

2024년 1학기 자바프로그래밍 중간시험(30점)

제출파일: midexam1.zip, midexam2.zip

- 기능 구현과 상관없이 프로그램 동작 중 예외가 발생하는 경우, 각 항목별 -2점 감점함
- 각 기능별 부분 점수는 없음 (기능에 대한 코드만 있고 동작이 안되면 점수 없음)
- 그 외 과제의 감점 요인과 동일: 컴파일 에러, 주석(학번, 이름), UTF-8 등
- 자바 소스파일(클래스이름.java)만 압축해서 제출하세요. (클래스이름.class 파일 제출 시 0점 처리함)

1. 주어진 문자열 검색 및 교체, slicing 기능 구현 (15점)

제출파일: midexam1.zip

■ 구현 파일: StringUtil.java, StringUtilApp.java

■ StringUtil 클래스 (9점)

- ✓ **public int getWordCount(String text, String searchWord)** 메소드 구현 (3점)
 - main()함수에 제공된 text 문자열을 파라미터로 전달하고, text 문자열에서 searchWord가 없는 경우, “검색어 is not found!” 출력
 - 해당 검색어가 text 문자열에 존재하면 검색된 모든 시작 index를 출력
 - 전체 검색된 개수를 리턴
- ✓ **public String replaceWord(String text, String searchWord, String replacement)** 메소드 구현 (3점)
 - text 문자열에서 searchWord를 찾아서 replacement 문자열로 변경
 - searchWord가 text 문자열 내부에 없는 경우, “검색어 is not found!”를 출력하고 빈 문자열을 리턴
 - searchWord가 text 문자열 내부에 존재하는 경우, 모든 searchWord를 replacement로 교체한 다음, 교체된 문자열을 리턴
- ✓ **public String sliceString(String text, int start, int end)** 메소드 구현 (3점)
 - start에서 end-1 인덱스까지의 문자열을 잘라서 리턴
 - 정상적인 인덱스만 전달됨(비정상적인 인덱스 체크는 StringUtilApp에서 확인)
 - **String.substring(start, end)**을 사용해서 구현한 경우, 0점 처리함

■ StringUtilApp 클래스 (6점)

- ✓ 메뉴 구현 및 0 번 입력 시 프로그램 종료 (3점)
 - 메뉴 선택 시 필요한 문자열을 화면상에서 입력받고 StringUtil 객체에 전달
 - StringUtil 메소드에서 리턴한 내용 화면 출력
- ✓ 메뉴 3번(Slice a String) 인덱스 검사 및 반환 문자열 출력 기능 구현 (3점)
 - Start 인덱스가 end 인덱스 큰 경우, 인덱스가 문자열의 범위를 벗는 경우(음수 인덱스, 문자열 길이보다 큰 경우), Start 인덱스와 end 인덱스가 동일한 경우(음수, 문자열 길이 보다 큰 경우)는 화면상에서 start, end 인덱스를 다시 입

력 받음

- StringUtil.sliceString()의 반환값 출력

■ String text는 그대로 복사해서 사용하세요.

```
String text =
    "Java was originally developed by James Gosling at Sun Microsystems." +
    "It was released in May 1995 as a core component of Sun's Java platform." +
    "Sun had relicensed most of its Java technologies under the GPL-2.0-only license.";
```

■ 문자열 인덱스

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]
J	a	v	a		w	a	s		o	r	i	g	i	n	a	l	l	y	
[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[31]	[32]	[33]	[34]	[35]	[36]	[37]	[38]	[39]
d	e	v	e	l	o	p	e	d		b	y		J	a	m	e	s		G
[40]	[41]	[42]	[43]	[44]	[45]	[46]	[47]	[48]	[49]	[50]	[51]	[52]	[53]	[54]	[55]	[56]	[57]	[58]	[59]
o	s	l	i	n	g		a	t		S	u	n		M	i	c	r	o	s
[60]	[61]	[62]	[63]	[64]	[65]	[66]	[67]	[68]	[69]	[70]	[71]	[72]	[73]	[74]	[75]	[76]	[77]	[78]	[79]
y	s	t	e	m	s	.	I	t		w	a	s		r	e	l	e	a	s
[80]	[81]	[82]	[83]	[84]	[85]	[86]	[87]	[88]	[89]	[90]	[91]	[92]	[93]	[94]	[95]	[96]	[97]	[98]	[99]
e	d		i	n		M	a	y		l	9	9	5		a	s		a	
[100]	[101]	[102]	[103]	[104]	[105]	[106]	[107]	[108]	[109]	[110]	[111]	[112]	[113]	[114]	[115]	[116]	[117]	[118]	[119]
c	o	r	e		c	o	m	p	o	n	e	n	t		o	f		S	u
[120]	[121]	[122]	[123]	[124]	[125]	[126]	[127]	[128]	[129]	[130]	[131]	[132]	[133]	[134]	[135]	[136]	[137]	[138]	[139]
n	'	s		J	a	v	a		p	l	a	t	f	o	r	m	.	S	u
[140]	[141]	[142]	[143]	[144]	[145]	[146]	[147]	[148]	[149]	[150]	[151]	[152]	[153]	[154]	[155]	[156]	[157]	[158]	[159]
n		h	a	d		r	e	l	i	c	e	n	s	e	d		m	o	s
[160]	[161]	[162]	[163]	[164]	[165]	[166]	[167]	[168]	[169]	[170]	[171]	[172]	[173]	[174]	[175]	[176]	[177]	[178]	[179]
t		o	f		i	t	s		J	a	v	a		t	e	c	h	n	o
[180]	[181]	[182]	[183]	[184]	[185]	[186]	[187]	[188]	[189]	[190]	[191]	[192]	[193]	[194]	[195]	[196]	[197]	[198]	[199]
l	o	g	i	e	s		u	n	d	e	r		t	h	e		G	P	L
[200]	[201]	[202]	[203]	[204]	[205]	[206]	[207]	[208]	[209]	[210]	[211]	[212]	[213]	[214]	[215]	[216]	[217]	[218]	[219]
-	2	.	0	-	o	n	l	y		l	i	c	e	n	s	e	.		

■ 실행 결과

```
-----
1. Search a word
2. Replace a word
3. Slice a String
0. Quit
-----
Choose a number: 1
Type a search word: AAA
AAA is not found!

AAA found: 0 times
-----
1. Search a word
2. Replace a word
```

```

3. Slice a String
0. Quit
-----
Choose a number: 1
Type a search word: Java
index: [0, 124, 169, ]

Java found: 3 times
-----
1. Search a word
2. Replace a word
3. Slice a String
0. Quit
-----
Choose a number: 1
Type a search word: a
index: [1, 3, 6, 15, 34, 47, 71, 78, 87, 95, 98, 125, 127, 131, 143, 170, 172, ]

a found: 17 times
-----
1. Search a word
2. Replace a word
3. Slice a String
0. Quit
-----
Choose a number: 2
Type two words(seachword replacement): Python Java
Python is not found!
-----
1. Search a word
2. Replace a word
3. Slice a String
0. Quit
-----
Choose a number: 2
Type two words(seachword replacement): Java C#
C# was originally developed by James Gosling at Sun Microsystems.It was released in May
1995 as a core component of Sun's C# platform.Sun had relicensed most of its C#
technologies under the GPL-2.0-only license.
-----
1. Search a word
2. Replace a word
3. Slice a String
0. Quit
-----
Choose a number: 3
Type a range(start ~ end): 10 1

```

start: 10, end: 1, len: 218

Index is out of range!

-
1. Search a word
 2. Replace a word
 3. Slice a String
 0. Quit
-

Choose a number: 3

Type a range(start ~ end): 200 219

start: 200, end: 219, len: 218

Index is out of range!

-
1. Search a word
 2. Replace a word
 3. Slice a String
 0. Quit
-

Choose a number: 3

Type a range(start ~ end): 10 20

start: 10, end: 20, len: 218

Sliced String: riginally

-
1. Search a word
 2. Replace a word
 3. Slice a String
 0. Quit
-

Choose a number: 3

Type a range(start ~ end): 9 29

start: 9, end: 29, len: 218

Sliced String: originally developed

-
1. Search a word
 2. Replace a word
 3. Slice a String
 0. Quit
-

Choose a number: 0



Quit Program

2. 주사위 게임 프로그램 (15점)

제출파일: midexam2.zip

2명의 선수(Alice, Bob)가 2개의 주사위(1~6)를 던져서 나온 숫자의 합 만큼 이동하는 게임을 구현하시오.

- 클래스 설계 및 구현 기능 (클래스 이름 및 클래스의 숫자에 제한 없음)
 - ✓ 두 선수의 객체 생성 (생성자에 이름 전달: Alice, Bob)
 - ✓ 경기 규칙
 - 주사위의 숫자는 1~6사이의 값을 랜덤 생성
 - 한 선수의 점수가 50점 이상이면 게임은 바로 종료 (2점)
 - 두 주사위의 숫자가 동일하고 (1, 1), (3, 3), (5, 5)인 경우, 두 주사위 숫자의 합을 점수에 더하고 해당 Player는 주사위를 한 번 더 던짐: "Roll Again" 출력 (5점)
 - 두 주사위의 숫자가 동일하고 (2, 2), (4, 4), (6, 6)인 경우, 두 주사위 숫자의 합만큼 점수에서 감점함: "OOPS" 출력 (5점)
 - 감점된 점수가 0보다 작은 경우에는 0부터 시작함
 - 각 선수가 던진 주사위의 값 및 누적된 합을 화면에 출력 (2점)
 - 게임이 끝난 다음 승패 결과 출력 (1점)

Player1 (Alice)	Player2 (Bob)
	

실행 결과 #1

Rolling Dice Game Start	
[Alice] (2, 4)= 6	[Bob] (5, 2)= 7
[Alice] (4, 3)= 13	[Bob] (1, 4)= 12
[Alice] (4, 5)= 22	[Bob] (2, 4)= 18
[Alice] (1, 2)= 25	[Bob] (5, 4)= 27
[Alice] (4, 6)= 35	[Bob] (1, 4)= 32

```
[Alice] (5, 5)= 45 Roll Again!  
[Alice] (6, 3)= 54  
-----  
Alice: 54 / Bob: 32: --> Alice Win  
-----
```

실행 결과 #2

```
-----  
Rolling Dice Game Start  
-----  
[Alice] (3, 2)= 5  
[Bob] (3, 5)= 8  
[Alice] (4, 2)= 11  
[Bob] (5, 5)= 18 Roll Again!  
[Bob] (6, 3)= 27  
[Alice] (4, 3)= 18  
[Bob] (3, 6)= 36  
[Alice] (4, 4)= 10 OOPS!  
[Bob] (3, 4)= 43  
[Alice] (4, 1)= 15  
[Bob] (6, 6)= 31 OOPS!  
[Alice] (6, 2)= 23  
[Bob] (4, 2)= 37  
[Alice] (5, 4)= 32  
[Bob] (4, 6)= 47  
[Alice] (4, 4)= 24 OOPS!  
[Bob] (1, 1)= 49 Roll Again!  
[Bob] (1, 6)= 56  
-----  
Alice: 24 / Bob: 56: --> Bob Win  
-----
```

실행 결과 #3

```
-----  
Rolling Dice Game Start  
-----  
[Alice] (1, 2)= 3  
[Bob] (6, 5)= 11  
[Alice] (2, 1)= 6  
[Bob] (3, 1)= 15  
[Alice] (2, 2)= 2 OOPS!  
[Bob] (5, 3)= 23  
[Alice] (5, 4)= 11  
[Bob] (5, 5)= 33 Roll Again!  
[Bob] (6, 5)= 44  
[Alice] (6, 6)= 0 OOPS!  
[Bob] (1, 1)= 46 Roll Again!
```

```
-----
[Bob] (4, 5)= 55
-----
Alice: 0 / Bob: 55: --> Bob Win
-----
```

실행 결과 #4

```
-----
Rolling Dice Game Start
-----
[Alice] (5, 5)= 10 Roll Again!
[Alice] (5, 5)= 20 Roll Again!
[Alice] (5, 5)= 30 Roll Again!
[Alice] (5, 5)= 40 Roll Again!
[Alice] (5, 5)= 50 Roll Again!
-----
Alice: 50 / Bob: 0: --> Alice Win
-----
```