FSE598 前沿计算技术

模块 2 数据与数据处理
单元 4 面向对象的编程
第 3 讲 案例研究

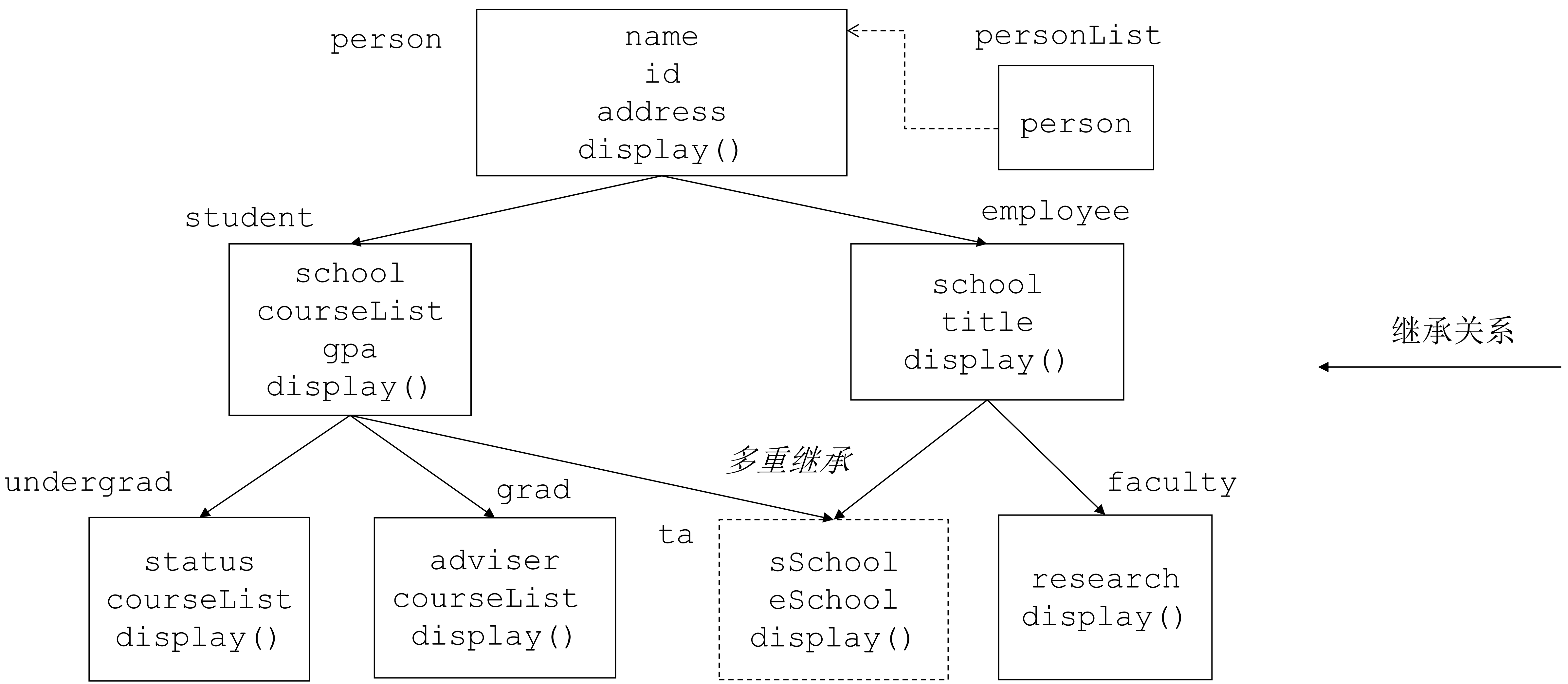
讲座的英文版内容基于本书：
Y. Chen 《编程语言入门：C、C++、Scheme、Prolog、C# 和 Python 编程》(Introduction to Programming Languages: Programming in C, C++, Scheme, Prolog, C#, and Python)，第 6 版，Kendall Hunt Publishing Company，2019 年。
https://www.public.asu.edu/~ychen10/book/IntroPl.html

本讲大纲

学习

- ❑ 将所有面向对象的特征放在一起
- ❑ 类方法和全局函数
- ❑ 使用和不使用继承
- ❑ 多态性和重写方法
- ❑ 用于选择不同函数的输入/输出

案例研究：继承和多重继承



定义基类：第 1 页

```
# person class is the base class
class person:
    name = ""
    id = 0
    address = ""
    def __init__(self, name, id, address):
        self.name = name
        self.id = id
        self.address = address
    def display(self):
        print(F'name = {self.name}, id = {self.id}, address
= {self.address}')
```

类属性

初始化类属性

待重写的
类方法

定义 Child 类：第 2 页

name = ""
id = 0
address = ""

```
class student(person):
    school = ""
    courseList = []
    gpa = 0.0
    def __init__(self, name, id, address, school):
        person.__init__(self, name, id, address)
        self.school = school
    def addCourse(self, courseName):
        self.courseList.append(courseName)
    def display(self):
        print(F'name = {self.name}, id = {self.id}, address = {self.address}')
        print(F'\t courseList = {self.courseList}, school = {self.school}, gpa = {self.gpa}')
```

student 类继承 person 类

这些成员已在“person”类中定义。这里无需在“student”类中定义。

定义 Child 类：第 3 页

employee 类继承 person 类

```
class employee(person):
    school = ""
    title = ""
    def __init__(self, name, id, address, school, title):
        person.__init__(self, name, id, address)
        self.school = school
        self.title = title
    def display(self):
        print(F'name = {self.name}, id = {self.id}, address = {self.address}')
        print(F'\t school = {self.school}, title = {self.title}')
```

```
name = ""
id = 0
address = ""
```

这些成员已在“person”类中定义。这里无需在“employee”类中定义。

定义 Student 类的 Child 类：第 4 页

undergrad 类继承
student 类

name = ""
id = 0
address = ""
school = ""
gpa = 0.0

这些成员已在“person”和“student”类中定义。这里无需在“undergrad”类中定义。

```
class undergrad(student):
    courseList = [] # necessary to have a different list for this class
                    # For int, float, and str, they can be inherited
    def __init__(self, name, id, address, school):
        student.__init__(self, name, id, address, school)
    def display(self):
        print(F'name = {self.name}, id = {self.id}, address = {self.address}')
        print(F'\t courseList = {self.courseList}, school = {self.school}, gpa = {self.gpa}')
```

定义 Student 类的 Child 类：第 5 页

grad 类继承 student 类

name = ""
id = 0
address = ""
school = ""
gpa = 0.0

这些成员已在“person”和“student”类中定义。这里无需在 “grad” 类中定义。

```
class grad(student):
    adviser = ""
    courseList = [] # necessary to have a different list for this class
    def __init__(self, name, id, address, school, adviser):
        student.__init__(self, name, id, address, school)
        self.adviser = adviser
    def display(self):
        print(F'name = {self.name}, id = {self.id}, address = {self.address}')
        print(F'\t courseList = {self.courseList}, school = {self.school}, gpa = {self.gpa}')
        print(F'\t adviser = {self.adviser}')
```

多重继承：第 7 页

```
class ta(student, employee):
    sSchool = ""
    eSchool = ""
    courseList = [] # necessary to have a different list for this class
    def __init__(self, name, id, address, sSchool, eSchool, title):
        self.sSchool = sSchool
        self.eSchool = eSchool
        student.__init__(self, name, id, address, sSchool)
        self.eSchool = eSchool
        self.title = title
    def display(self):
        print(F'name = {self.name}, id = {self.id}, address = {self.address}')
        print(F'\t student school = {self.sSchool}, courseList = {self.courseList}, gpa = 
{self.gpa}')
        print(F'\t employee school = {self.eSchool}, title = {self.title}')
```

ta 类继承 student 和 employee 类

name = ""
id = 0
address = ""
~~school = ""~~
gpa = 0.0
~~school = ""~~
title = ""

有来自 student 的 school 和来自 employee 的 school。两者都无法使用。

继承示例：第 8 页

faculty 类继承 employee 类

```
class faculty(employee):
    research = ""
    def __init__(self, name, id, address, school, title, research):
        employee.__init__(self, name, id, address, school, title)
        self.research = research
    def display(self):
        print(F'name = {self.name}, id = {self.id}, address = {self.address}, school = {self.school}, title = {self.title}')
        print(F'\t research = {self.research}')

personList = []  # global list to hold all persons
```

name = ""
id = 0
address = ""
school = ""
title = ""

这些成员已在“person”和“employee”类中定义。这里无需在“faculty”类中定义。

Display Menu in main()

```
def main():
    selection = 'a'
    while True:
        print("Please enter your selection:")
        print("\t p: add a new person to the person list")
        print("\t s: add a new student to the person list")
        print("\t e: add a new employee to the person list")
        print("\t u: add a new undergrad to the person list")
        print("\t g: add a new grad to the person list")
        print("\t t: add a new ta to the person list")
        print("\t f: add a new faculty to the person list")
        print("\t c: add a new course into a student's course list")
        print("\t d: display person list of all types")
        print("\t r: remove a person of any type from the list")
        print("\t q: quit")
        selection = input()
        if selection == 'q':
            break
        executeAction(selection)
```

定义 faculty 类，但不使用继承？

如果我们不使用继承，我们也可以做到，但是……这不是一个好主意

```
class faculty:
    name = ""
    id = 0
    address = ""
    school = ""
    title = ""
    research = ""
    def __init__(self, name, id, address, school, title, research):
        employee.__init__(self, name, id, address, school, title)
        self.research = research
    def display(self):
        print(F'name = {self.name}, id = {self.id}, address = {self.address}, school = 
{self.school}, title = {self.title}')
        print(F'\t research = {self.research}')
```

主要问题是：如果我们更改 person 类或 employee 类，这些成员不会自动更改。

全局函数 - 处理输入选择：第 9 页

```
# Global function definitions
def executeAction(c):
    if(c == 'p'):
        name = input("Please enter the person's name: ")
        if(searchPerson(name) == -1):
            id = input("Please enter the person's id: ")
            address = input("Please enter the person's address: ")
            addPerson(name, id, address)
            print("\nPerson successfully added to the list!")
        else:
            print("\nThat person is already on the list!\n")
```

全局函数 - 处理输入选择：第 10 页

```
elif(c == 's'):
    name = input("Please enter the student's name: ")
    if(searchPerson(name) == -1):
        id = input("Please enter the student's id: ")
        address = input("Please enter the student's address: ")
        school = input("Please enter the student's school: ")
        addStudent(name, id, address, school)
        print("\nStudent successfully added to the list!")
    else:
        print("\nThat person is already on the list!\n")
```

全局函数 - 处理输入选择：第 11 页

```
elif(c == 'e'):
    name = input("Please enter the employee's name: ")
    if(searchPerson(name) == -1):
        id = input("Please enter the employee's id: ")
        address = input("Please enter the employee's address: ")
        school = input("Please enter the employee's school: ")
        title = input("Please enter the employee's title: ")
        addEmployee(name, id, address, school, title)
        print("\nEmployee successfully added to the list!")
    else:
        print("\nThat person is already on the list!\n")
```

全局函数 - 处理输入选择：第 12 页

```
elif(c == 'u'):
    name = input("Please enter the undergraduate student's name: ")
    if(searchPerson(name) == -1):
        id = input("Please enter the undergraduate student's id: ")
        address = input("Please enter the undergraduate student's address: ")
        school = input("Please enter the undergraduate student's school: ")
        addUndergrad(name, id, address, school)
        print("\nStudent successfully added to the list!")
    else:
        print("\nThat person is already on the list!\n")
```

全局函数 - 处理输入选择：第 13 页

```
elif(c == 'g'):
    name = input("Please enter the grad student's name: ")
    if(searchPerson(name) == -1):
        id = input("Please enter the grad student's id: ")
        address = input("Please enter the grad student's address: ")
        school = input("Please enter the grad student's school: ")
        adviser = input("Please enter the grad student's adviser: ")
        addGrad(name, id, address, school, adviser)
        print("\nStudent successfully added to the list!")
    else:
        print("\nThat person is already on the list!\n")
```

全局函数 - 处理输入选择：第 14 页

```
elif(c == 'f'):
    name = input("Please enter the faculty's name: ")
    if(searchPerson(name) == -1):
        id = input("Please enter the faculty's id: ")
        address = input("Please enter the faculty's address: ")
        school = input("Please enter the faculty's school: ")
        title = input("Please enter the faculty's title: ")
        research = input("Please enter the faculty's research: ")
        addFaculty(name, id, address, school, title, research)
        print("\nFaculty successfully added to the list!")
    else:
        print("\nThat person is already on the list!\n")
```

全局函数 - 处理输入选择：第 15 页

```
elif(c == 't'):
    name = input("Please enter the TA's name: ")
    if(searchPerson(name) == -1):
        id = input("Please enter the TA's id: ")
        address = input("Please enter the TA's address: ")
        sSchool = input("Please enter the TA's student school: ")
        title = input("Please enter TA employee's title: ")
        eSchool = input("Please enter the TA's employment school: ")
        addTa(name, id, address, sSchool, title, eSchool)
        print("\nTA successfully added to the list!")
    else:
        print("\nThat person is already on the list!\n")
```

全局函数 - 处理输入选择：第 16 页

```
elif(c == 'c’):  # add a course to a student record
    name = input("Please enter the student's name: ")
    i = searchPerson(name)
    if(i == -1):
        print("\nThat student does not exist on the list!\n")
    elif isinstance(personList[i], student) or issubclass(personList[i], student):
        course = input("Please enter a course name: ")
        gp = input("Please enter the grade point of the course: ")
        courses = len(personList[i].courseList)
        personList[i].gpa = round((courses*personList[i].gpa + float(gp)) / (courses+1), 2)
        personList[i].addCourse(course)
        print(F"\n{course} successfully added to the course list of {personList[i].name}!")
    else:
        print("\nThat name you entered is not a student, undergrad, grad, or ta!\n")
```

全局函数 - 处理输入选择：第 17 页

```
elif(c == 'd'): # Display
    displayList(personList)
elif(c=='r'):
    name = input("Please enter person's name: ")
    if(searchPerson(name) == -1):
        print("\nThat Person name does not exist! \n\n")
    else:
        removePerson(name)
        print("\nPerson successfully removed from the list! \n\n")
```

全局函数 - 前向声明：第 18 页

```
def addPerson(name, id, address):  #Adds person to personList
    personList.append(person(name, id, address))
def addStudent(name, id, address, school):  #Adds student to personList
    personList.append(student(name, id, address, school))
#Adds employee to personList
def addEmployee(name, id, address, school, title):
    personList.append(employee(name, id, address, school, title))
#Adds undergrad to personList
def addUndergrad(name, id, address, school):
    personList.append(undergrad(name, id, address, school))
#Adds grad to personList
def addGrad(name, id, address, school, adviser):
    personList.append(grad(name, id, address, school, adviser))
```

全局函数 - 添加 Faculty：第 19 页

```
#Adds faculty to personList
def addFaculty(name, id, address, school, title, research):
    personList.append(faculty(name, id, address, school, title, research))
#Adds TA to personList
def addTa(name, id, address, sSchool, title, eSchool):
    personList.append(ta(name, id, address, sSchool, title, eSchool))
#Searches for person by name and returns their object
def searchPerson(name):
    i = 0
    for p in personList:
        if p.name == name:
            return i # return the position in the list
        else:
            i += 1
    return -1 # not found
```

全局函数 - 显示和删除：第 20 页

```
#Displays list of persons of all types in the list
def displayList(list):
    for p in list:
        p.display() # It will call the display() in p object

def removePerson(name):  ##Removes person of given name from list
    for p in personList:
        if p.name == name:
            personList.remove(p) # remove p from list

if __name__ == "__main__":
    main()
```

示例输入和输出第1部分

```
Please enter your selection:
     p: add a new person to the person list
     s: add a new student to the person list
     e: add a new employee to the person list
     u: add a new undergrad to the person list
     g: add a new grad to the person list
     t: add a new ta to the person list
     f: add a new faculty to the person list
     c: add a new course into a student's course list
     d: display person list of all types
     r: remove a person of any type from the list
     q: quit
s
Please enter the student's name: John
Please enter the student's id: 1234
Please enter the student's address: 2nd Street
Please enter the student's school: CS

Student successfully added to the list!
```

```
Please enter your selection:
         p: add a new person to the person list
         s: add a new student to the person list
         e: add a new employee to the person list
         u: add a new undergrad to the person list
         g: add a new grad to the person list
         t: add a new ta to the person list
         f: add a new faculty to the person list
         c: add a new course into a student's
course list
         d: display person list of all types
         r: remove a person of any type from the
list
         q: quit
g
Please enter the grad student's name: Jenn
Please enter the grad student's id: 4567
Please enter the grad student's address: 3rd Ave
Please enter the grad student's school: EE
Please enter the grad student's adviser: Smith

Student successfully added to the list!
```

示例输入和输出第2部分

```
Please enter your selection:
     p: add a new person to the person list
     s: add a new student to the person list
     e: add a new employee to the person list
     u: add a new undergrad to the person list
     g: add a new grad to the person list
     t: add a new ta to the person list
     f: add a new faculty to the person list
     c: add a new course into a student's course list
     d: display person list of all types
     r: remove a person of any type from the list
     q: quit
d
name = John, id = 1234, address = 2nd Street
         courseList = [], school = CS, gpa = 0.0
name = Jenn, id = 4567, address = 3rd Ave
         courseList = [], school = EE, gpa = 0.0
         adviser = Smith
```

```
Please enter your selection:
     p: add a new person to the person list
     s: add a new student to the person list
     e: add a new employee to the person list
     u: add a new undergrad to the person list
     g: add a new grad to the person list
     t: add a new ta to the person list
     f: add a new faculty to the person list
     c: add a new course into a student's course list
     d: display person list of all types
     r: remove a person of any type from the list
     q: quit
c
Please enter the student's name: John
Please enter a course name: CSE110
Please enter the grade point of the course: 3.8

CSE110 successfully added to the course list of John!
```

示例输入和输出第3部分

```
Please enter your selection:
         p: add a new person to the person list
         s: add a new student to the person list
         e: add a new employee to the person list
         u: add a new undergrad to the person list
         g: add a new grad to the person list
         t: add a new ta to the person list
         f: add a new faculty to the person list
         c: add a new course into a student's course list
         d: display person list of all types
         r: remove a person of any type from the list
         q: quit
c
Please enter the student's name: Jenn
Please enter a course name: CSE205
Please enter the grade point of the course: 3.7

CSE205 successfully added to the course list of Jenn!
```

```
Please enter your selection:
         p: add a new person to the person list
         s: add a new student to the person list
         e: add a new employee to the person list
         u: add a new undergrad to the person list
         g: add a new grad to the person list
         t: add a new ta to the person list
         f: add a new faculty to the person list
         c: add a new course into a student's course list
         d: display person list of all types
         r: remove a person of any type from the list
         q: quit
t
Please enter the TA's name: Lee
Please enter the TA's id: 9876
Please enter the TA's address: Rural Road
Please enter the TA's student school: Bio
Please enter the TA student's adviser: Mary
Please enter TA employee's title: GA
Please enter the TA's employment school: CS

TA successfully added to the list!
```

示例输入和输出第4部分

```
Please enter your selection:
         p: add a new person to the person list
         s: add a new student to the person list
         e: add a new employee to the person list
         u: add a new undergrad to the person list
         g: add a new grad to the person list
         t: add a new ta to the person list
         f: add a new faculty to the person list
         c: add a new course into a student's course list
         d: display person list of all types
         r: remove a person of any type from the list
         q: quit
d
name = John, id = 1234, address = 2nd Street
         courseList = ['CSE110'], school = CS, gpa = 3.8
name = Jenn, id = 4567, address = 3rd Ave
         courseList = ['CSE205'], school = EE, gpa = 3.7
         adviser = Smith
name = Lee, id = 9876, address = Rural Road
         student school = , courseList = [], gpa = 0.0
         employee school = , title = CS
```

```
Please enter your selection:
         p: add a new person to the person list
         s: add a new student to the person list
         e: add a new employee to the person list
         u: add a new undergrad to the person list
         g: add a new grad to the person list
         t: add a new ta to the person list
         f: add a new faculty to the person list
         c: add a new course into a student's course list
         d: display person list of all types
         r: remove a person of any type from the list
         q: quit
c
Please enter the student's name: Lee
Please enter a course name: CSE445
Please enter the grade point of the course: 3.5

CSE445 successfully added to the course list of Lee!
```

示例输入和输出第5部分

```
Please enter your selection:
         p: add a new person to the person list
         s: add a new student to the person list
         e: add a new employee to the person list
         u: add a new undergrad to the person list
         g: add a new grad to the person list
         t: add a new ta to the person list
         f: add a new faculty to the person list
         c: add a new course into a student's course list
         d: display person list of all types
         r: remove a person of any type from the list
         q: quit
c
Please enter the student's name: Lee
Please enter a course name: CSE520
Please enter the grade point of the course: 3.4

CSE520 successfully added to the course list of Lee!
```

```
Please enter your selection:
         p: add a new person to the person list
         s: add a new student to the person list
         e: add a new employee to the person list
         u: add a new undergrad to the person list
         g: add a new grad to the person list
         t: add a new ta to the person list
         f: add a new faculty to the person list
         c: add a new course into a student's course list
         d: display person list of all types
         r: remove a person of any type from the list
         q: quit
d
name = John, id = 1234, address = 2nd Street
         courseList = ['CSE110'], school = CS, gpa = 3.8
name = Jenn, id = 4567, address = 3rd Ave
         courseList = ['CSE205'], school = EE, gpa = 3.7
         adviser = Smith
name = Lee, id = 9876, address = Rural Road
         student school = , courseList = ['CSE445', 'CSE520'],
gpa = 3.45
         employee school = , title = CS
```

示例输入和输出第6部分

```
Please enter your selection:
         p: add a new person to the person list
         s: add a new student to the person list
         e: add a new employee to the person list
         u: add a new undergrad to the person list
         g: add a new grad to the person list
         t: add a new ta to the person list
         f: add a new faculty to the person list
         c: add a new course into a student's course list
         d: display person list of all types
         r: remove a person of any type from the list
         q: quit
u
Please enter the undergraduate student's name: Lisa
Please enter the undergraduate student's id: 5678
Please enter the undergraduate student's address: Mill Ave
Please enter the undergraduate student's school: CS
Please enter status - freshman, sophomore, junior, or
senior: freshman

Student successfully added to the list!
```

```
Please enter your selection:
         p: add a new person to the person list
         s: add a new student to the person list
         e: add a new employee to the person list
         u: add a new undergrad to the person list
         g: add a new grad to the person list
         t: add a new ta to the person list
         f: add a new faculty to the person list
         c: add a new course into a student's course list
         d: display person list of all types
         r: remove a person of any type from the list
         q: quit
d
name = John, id = 1234, address = 2nd Street
         courseList = ['CSE110'], school = CS, gpa = 3.8
name = Jenn, id = 4567, address = 3rd Ave
         courseList = ['CSE205'], school = EE, gpa = 3.7
         adviser = Smith
name = Lee, id = 9876, address = Rural Road
         student school = , courseList = ['CSE445', 'CSE520'],
gpa = 3.45
         employee school = , title = CS
name = Lisa, id = 5678, address = Mill Ave
         courseList = [], school = CS, gpa = 0.0
         status = freshman
```