

(고급프로그래밍 과제 2)

09/25/2021 (MM/DD/YYYY)

32212488 안태현 (고급프로그래밍 1 분반)

Sections

1. Description	3
2. Initial Class Structure	4
3. Problem	7
4. Solution	8
5. Exception Handling	10
6. Data File	12
7. Program Execution	13
7.1. Execution with Invalid Files	13

Section 1. Description

다음과 같은 형식을 따르는 파일이 있다.

```
6,Algebra,Calculus,Geometry,Statistics,Physics,Chemistry
Student1,76,25,14,41,52,48
Student2,12,95,72,83,29,79
Student3,64,64,18,57,72,69
...
```

파일의 첫 번째 라인은 평가 항목의 수와 평가 항목 명칭의 리스트를 나타낸다. 두 번째 라인부터는 학생 이름과 각 항목들에 대한 점수들로 구성된다. 각각의 값들은 콤마(,)로 구분된다.

이때, 파일에 저장된 학생들의 성적 정보를 읽고 이의 내용을 화면에 보이는 프로그램을 작성하시오. *(단, 최대한 객체 지향적으로 작성할 것.)*

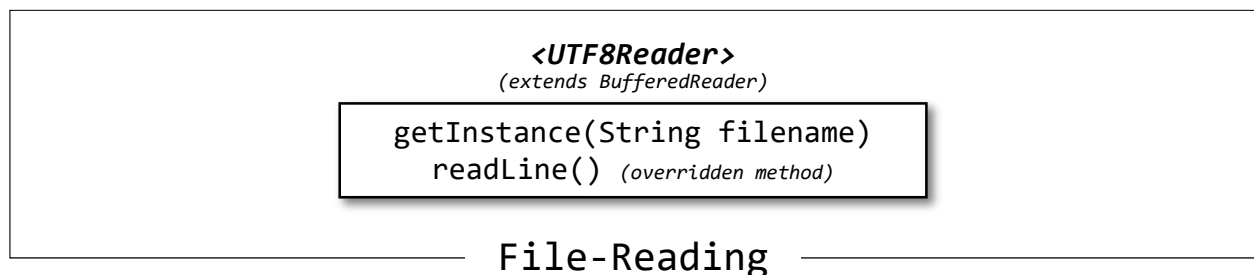
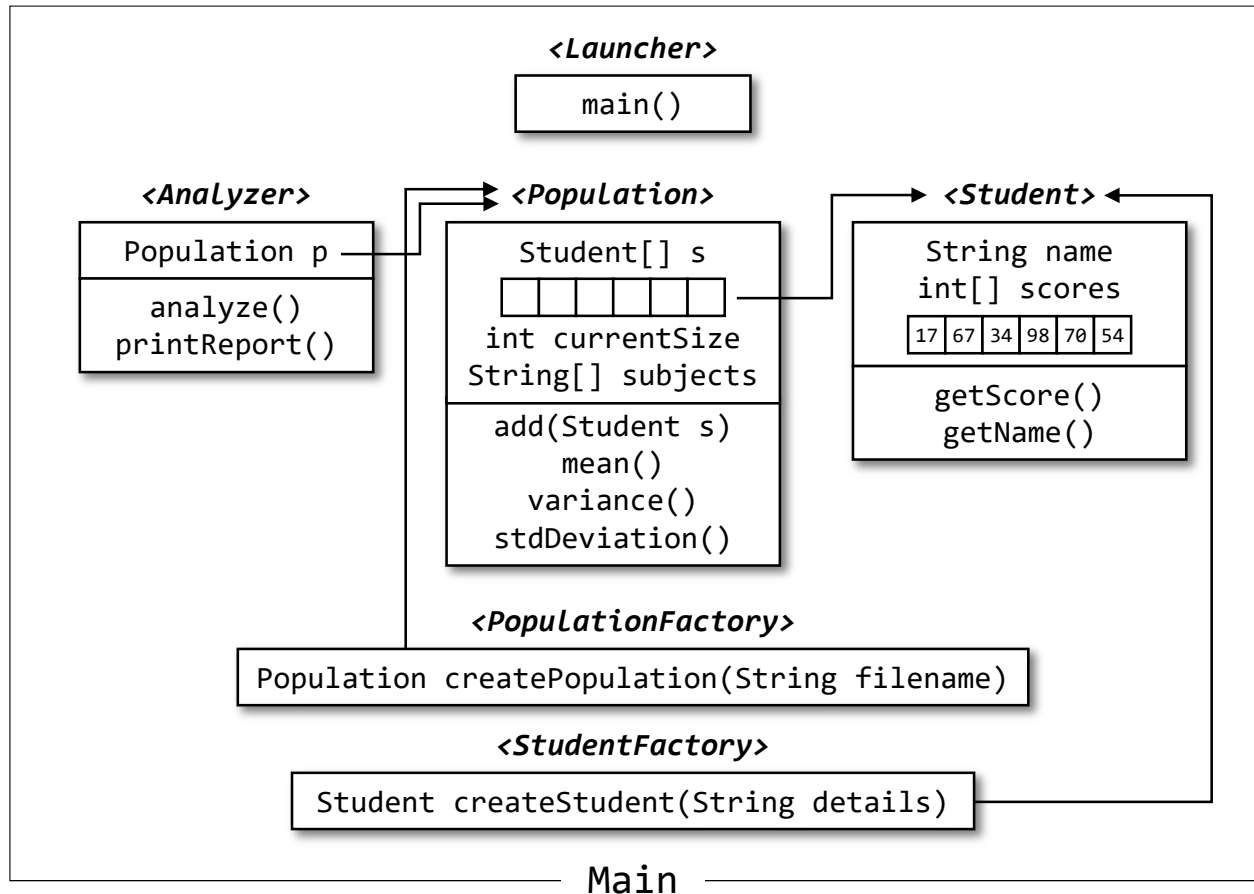
각 항목의 위치와 간격을 사용자들이 읽기 쉽게 화면에 출력하고, 마지막 행에서는 각 항목의 평균 값을 출력하시오.

(예시 출력 결과)

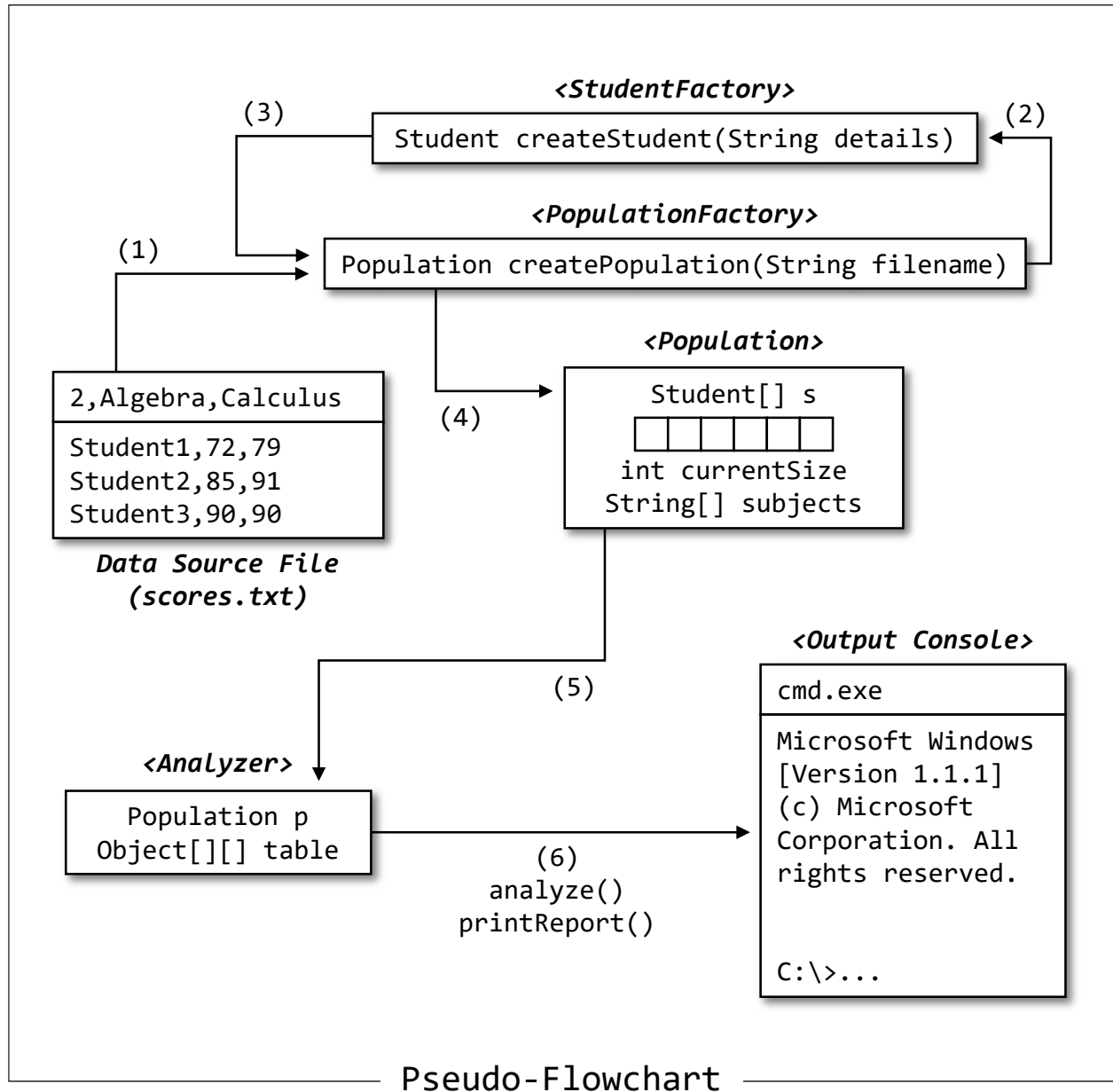
	Algebra	Calculus	Geometry	Statistics	Physics	Chemistry
Student1	76	25	14	41	52	48
Student2	12	95	72	83	29	79
Student3	64	64	18	57	72	69
...						
E(X)	50.57	50.44	50.00	50.27	49.74	48.61

Section 2. Initial Class Structure

어떤 모집단의 학생들의 성적을 조사하는 프로그램이므로, 모집단을 나타낼 'Population' 클래스, 학생을 나타낼 'Student' 클래스, 그리고 조사를 위한 'Analyzer' 클래스가 필요하다. 파일로부터의 생성을 위한 Factory 계열 클래스도 필요하다.



주요 클래스들을 용도에 따라 분류하면 위의 그림과 같다. 아래는 프로그램 실행의 유사-순서도를 그림으로 간략히 표기한 것이다.



(1) 우선 PopulationFactory 클래스의 createPopulation() 메소드를 호출한다. (전달하는 인자는 “scores.txt”)

(2) 모집단 생성 중, StudentFactory 클래스의 createStudent() 메소드를 호출한다.

(3) createStudent() 메소드를 통하여 생성된 Student 객체의 레퍼런스를 반환한다.

(4) 반환받은 Student 레퍼런스를 전부 모집단에 추가하고, 생성된 모집단 객체의 레퍼런스를 반환한다.

(5) (4)에서 반환받은 모집단 객체를 이용하여 Analyzer 객체를 생성한다.

(6) (5)에서 생성한 Analyzer 객체에 대하여 analyze()와 printReport() 메소드를 차례로 호출한다. (출력 스트림에 결과가 뜰 것이다.)

Section 3. Problem

Population 클래스에서 구현한 학생 배열을 초기화하여야 하므로 전체 학생 수를 미리 알아야 한다.

그런데 학생 수를 미리 알 수 있는 방법이 없다. 파일에는 과목의 수만 명시적으로 표기되어있기 때문에 학생 수를 알려면 파일을 끝까지 읽어야 한다. 만일 이 방법을 따른다면 각각의 Student 객체 생성까지 파일을 2 번 읽어야 하므로 효율적인 방법이 아니다.

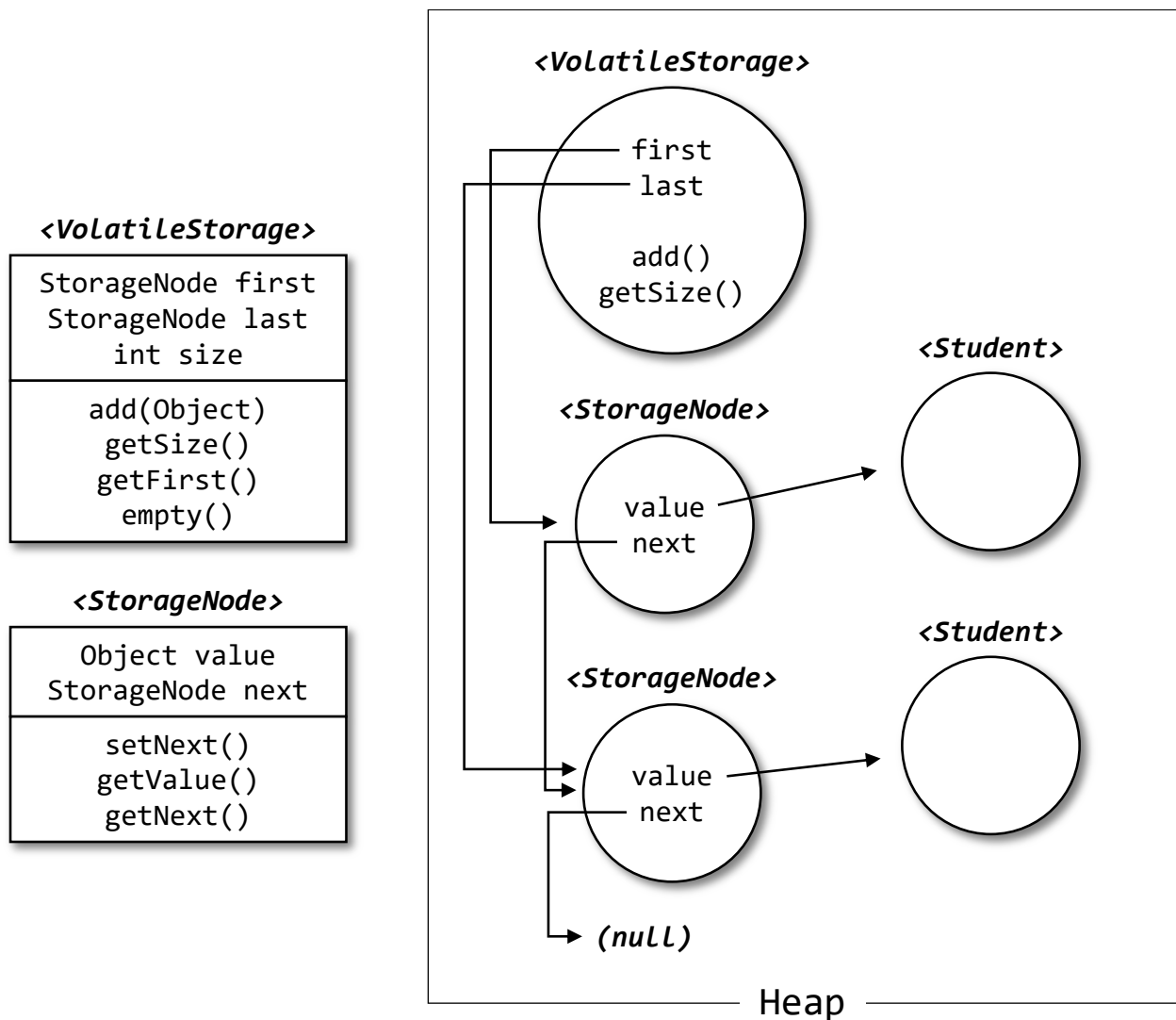
Student 객체 생성과 동시에 배열에 추가할 수 있으면 파일을 1 번만 읽어도 되므로 좋을 것 같다. 그러나 배열의 크기는 중간에 임의로 변경할 수 없으며, 배열의 크기를 매우 크게 하여 초기화하면 메모리 낭비의 가능성이 크다.

(자바에서 제공하는 `java.util.ArrayList` 클래스를 사용할 수 있지만, 제네릭 문법을 사용하지 않는 해결책을 찾아보았다.)

Section 4. Solution

임시로 Student 객체를 메모리에 저장하여야 한다. 무작위로 객체를 생성하면 GC 에 의하여 처리되기 때문에 일시적으로 객체를 어떤 레퍼런스가 참조하고 있어야 한다.

따라서 VolatileStorage 클래스와 StorageNode 클래스를 도입하였다. 이중 말단 연결 리스트의 구조이므로, 배열과 달리 크기가 동적으로 변할 수 있다. 하지만 배열만큼 인덱싱이 용이하지 않으므로, Student 배열의 초기화를 위해서만 사용할 것이다.



(위의 그림 참조) 파일의 각 줄을 읽으면서 Student 객체를 생성한 뒤, VolatileStorage 클래스의 add() 메소드를 이용하여 다음과 같이 임시 공간에 추가한다.

모든 학생을 다 추가하면, 학생 수는 getSize() 메소드를 통하여 알 수 있고, 생성된 모든 Student 객체들은 Heap 에 접근 가능한 상태로 남아있기 때문에 Population 의 Student 배열을 다음과 같은 방법으로 초기화할 수 있다.

```
Population p = new Population(storage.getSize());
for (StorageNode node = storage.getFirst(); node != null; node = node.getNext()) {
    Student s = (Student) node.getValue();
    p.add(s);
}
```

임시 저장 공간의 사용이 끝나면 공간을 비워 주어야 한다. VolatileStorage 클래스의 empty() 메소드를 호출하면 된다. 메소드의 구현은 다음과 같다.

```
public void empty() {
    last = null;
    first = null;
    size = 0;
}
```

객체와 레퍼런스의 연결이 끊어지므로 힙(Heap)에 남은 StorageNode 객체들은 GC 에 의하여 처리된다.

Section 5. Exception Handling

성적을 저장하는 파일의 부분들을 다음과 같이 정의하자.

3,Algebra,Calculus,Geometry	(header)
Student1,72,79,100 Student2,85,91,72 Student3,90,90,90	(details)

scores.txt

헤더(header) 또는 디테일(detail)을 읽는 과정에서 문제가 발생할 수 있으며, 경우는 다음과 같다.

<table><tr><td>4,Algebra,Calculus,Geometry</td></tr><tr><td>Student1,72,79,100 Student2,85,91,72 Student3,90,90,90</td></tr></table> <p>(not enough header arguments)</p>	4,Algebra,Calculus,Geometry	Student1,72,79,100 Student2,85,91,72 Student3,90,90,90	<table><tr><td>Algebra,Calculus,Geometry</td></tr><tr><td>Student1,72,79,100 Student2,85,91,72 Student3,90,90,90</td></tr></table> <p>(invalid header argument)</p>	Algebra,Calculus,Geometry	Student1,72,79,100 Student2,85,91,72 Student3,90,90,90
4,Algebra,Calculus,Geometry					
Student1,72,79,100 Student2,85,91,72 Student3,90,90,90					
Algebra,Calculus,Geometry					
Student1,72,79,100 Student2,85,91,72 Student3,90,90,90					
<table><tr><td>3,Algebra,Calculus,Geometry</td></tr><tr><td>Student1,72 Student2,85,91,72 Student3,90,90,90</td></tr></table> <p>(not enough arguments for 'Student1')</p>	3,Algebra,Calculus,Geometry	Student1,72 Student2,85,91,72 Student3,90,90,90	<table><tr><td>3,Algebra,Calculus,Geometry</td></tr><tr><td>Student1,72,79,hundred Student2,85,91,72 Student3,90,90,90</td></tr></table> <p>(invalid argument for 'Student1')</p>	3,Algebra,Calculus,Geometry	Student1,72,79,hundred Student2,85,91,72 Student3,90,90,90
3,Algebra,Calculus,Geometry					
Student1,72 Student2,85,91,72 Student3,90,90,90					
3,Algebra,Calculus,Geometry					
Student1,72,79,hundred Student2,85,91,72 Student3,90,90,90					
<table><tr><td></td></tr><tr><td></td></tr></table> <p>(blank file)</p>					

Possible Invalid Cases

헤더(header)를 파싱하는 과정에서 예외가 발생하면 모집단의 생성이 불가능하기 때문에 그러한 예외를 나타내기 위한 'InvalidHeaderException' 클래스가 필요하다. *(디테일(detail) 파싱 도중 발생한 예외는 다음 디테일(detail)로 넘어가면 끝이므로 모집단 생성에 영향을 주지는 않는다.*

'InvalidHeaderException' 클래스는 다음과 같다.

<InvalidHeaderException>
(extends Exception)

```
InvalidHeaderException(int expected)
InvalidHeaderException(String invalidArg)
```

헤더(header) 파싱 도중 발생할 수 있는 두 가지 예외에 대한 생성자가 존재하며, 다음과 같이 발생시킨다.

(헤더의 교과목의 개수의 파싱이 불가능한 경우)

```
try {subjectAmount = Integer.parseInt(headerArray[0]);}
catch (NumberFormatException e) {
    throw new InvalidHeaderException(headerArray[0]);
}
```

(헤더에 기입된 교과목의 수가 충분하지 않은 경우)

```
if (subjectAmount > headerArray.length - 1) {
    throw new InvalidHeaderException(subjectAmount);
}
```

(헤더가 존재하지 않는 경우)

```
if (header == null) {
    throw new InvalidHeaderException("");
}
```

Section 6. Data File

[scores.txt] (UTF-8 encoded)

```
6,Algebra,Calculus,Geometry,Statistics,Physics,Chemistry
Student1,72,12,78,46,10,100
Student2,73,20,92,71,28,92
Student3,82,30,78,60,1,92
Student4,92,12,61,57,1,93
Student5,93,49,29,47,28,82
Student6,75,28,94,54,25,82
Student7,94,43,41,59,21,85
Student8,76,23,92,61,20,100
Student9,91,16,91,47,16,93
Student10,92,15,66,45,27,94
Student11,71,20,25,44,13,94
Student12,81,38,93,46,2,93
Student13,95,50,63,61,3,97
Student14,92,42,64,72,27,86
Student15,74,50,40,66,15,85
Student16,71,36,51,59,22,92
Student17,73,30,65,51,27,91
Student18,92,50,77,73,22,86
Student19,83,35,80,56,11,89
Student20,74,26,68,59,6,100
Student21,83,46,69,66,23,89
Student22,87,12,79,60,22,89
Student23,95,15,82,41,4,96
Student24,75,11,23,65,17,97
Student25,75,44,99,50,12,98
Student26,87,27,95,46,19,81
Student27,77,48,35,73,10,92
Student28,82,33,73,57,28,89
Student29,84,20,71,78,29,96
Student30,71,44,77,53,10,94
Student31,79,20,54,49,30,83
Student32,94,32,98,46,27,87
```

Section 7. Program Execution

(Execution using the above file)

Statistics of 32 student(s)						
	Algebra	Calculus	Geometry	Statistics	Physics	Chemistry
Student1	72	12	78	46	10	100
Student2	73	20	92	71	28	92
Student3	82	30	78	60	1	92
Student4	92	12	61	57	1	93
Student5	93	49	29	47	28	82
Student6	75	28	94	54	25	82
Student7	94	43	41	59	21	85
Student8	76	23	92	61	20	100
Student9	91	16	91	47	16	93
Student10	92	15	66	45	27	94
Student11	71	20	25	44	13	94
Student12	81	38	93	46	2	93
Student13	95	50	63	61	3	97
Student14	92	42	64	72	27	86
Student15	74	50	40	66	15	85
Student16	71	36	51	59	22	92
Student17	73	30	65	51	27	91
Student18	92	50	77	73	22	86
Student19	83	35	80	56	11	89
Student20	74	26	68	59	6	100
Student21	83	46	69	66	23	89
Student22	87	12	79	60	22	89
Student23	95	15	82	41	4	96
Student24	75	11	23	65	17	97
Student25	75	44	99	50	12	98
Student26	87	27	95	46	19	81
Student27	77	48	35	73	10	92
Student28	82	33	73	57	28	89
Student29	84	20	71	78	29	96
Student30	71	44	77	53	10	94
Student31	79	20	54	49	30	83
Student32	94	32	98	46	27	87
E(X)	82.3438	30.5313	68.8438	56.8125	17.3750	91.1563
V(X)	70.3506	169.7490	468.5068	96.5273	85.2344	29.2568
$\sigma(X)$	8.3875	13.0288	21.6450	9.8248	9.2322	5.4090

Section 7.1. Execution with Invalid Files

(Invalid file 1 - insufficient header arguments)

```
4,Algebra,Calculus,Geometry
Student1,72,12,78
Student2,73,20,92
Student3,82,30,78
```

(Execution - 1)

```
Exception occurred while constructing 'Population' object
InvalidHeaderException: Found less arguments than the expected 4
```

(Invalid file 2 - invalid header argument)

```
Three,Algebra,Calculus,Geometry
Student1,72,12,78
Student2,73,20,92
Student3,82,30,78
```

(Execution - 2)

Exception occurred while constructing 'Population' object
InvalidHeaderException: 'Three' is not a valid argument to define the number of subjects

(Invalid file 3 - invalid detail arguments)

```
3,Algebra,Calculus,Geometry
Student1,72,12
Student2,73,20,Ninety-Two
Student3,82,30,78
```

(Execution - 3)

'Student1' contains less arguments than the expected 3
'Ninety-Two' is not a valid argument to define the score of 'Student2'
Statistics of 1 student(s)

	Algebra	Calculus	Geometry
Student3	82	30	78
E(X)	82.0000	30.0000	78.0000
V(X)	0.0000	0.0000	0.0000
$\sigma(X)$	0.0000	0.0000	0.0000

(Invalid file 4 - blank file)

(Execution - 4)

Exception occurred while constructing 'Population' object
InvalidHeaderException: '' is not a valid argument to define the number of subjects

(Invalid file 5 - different file format)

```
Christmas Gifts
1. new laptop
2. new mechanical keyboard
3. new scientific calculator
```

(Execution - 5)

Exception occurred while constructing 'Population' object
InvalidHeaderException: 'Christmas Gifts' is not a valid argument to define the number of subjects