.err.flush();
            e.printStackTrace();
            System.exit(1);
        }
    }
}
```

```

/**
 * Create Task class which be used. if already have <code>IAddOn</code>, send it to Task class.
 * @param TS name of class that implement <code>ITaskSel</code>
 */
public void setTaskClass(String TS) {
    _TS_name = TS;
    // create task class here (with interface ITaskSel
    try {
        myTask = (ITaskSel)Class.forName(_TS_name).newInstance();
        myTask.setConnectionManager(_cm);
    } catch (Exception e1) {
        System.err.println("Exception in ServerConnection class,setTaskClass()!");
        e1.printStackTrace();
        System.err.println("ServerConnection class,setTaskClass(): exit with error.");
        System.err.flush();
        System.exit(1);
    }
    if (_AddOn!=null) myTask.setAddOnObject(_AddOn);
}

/**
 * Set name of encode class for <code>CommProxy</code>
 * @param ENC name of class that implement <code>IEncode</code>
 */
public void setEncodeClass(String ENC) {
    _ENC_name = ENC;
    // send encoded class name to ClientProxy
    myClient.setEncodeClass(_ENC_name);
}

/**
 * Set add-on object, to send to Task class
 * @param _addit <code>IAddOn</code> object
 */
public void setAddOnObject(IAddOnServer _addit) {
    _AddOn = _addit;    // will send to TaskSel, let them do by their self
}

```

```

        if (myTask!=null) myTask.setAddOnObject(_AddOn);
    }

/**
 * Return Task status from Task class
 * @return <code>java.lang.String</code> about what <code>ITaskSel</code> done.
 */
public String getTaskStatus() {
    return myTask.getTaskStatus();
}

/**
 * Override method run() in <code>java.lang.Thread</code>
 */
public void run() {
    System.out.println("Connected to: " + myClient.getInfo());
    while(true) {
        try {
            //Lets get busy!
            if(doServerWork()==false) break; //O+
        } catch ( Exception e ) {
            System.err.println( "Exception:\n" + e );
            //We are outta here....
            break;
        }
    }

    try {
        myClient.close();
    } catch(Exception e) {
        System.err.println( "Exception:\n" + e );
        System.err.flush();
    }

    System.out.println("Disconnecting: " + myClient.getInfo());
    System.out.flush();
    _cm.notifyRemove(this);
}

```

```

        stop();
    }

/**
 * Manipulate incoming and outgoing message.
 * @return <code>true</code> if normal terminate, <code>false</code> if abnormal terminate.
 */
private boolean doServerWork() {
    //This is where the server actually does its work.
    //when this method finishes, the socket will be closed
    //and this thread will exit.
    //This is just some junk... put your real work here.
    try {
        String inp=null,out=null;
        // get request from ClientProxy
        inp = myClient.receive();
        // check for checksum failed.... receive() return null
        if (inp==null) {
            out = new String("REP");
        } else {
            // send request to ITaskSel
            myTask.setInput(inp);
            // receive result from ITaskSel
            out = myTask.getOutputString();
        }
        // reply request to ClientProxy
        myClient.send(out);
    } catch(Exception e) {
        System.err.println( "doServerWork Exception:"+myClient.getInfo()+" "+ e );
        e.printStackTrace();
        return(false);
    }
    return(true);
}
/**

```

```

    * Finalize,,, send <code>notifyRemove()</code> to <code>ConnectionManager</code>
    */
protected void finalize(){
    System.out.flush();
    System.err.flush();
    _cm.notifyRemove(this);
}

/**
 * Return Internet address of this connection.
 * @return Inet address in string
 */
public String getInetAddress(){
    return _mysocket.getInetAddress().toString();
}

/**
 * Return port of this connection.
 * @return port number
 */
public int getPort(){
    return _mysocket.getPort();
}
}

```

คลาส CommProxy

```

/**
 * CommProxy.java
 *
 * @author Jakrit Siriwathanakul
 * @version 1.00, 31.7.41
 * @see
 * @since JDK 1.0

```



```

*/

import java.net.Socket;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.lang.Exception;

/**
 * Use to communicate with client, adding checksum or something...
 */
public class CommProxy {
    private Socket _mysocket; /* socket to send/receive from client */
    private InputStream _sinp; /* input stream */
    private OutputStream _sout; /* output stream */
    private byte outbuf[], inbuf[]; /* buffer */
    private int outlen,inlen; /* message length */
    private IEncode _encoder = null; /* add-on */

    /**
     * Set name of encode class for <code>CommProxy</code>
     * @param ENC name of class that implement <code>IEncode</code>
     */
    public void setEncodeClass(String ENC){
        try {
            _encoder = (IEncode)Class.forName(ENC).newInstance();
        } catch (Exception e1) {
            System.err.println("Exception in CommProxy class,setEncodeClass(!");
            e1.printStackTrace();
            System.err.println("CommProxy class,setEncodeClass(): continue without IEncode class.[*+ENC+*]");
            System.err.flush();
            _encoder = null;
        }
    }
}

/**

```

```

    * Close this connection.
*/
public void close() throws IOException {
    _mysocket.close();
}

/**
 * Return IP address and Port which connected to.
*/
public String getInfo() {
    return new String(_mysocket.getInetAddress() + ":" + _mysocket.getPort());
}

/**
 * Send <code>java.lang.String</code> to other site.
 * @param msg string to send.
 * @exception IOException when write error.
*/
public void send(String msg) throws IOException {
    String msg_enc;
    /* use IEncode class if available */
    if (_encoder!=null) msg_enc = _encoder.encode(msg);
    else msg_enc = msg;
    /* convert to array of byte and add '\0' at the end of array */
    outlen = msg_enc.length();
    outbuf = new byte[outlen+1];
    msg_enc.getBytes(0,outlen,outbuf,0);
    outbuf[outlen] = (byte)\0;
    /* put into output stream */
    _sout.write(outbuf);
}

/**
 * Return <code>java.lang.String</code> from other site.
 * @exception IOException when read error.
*/

```

```

public String receive() throws IOException {
    inbuf = new byte[1024];
    inlen = _sinp.read(inbuf,0,1024);
    String instr = new String(inbuf,0,0,inlen-1); // convert array of byte to String
    String destr = _encoder.decode(instr); // if Checksum failed return null
    return destr;
}
/**
 * Create agent to connect to client,
 *
 * @param _mysock Socket to client
 * @exception IOException when can't get in/out stream from <code>java.net.Socket</code>.
 *
 */
CommProxy(Socket _mysock) throws IOException {
    _mysocket = _mysock; /* socket which will use to communicate with other site */
    _sinp = _mysocket.getInputStream(); /* use to receive */
    _sout = _mysocket.getOutputStream(); /* use to send */
}
}

```

ประวัติผู้เขียน

นายจักรกฤษ ศิริวัฒน์กุล เกิดวันที่ 22 เมษายน พ.ศ. 2516 ที่เขตปทุมวัน จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิทยาศาสตร์บัณฑิต สาขาฟิสิกส์ ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2538 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต ที่ จุฬาลงกรณ์มหาวิทยาลัยเมื่อ พ.ศ. 2538

