**Mobile Software Programming Homework #2**

1. (30점) Java의 상속 및 다형성을 사용하여 게임을 제작하려 한다. 이 때 게임의 캐릭터는 다음과 같이 구성되며 공격 시 상대방에게 입힐 수 있는 데미지는 다음과 같이 계산된다. 게임이 시작되고 인스턴스가 형성되면 모든 캐릭터의 체력량은 최대 체력값을 가지도록 하며, 공격을 받으면 체력량을 상대방이 주는 Damage만큼 소모하게 된다. 또한 주어진 체력을 다 소모할 경우에 캐릭터는 게임에서 패배하는 것으로 간주한다.

|  |  |  |
| --- | --- | --- |
| 게임 캐릭터 | Damage | 최대 체력 |
| Human | 50 | 200 |
| Cyber Daemon | 40, 30% 확률로 60 | 250 |
| Barlog | 25씩 두 번 공격. 각 공격마다 20% 확률로 35 | 170 |
| Elf | 35, 30%의 확률로 70 | 180 |

또한 물약을 먹으면 체력이 회복되는 데 사용할 수 있는 물약의 수는 모두 5개로 제한되며 회복량은 다음과 같다. 또한 회복후의 체력은 물약의 투입 횟수에 관계없이 최대 체력 수치를 초과할 수 없다.

|  |  |
| --- | --- |
| 게임 캐릭터 | Damage |
| Human | 40 |
| Cyber Daemon | 30, 30% 확률로 50 회복 |
| Barlog | 30, 하지만 한 번에 물약을 두 병을 먹을 수 있다. |
| Elf | 30, 25%의 확률로 60 회복 |

이들 캐릭터는 Character 클래스를 상속하며 Character 클래스는 다음과 같은 메소드를 가진다. 이 때 Character 클래스는 추상클래스로 구현하도록 하며 Character 클래스를 상속하는 클래스의 메소드 일부는 오버라이드하며 일부 메소드는 상속하도록 한다.

* int hitPoint; // 캐릭터의 체력.
* int portionNumber; // 가지고 있는 물약의 수를 나타낸다.
* void initialize(); // 체력과 물약의 수를 최대치로 초기화한다.
* void attack(Character c); // 캐릭터 c에게 데미지를 준다. 데미지의 숫자는 주어진 특성에 맞도록 주며 확률이 필요한 경우 랜덤함수를 사용하여 확률별로 데미지를 주도록 한다.
* int getDamage(int damage); // 특정 숫자의 데미지량을 받아 자신의 체력에 반영하며 필요하다면 recover() 메소드를 통해 물약을 먹고 체력을 회복하도록 한다.
* void recover(); // 물약을 먹고 체력을 캐릭터 특성에 따라 회복한다. Barlog의 경우 필요하다면 한번에 2번 물약을 먹을 수 있다.
* boolean isDie(); // 자신이 죽었는 지 아닌지를 판단한다. 죽은 경우 “<\_\_\_> is dead.” 메시지를 출력한다.
* boolean needPortion(); // 지금 물약을 먹어야 하는 지를 판단하고 남은 물약이 있다면 물약을 소비한다.
* void whoAmI(); // 자신이 어떤 클래스에 속하는 지를 화면에 출력한다.

여기서 main() 메소드를 담고 있어야 하는 클래스가 필요한 데 이 클래스를 Game이라고 하고 격투를 구현하는 메소드를 다음과 같이 구현한다.

Character battle(Character blue, Character red); // 두 character 인스턴스를 매개변수로 받아 이 중 승자 인스턴스를 리턴한다. Battle 방식은 두 character 인스턴스가 번갈아 가며 공격을 하며 한 인스턴스의 체력이 0이하가 되면 Battle을 멈추고 체력이 높은 character를 리턴한다. 이 때 blue가 먼저 선공을 하는 것으로 하며 어떤 character를 blue에 놓을 것인가 하는 문제는 프로그램상에서 임의로 선택하도록 한다. 또한 blue, red는 모두 전투 실행 전의 체력과 물약은 최대치를 가지고 있는 것으로 한다. 단 battle시에는 다음과 같은 메시지를 화면에 계속 출력하여야 한다.

…

Human attacks with damage 50. Elf has 70 hit point now.

Elf attacks with damage 35. Human has 55 hit point now.

Human eats portion and 95 hit point now.

…

void tournament(Character hu, Character cyber, Character bar, Character el); // 4개의 Character의 인스턴스(각 인스턴스들은 서로 다른 캐릭터 타입이어야 함)들을 매개변수로 받으며 이들을 토너먼트로 경기를 시켜 우승자의 whoAmI() 메소드를 통해 우승자의 메시지를 출력하도록 한다. 토너먼트로 경기를 한다는 의미는 1:1로 2경기를 하고 그 승자간에 결승전을 가지도록 하여 결승전 승자가 우승을 한다는 것을 말한다.

1. (30점) 입력 값들의 합과 평균을 구하는 generic 클래스를 작성하시오. 합과 평균을 구하는 클래스는 합을 위한 sum 메소드와 평균을 구하는 average 메소드를 포함해야 한다. 이 메소드는 모두 generic으로 구성되어야 한다. 입력은 20개 크기의 배열로 주어지며 정수와 실수에 대하여 합이 가능해야 한다.

합과 평균을 구하는 데이터는 정수형, 실수형의 2가지 배열로 구성하며 배열의 내용은 다음과 같이 구성한다. 배열의 내용은 다음과 같이 소스코드 내의 메인 함수 안에서 주도록 한다.

출력은 배열의 합과 평균을 출력한다.

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}

{1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0}

1. (20 점) 숙제 #1에서 구현한 Person, Employee, Student 클래스를 활용하여 생성된 모든 인스턴스들을 vector()에 저장하고 인스턴스들의 이름을 loop를 통해서 출력하는 루틴을 구현하라. 각 클래스들의 인스턴스들은 모두 지난 숙제의 예를 그대로 활용하도록 한다. 이 때 vector는 vector<Person>()으로 선언하도록 한다.
2. (20 점) 다음과 같이 배열이 주어질 때, C[i] = A[i] \* B[i]를 계산하는 루틴을 작성하시오. 단 thread를 사용하여 각 연산이 독립적으로 수행되도록 해야 한다. 여기서 배열의 크기는 10이므로 10개의 thread를 발생시켜 각 곱셈이 수행되도록 한다.

int[] A= {3, 2, 5, 7, 8, 9, 10, 2, 7, 8};

int[] B= {4, 3, 7, 3, 2, 6, 1, 0, 1, -2};