

Java 2級 解答

©商業科のJava教育を考える会

第1章 プログラミングの基礎

- | | |
|------------------------------|-------------------------------------|
| 1- 1 System.out.print | 2- 4 (1) if |
| 別解 System.out.println | (2) else if |
| | (3) else |
| 1- 2 System.out.println | 2- 5 (1) Scanner |
| 1- 3 Hello¥nJava World! | (2) number >= 100 |
| 1- 4 "得点は" + tokuten + "点です" | (3) number >= 80 |
| 1- 5 (a + b) | 2- 6 x == y && y == z |
| 1- 6 (int) avg1 | 2- 7 japanese >=80 english >= 80 |
| 1- 7 (1) String | 2- 8 (1) kubun < 1 kubun > 4 |
| (2) final | (2) switch(kubun) |
| 1- 8 (1) tanka | (3) break |
| (2) printf | (4) default |
| 1- 9 (1) Scanner | 2- 9 int i = 1; i < 11; i++ |
| (2) sc.next() | 2-10 i++ |
| (3) sc.nextInt() | 2-11 for(int j = 1; j < 10; j++) |
| 1-10 (1) double PI | 2-12 (1) startNum |
| (2) Scanner sc | (2) i++ |
| (3) ensyu | (3) sum |

第2章 分岐と繰り返し処理

- | | |
|---------------------------|---------------------|
| 2- 1 tokuten < kijyun | 2-13 (1) number |
| | (2) answer > number |
| | (3) answer < number |
| 2- 2 tokuten == saikou | 2-14 continue |
| 2- 3 (1) String | 2-15 break |
| (2) answer = sc.nextInt() | |
| (3) else | |

第3章 クラスの基礎

- 3- 1 (1) int grade
 - (2) String name
 - (3) int age※(1)(2)(3)順不同
- 3- 2 (1) new Student()
 - (2) tanaka.name
- 3- 3 (1) Student
 - (2) tanaka.showName()
 - (3) yamada.showName()
- 3- 4 (1) String name, int price
 - (2) this
- 3- 5 (1) return
 - (2) pen.showItem()
 - (3) pen.showSales()
- 3- 6 new SalesTest1()
- 3- 7 SalesTest2
- 3- 8 (1) setName("山本")
 - (2) score = 90※(1)(2)順不同
- 3- 9 (1) this.level = level
 - (2) player
 - (3) player.name
 - (4) enemy.name
- 3-10 (1) String name, int price
 - (2) Item
 - (3) name, price
 - (4) getItem

第4章 配列の利用

- 4- 1 (1) new String[3]
 - (2) name[0]
 - (3) name[1]
 - (4) name[2]
- 4- 2 (1) new int[3]
 - (2) i < 3
 - (3) score[i]
 - (4) average
- 4- 3 (1) {380, 240, 520}
 - (2) code[i]
 - (3) price[i]
- 4- 4 (1) sum = 0
 - (2) length
 - (3) data[i]
- 4- 5 (1) Student[]
 - (2) class1[i]
- 4- 6 (1) Card(name, power)
 - (2) cardNumber++
 - (3) player.myCardShow()

第5章 データの集計と平均

- 5- 1 (1) int goukei = 0
 - (2) goukei + uriage
- 5- 2 (1) goukei = goukei + uriage
 - (2) kensu = kensu + 1※(1)(2)順不同
- 5- 3 (1) uriage == 999999
 - (2) uriage != 999999

5- 4 (1) tanka * suu * 0.9
(2) goukei += kingaku
5- 5 (1) true
(2) ninkei = ninkei + nin
(3) syokuhikei + syokuhi

5- 6 (1) code == 1
(2) ji * 100
(3) ji * 60

5- 7 (1) ninzu++
(2) goukaku++
(3) goukakuritu

5- 8 (1) ninzu++
(2) gouhi.input()
(3) gouhi.output2()

第 6 章 最大値と最小値の算出

6- 1 (1) int max = 0
(2) max = score

6- 2 (1) score > max
(2) min = score

6- 3 (1) score > max
(2) maxNamae = namae

6- 4 (1) score >= max
(2) max = score
(3) maxNamae = namae

※(2)(3)順不同

6- 5 (1) double kiroku
(2) saisoku > kiroku
(3) saisoku = kiroku

6- 6 (1) uriMax = urisuu
(2) uriMin = urisuu
(3) maxMin.hantei()
(4) maxMin.output()

第 7 章 配列を用いた集計

7- 1 (1) int inputNo
(2) inputNo - 1
(3) option[i]

7- 2 (1) code
(2) sales
(3) code
(4) sum / count

7- 3 (1) code
(2) sum += inputSales
(3) percent

7- 4 (1) sitenCode * 10000
(2) sitenSales[sitenCode - 1]
(3) code = sc.nextInt()

7- 5 (1) kinsyu[i]
(2) number[i]

7- 6 (1) nendai = 4
(2) nendai = nendai - 2
(3) syuukei[i]

第8章 線形探索

- 8- 1 (1) `code[i] != inputCode`
(2) `price[i]`
- 8- 2 (1) `code[5] = inputCode`
(2) `i++`
- 8- 3 (1) `int inputWeight`
(2) `weight[i] < inputWeight`
- 8- 4 (1) `int i = 0`
(2) `i++`
(3) `uriagesuu[i]++`
- 8- 5 (1) `sm.inputCode != 999`
(2) `sm.uriagesuu[sm.i]++`
- 8- 6 (1) `i++`
(2) `uriage[i] += inputUriage`
(3) `uriage[i] > max`
(4) `maxMei = hinmei[i]`

第9章 ファイル入出力

- 9- 1 `moji`
- 9- 2 (1) `fr`
(2) `readLine()`
- 9- 3 (1) `br`
(2) `line`
(3) `name`
(4) `score`
- 9- 4 `fw`
- 9- 5 (1) `syouhin2`
(2) `syouhin3.length()`
※別解「4」