Platform-based Programming

#2 Classes

1) Class definition
2) constructor
3) Method definition
4) toString()

2018년 2학기

Program Output

- Make a Java program that can manage students for a school

```
Enter Operation String! add
James 1
Enter Operation String! list
School Name: PNU Student Count: 1
[James, 1학년]

Enter Operation String! add
Brown 2
Enter Operation String! list
School Name: PNU Student Count: 2
[James, 1학년]
[Brown, 2학년]
```

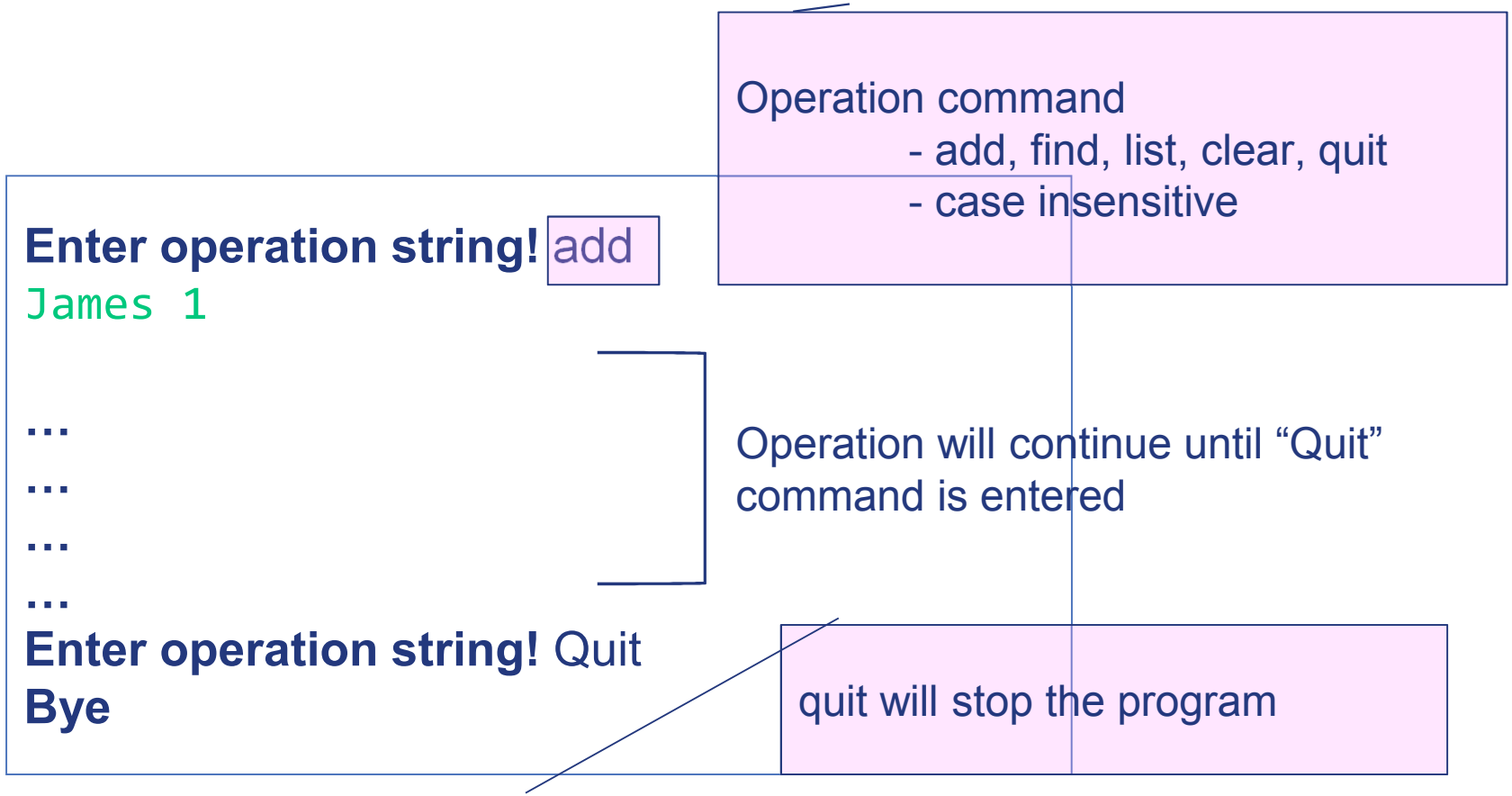
```
Enter Operation String! find
Brown 2
[Brown, 2학년]
Enter Operation String! find
Brown 1
Student Not Found with name Brown and year 1
Enter Operation String! add
Kim 4
Enter Operation String! list
School Name: PNU Student Count: 3
[James, 1학년]
[Brown, 2학년]
[Kim, 4학년]

Enter Operation String! clear
Enter Operation String! list
School Name: PNU Student Count: 0
```

```
Enter Operation String! find
Brown 2
Student Not Found with name Brown and year 2
Enter Operation String! add
Brown 2
Enter Operation String! find
Brown 2
[Brown, 2학년]
Enter Operation String! list
School Name: PNU Student Count: 1
[Brown, 2학년]

Enter Operation String! find
brown 2
Student Not Found with name brown and year 2
Enter Operation String! quit
Bye
```

Problem Description



Program Skeleton

- Write a program by extending the following code!
  - Just add you code! Do not modify/remove the given code

```
enum OperationKind {ADD, LIST, CLEAR, FIND, QUIT, INVALID} ;

public class SchoolTest {
   private static Scanner scanner = new Scanner(System.in);

   public static void main(String[] args) {
            School pnu = new School("PNU", 100) ;

            while ( true ) {
                     OperationKind op = getOperation() ;
                     if ( op == OperationKind.QUIT ) {
                              System.out.println("Bye") ;
                              break;
                     }
                     if ( op == OperationKind.INVALID ) {
                              System.out.println("Invalid Operation!") ;
                              continue ;
                     }
```

```
enum OperationKind {ADD, LIST, CLEAR, FIND, QUIT, INVALID} ;
public class SchoolTest {
   private static Scanner scanner = new Scanner(System.in);

   public static void main(String[] args) {
          School pnu = new School("PNU", 100) ;
          while ( true ) {
                 // same code in the previous slide

                 switch ( op ) {
                 case ADD : {
                        Student newStudent = createStudent() ;
                        pnu.addStudent(newStudent) ;
                        break ;
                 }
                 case FIND: findStudent(pnu) ; break ;
                 case CLEAR: pnu.removeAllStudent() ; break ;
                 case LIST: System.out.println(pnu) ; break ;
                 }
          }
   }
```

```
private static OperationKind getOperation() {
        System.out.print("Enter Operation String! ") ;
        String operation = scanner.next() ;

        OperationKind kind = OperationKind.INVALID ;
        if ( operation.equalsIgnoreCase("ADD"))
                kind = OperationKind.ADD ;
        else if ( operation.equalsIgnoreCase("LIST"))
                kind = OperationKind.LIST ;
        else if ( operation.equalsIgnoreCase("FIND"))
                kind = OperationKind.FIND ;
        else if ( operation.equalsIgnoreCase("CLEAR"))
                kind = OperationKind.CLEAR ;
        else if ( operation.equalsIgnoreCase("QUIT"))
                kind = OperationKind.QUIT ;
        return kind ;
}
```

```
private static void findStudent(final School school) {
 String studentName = scanner.next() ;
 int schoolYear = scanner.nextInt() ;
 Student foundStudent =
        school.findStudent(studentName, schoolYear) ;
 if ( foundStudent != null )
        System.out.println(foundStudent) ;
 else
        System.out.println("Student Not Found with name " + studentName +
                " and year " + schoolYear) ;
}
private static Student createStudent() {
 String studentName = scanner.next() ;
 int schoolYear = scanner.nextInt() ;
 return new Student(studentName, schoolYear) ;
}
```

```
Enter Operation String! find
Brown 2
[Brown, 2학년]
```

```
// School.java
public class School {
   private String name ;
   private int limit ;

   private Student[] students ;
   private int studentCount ;

   public String toString() {
          String msg = "School Name: " + name + " Student Count: “
                 + studentCount + "\n" ;

          for ( int i = 0 ; i < studentCount ; i ++ ) {
                 msg += "\t" + students[i] + "\n" ;
          }
          return msg ;
   }

   // other methods including constructor
}
```

```
School Name: PNU Student Count: 3
[James, 1학년]
[Brown, 2학년]
[Kim, 4학년]
```

```
// Student.java
public class Student {
   private String name ;
   private int year ;

            // other methods including constructor
}
```