

```
int main() {
                                                33
                                                             cout << "게임을 시작합니다.₩n₩n";
                                                34
                                                             SaveLoadManager saveLoadManager;
main.cpp
                                                35
                                                             Character *character = new Character:
                                                36
                                                             Dungeon dungeon;
                                                37
                                                             try {
                                                                 cout << "저장된 게임을 불러오시겠습니까?₩n"
                                                38
Implementation of main.cpp
                                                                     << "1. 여\\n"
                                                39
                                                                      << "2. 아니오\n"
                                                40
                                                                     << "선택: ";
                                                41
                                                                 int choice; cin >> choice;
                                                                 cout << '\n';

∨ Character* createCharacter() {
 12
                                                                 switch(choice) {
                                                45
            cout << "캐릭터 이름을 입력하세요: ";
 13
                                                46
                                                                    case 1:
            string name; cin >> name;
 14
                                                                        saveLoadManager.loadGame(character, dungeon);
 15
            cout << '\n';
                                                                        cout << '\n';
                                                48
 16
            cout << "캐릭터 타입을 선택하세요:₩n"
                                                                        break:
 17
                << "1. 전사₩n"
 18
                                                                    default:
                << "2. 마법사\m"
 19
                                                                        cout << "새로운 캐릭터를 생성합니다.\\";
                                                51
                << "선택: ";
 20
                                                                        character = createCharacter();
            int choice; cin >> choice;
 21
                                                                        break:
                                                53
            cout << '₩n';
 22
                                                54
 23
                                                55
                                                             } catch(const exception&) {
            switch(choice) {
 24
 25
                case 1:
                                                56
                                                                 cout << "저장된 게임이 없습니다. 새로운 캐릭터를 생성합니다
                   return new Warrior(name);
 26
                                                57
                                                                 character = createCharacter();
 27
                default:
                                                58
                   return new Wizard(name);
 28
                                                             dungeon.enter(character);
                                                59
 29
                                                60
 30
```

## main.cpp

Program Output from main.cpp

ⓒ C:₩Users₩rlatj₩source₩repos × + ✓ 게 임을 시작합니다.

저장된 게임을 불러오시겠습니까?

서상된 게임을 물러오시겠습니까? 1. 예

2. 아니오 선택: 1

캐릭터 불러오기 완료! 던전 불러오기 완료! 게임 불러오기 완료! C:\Users\Union\Uni

게임을 시작합니다.

저장된 게임을 불러오시겠습니까? 1. 예

2. 아니오 선택: 2

새로운 캐릭터를 생성합니다. 캐릭터 이름을 입력하세요: aaa

캐릭터 타입을 선택하세요:

1. 전사 2. 마법사 선택: 1



## SaveLoadManager

14 15

16

17

20

26

29

30

31

33

```
☞ C:₩Users₩rlatj₩source₩repos × +
게임을 시작합니다.
```

저장된 게임을 불러오시겠습니까? 1. 예

```
2. 아니오
선택: 1
```

```
캐릭터 불러오기 완료!
던전 불러오기 완료!
게임 불러오기 완료!
```

```
character->saveCharacter();
      dungeon.saveDungeon();
      cout << "게임 저장 완료!\n";
void SaveLoadManager::loadGame(Character*& character, Dungeon& dungeon) {
      trv {
          ifstream ifs("character.txt");
          string type; ifs >> type;
          if(type.compare("[전사]")==0) {
              character = new Warrior();
          } else if(type.compare("[마법사]")==0) {
              character = new Wizard();
          character->loadCharacter();
          dungeon.loadDungeon();
      } catch(const exception&) {
          throw runtime_error("저장된 게임이 없습니다.");
      cout << "게임 불러오기 완료!\n";
```

void SaveLoadManager::saveGame(Character\*& character. Dungeon& dungeon) {



```
Dungeon
                                                          ∨ void Dungeon::saveDungeon() {
                                                   110
                                                   111
                                                                trv {
                                                                    ofstream ofs("monster.txt");
                                                   112
Implementation of Dungeon Class
                                                                    ofs << monsters.size() << '\m';
                                                   113
                                                   114
                                                                    for(Monster* monster: monsters) {
                                                                        monster->saveMonster(ofs);
                                                   115
                                                   116
                                                                  catch(const exception&) {
                                                   117
                                                                    throw runtime_error("던전 저장 실패");
                                                   118
                                                   119
                                                   120
                                                   121

    void Dungeon∷loadDungeon() {
                                                   122
                                                                try {
                                                   123
                                                                    ifstream ifs("monster.txt");
                                                   124
                                                   125
                                                                    int cnt; ifs >> cnt;
                                                                    monsters.clear();
                                                   126
                                                   127
                                                                    while(cnt--) {
                                                   128
                                                                        monsters.push_back(new Monster);
                                                   129
                                                                        monsters.back()->loadMonster(ifs);
                                                   130
                                                   131
                                                   132
                                                                    cout << "던전 불러오기 완료!₩n";
                                                   133
                                                                 } catch(const exception&) {
                                                                    throw runtime_error("저장된 던전이 없습니다.");
                                                   134
                                                   135
                                                   136
```

# Dungeon

Program Output of Dungeon Class 공격력: 6

----- [전사] aaa -----

레벨: 1

경험치: 0 / 50

[던전에 입장중입니다.

체력: 120 / 120

골드: 0 현재 남은 몬스터 수: 13

0. 세이브 1. 상점에 가기

2. [몬스터] 고블린 3. [몬스터] 슬라임

4. [몬스터] 오리 5. [몬스터] 박쥐 6. [몬스터] 뱀

7. [몬스터] 전갈

8. [몬스터] 오크 9. [몬스터] 오크

10. [몬스터] 슬라임

11. [몬스터] 박쥐 12. [몬스터] 슬라임

13. [몬스터] 임프 14. [몬스터] 뱀

선택 :

© C:₩Users₩rlatj₩source₩repos ×

선택 : 6

[전사] aaa의 공격!

[몬스터] 뱀(은/는) 공격을 받았다. [몬스터] 뱀의 체력이 11로 감소했다.

[몬스터] 뱀의 공격!

[전사] aaa(은/는) 공격을 받았다. [전사] aaa의 체력이 110로 감소했다.

[전사] aaa의 공격! [몬스터] 뱀(은/는) 공격을 받았다. [몬스터] 뱀의 체력이 5로 감소했다.

[몬스터] 뱀의 공격! [전사] aaa(은/는) 공격을 받았다.

[전사] aaa의 체력이 100로 감소했다.

[전사] aaa의 공격! [몬스터] 뱀(은/는) 공격을 받았다. [몬스터] 뱀의 체력이 0로 감소했다.

[몬스터] 뱀(은/는) 사망했다.

승리! 경험치24, 골드 5 획득!



# Dungeon

Program Output of Dungeon Class

```
----- [전사] aaa -----
레벨: 1
경험치: 24 / 50
체력: 100 / 120
공격력: 6
```

방어력: 1 골드: 5

현재 남은 몬스터 수: 12 0. 세이브

1. 상점에 가기 2. [몬스터] 고블린

3. [몬스터] 슬라임 4. [몬스터] 오리

5. [몬스터] 박쥐 6. [몬스터] 전갈

7. [몬스터] 오크 8. [몬스터] 오크

8. [콘트더] 오크 9. [몬스터] 슬라임 10. [몬스터] 박쥐

10. [본교리] 국위 11. [몬스터] 슬라임

12. [몬스터] 임프 13. [몬스터] 뱀 선택 : 1

상점에 있는 아이템:

0. 취소

1. 회복 물약(10): 5골드 2. 회복 물약(20): 20골드

3. 회복 물약(50): 60골드 구매할 아이템 번호를 선택하세요: 1

[전사] aaa이(가) 회복 물약을 사용했습니다. 체력이 100에서 110 로 변화했습니다. ----- [전사] aaa -----

레벨: 1

경험치: 24 / 50 체력: 110 / 120

공격력: 6 방어력: 1 골드: 0

현재 남은 몬스터 수: 12 0. 세이브

1. 상점에 가기 2. [몬스터] 고블린

2. [몬스터] 슬라임

4. [몬스터] 오리 5. [몬스터] 박쥐

6. [몬스터] 전갈 7. [몬스터] 오크

8. [몬스터] 오크

9. [몬스터] 슬라임 10. [몬스터] 박쥐

11. [몬스터] 슬라임 12. [몬스터] 임프

13. [몬스터] 뱀 선택 : 0

게임 저장 완료!



# BattleManager

```
Implementation of BattleManager Class
```

```
class BattleManager {
  public:
    void startBattle(Character*& character, Monster*& monster);
};

void BattleManager::startBattle(Character*& character, Monster*& monster) {
  while(true) {
    if(!character->attack(monster)) return;
    cout << 'Wn';
    this_thread::sleep_for(chrono::seconds(1));
    if(!monster->attack(character)) return;
    cout << 'Wn';
    this_thread::sleep_for(chrono::seconds(1));
}</pre>
```



18

#### Character public: 27 28 \* 캐릭터 초기 설정 \* 레벨: 1 Implementation of Character Class \* 체력: 100 31 \* 최대 체력: 100 33 \* 공격력: 10 \* 경험치: 0 34 → class Character { 10 35 11 protected: Character(); 12 /\*\* 37 Character(string name); \* type: 캐릭터 타입 13 void killMonster(Character\*& target); \* name: 이름 14 void killMonster(Monster\*& target); 39 \* level: 레벨 15 virtual void levelUp(); 40 16 \* hp: 체력 41 bool damageLog() const; \* maxHp: 최대 체력 17 virtual bool takeDamage(int damage); \* atk: 공격력 18 bool attack(Character\*& target) const; 43 \* exp: 경험치 19 bool attack(Monster\*& target) const; \* plusHp: 레벨업 할때마다 오르는 체력 20 bool isAlive() const; \* plusAtk: 레벨업 할때마다 오르는 공격력 21 virtual void displayInfo() const; 22 \* gold: 소지 금액 47 virtual void saveCharacter(); 23 virtual void loadCharacter(); 48 24 string type; int getRandomInt(int min, int max); 49 25 string name; bool useMedicine(int hp. int gold); 26 int level, hp, maxHp, atk, exp, plusHp, plusAtk, gold; }; 51

#### Character

#### Implementation of Character Class

Character::Character() : Character("무명") {}

19

20

30

31

32

35

36

38

41

```
∨ Character::Character(string name) {
       this->name = name;
       level = 1:
       hp = maxHp = 100;
       atk = 10:
       exp = 0;
       gold = 0;
       plusHp = 10;
       plusAtk = 5;
       type = "[캐릭터] ";
void Character::killMonster(Character*& target) {
       exp += target->exp;
       gold += target->gold;
       cout << "승리! 경험치" << target->exp << ", 골드 " << target->exp << " 획득!\wn\wn\";
       while(exp \geq 50) {
          exp -= 50;
          levelUp();
```

44 45

46 47

48

49

51

52

53

54

55

56

57

58

60

61

62

63

64

65

67

68

69

70

72 73

```
∨ void Character::killMonster(Monster*& target) {
      exp += target->exp;
      gold += target->gold;
      cout << "Wn승리! 경험치" << target->exp << ", 골드 " << target->gold << " 획득!\wn\wn";
      while(exp >= 50) {
          exp -= 50;
          levelUp();
∨ void Character::levelUp() {
      cout << "레벨업: " << level << " -> " << level + 1 << '\m'
           << "최대 체력: " << maxHp << " -> " << maxHp + plusHp << '\m'
           << "체력: " << hp << " -> " << hp + plusHp << '\m'
           << "공격력: " << atk << " -> " << atk + plusAtk << '\m\m';
      level++;
      hp += plusHp;
      maxHp += plusHp;
      atk += plusAtk;

    bool Character::damageLog() const {
      cout << type << name << "(은/는) 공격을 받았다.₩n"
           << type << name << "의 체력이 " << hp << "로 감소했다.\\n";
      if(!isAlive()) {
          cout << type << name << "(은/는) 사망했다.₩n";
          return false:
      return true;
```

## Character

#### Implementation of Character Class

```
76
          hp = max(hp - damage, 0);
77
          return damageLog();
78
79
80
81
          cout << type << name << "의 공격!\m';
          return target->takeDamage(atk);
85
86
```

```
v bool Character::attack(Monster*& target) const {
            cout << type << name << "의 공격!\m';
87
            return target->takeDamage(atk);
90
```

```
    → bool Character::attack(Character*& target) const {

    → bool Character::isAlive() const {
91
             return hp > 0;
93
94
      void Character::displayInfo() const {
             cout << "---- " << type << name << " -----\#n"
96
                  << "레벨: " << level << "\m"
                  << "경험치: " << exp << " / 50\m"
                  << "체력: " << hp << " / " << maxHp << "\\m"
99
                  << "공격력: " << atk << "₩n"
101
                  << "골드: " << gold << "\m\m";
102
```

```
ofs << type
          << name << ' '
          << level << '
          << hp << '
          << maxHp << '
          << atk << ' '
          << exp << ' '
          << gold << ' ';
      ofs.close();
void Character::loadCharacter() {
      trv {
          ifstream ifs("character.txt");
          ifs >> type >> name >> level >> hp >> maxHp >> atk >> exp >> gold;
          type += ' ';
          if(ifs.fail()) {
              cout << "캐릭터 불러오기 실패!\"";
```

 void Character∷saveCharacter() + ofstream ofs("character.txt");

106

108 109

111

114

115 116

120

121

123

124

125 126

128

129

130

133

134 135

136

137

138

139

140

141

142

143

144

145

146

147

148 149

```
throw runtime error("파일 읽기 실패");
     cout << "캐릭터 불러오기 완료!\"";
  } catch(const exception&) {
     throw runtime error("저장된 게임이 없습니다."):

✓ int Character: getRandomInt(int min. int max) {
       return rand() % (max-min+1) + min:
```

```
if(this->gold < gold) {</pre>
        cout << "골드가 부족합니다.₩n₩n";
        return false:
     int nextHp = min(this->hp + hp. maxHp);
     cout << type << name << "이(가) 회복 물약을 사용했습니다.\\n"
         << "체력이 " << this->hp << "에서 " << nextHp << " 로 변화했습니다.\\n\\\\n";
```

this->gold -= gold;

this->hp = nextHp;

return true;

### Warrior

```
Implementation of Warrior Class
```

```
∨ class Warrior : public Character {
        protected
10
            /**
            ├* defense: 방어력
            * plusDefense: 레벨업 할 때마다 오르는 방어력
```

```
int defense, plusDefense;
```

public: 15 16 ├\* 전사 초기 설정

\*/

13

14

20

24

25

26

\* 레벨: 1 \* 체력: 120 ├\* 최대 체력: 120

\* 공격력: 6 \* 방어력: 1 ├\* 경험치: 0

Warrior(); Warrior(string name); bool takeDamage(int damage) override;;

void levelUp() override; void displayInfo() const override;

void saveCharacter() override; void loadCharacter() override; 32

```
∨ Warrior: Warrior(string name) {
       this->name = name;
       level = 1;
      hp = maxHp = 120;
      atk = 6;
      exp = 0:
      plusHp = 20;
      plusAtk = 3;
      plusDefense = 1;
      gold = 0:
      defense = 1;
```

16 17

20

21

28

29

30

31

33

34 35

37

38

40

41

44

45 46

47

Warrior::Warrior() : Warrior("무명") {}

type = "[전사] "; y bool Warrior: takeDamage(int damage) {

∨ void Warrior::levelUp() {

hp = max(hp - max(damage-defense, 0), 0);return damageLog();

cout << "레벨업: " << level << " -> " << level + 1 << '\m' << "최대 체력: " << maxHp << " -> " << maxHp + plusHp << '\m' << "체력: " << hp << " -> " << hp + plusHp << '\m"

<< "공격력: " << atk << " -> " << atk + plusAtk << '\m' << "방어력: " << defense + plusDefense << '\m\\\\n'; level++;

hp += plusHp; maxHp += plusHp;

defense += plusDefense;

atk += plusAtk;

### Warrior

#### Implementation of Warrior Class

```
50  void Warrior::displayInfo() const {
51  cout << "-----" "<< type << name << " ------\\n"
52  < "레벨: " << level << "\\n"
53  << "경험치: " << exp << " / 50\\n"
54  << "체력: " << hp << " / " << maxHp << "\\n"
55  << "공격력: " << atk << "\\n"
56  << "방어력: " << defense << "\\n"
57  << "골드: " << gold << "\\n\\n";
```

```
void Warrior::loadCharacter() {

try {
    ifstream ifs("character.txt");
    ifs >> type >> name >> level >> hp >> maxHp >> atk >> exp >> gold >> defense;
    type += ' ';

if(ifs.fail()) {
    cout << "캐릭터 불러오기 실패!\"n";
    throw runtime_error("파일 읽기 실패");
    }
    cout << "캐릭터 불러오기 완료!\"n";
} catch(const exception&) {</pre>
```

throw runtime\_error("저장된 게임이 없습니다.");

#### Wizard

#### Implementation of Wizard Class

```
∨ class Wizard : public Character {
 9
        protected:
           /**
10
11
            * defense: 방어력
            * plusDefense: 레벨업 할 때마다 오르는 방어력
12
13
            */
14
            int defense, plusDefense;
        public:
15
16
            /**
17
            ├* 마법사 초기 설정
            * 레벨: 1
18
            * 체력: 60
19
20
             ├* 최대 체력: 60
             * 공격력: 10
21
             * 방어력: -5
             * 경험치: 0
24
             */
25
            Wizard():
            Wizard(string name);
27
            void levelUp() override;
            void displayInfo() const override;
            void saveCharacter() override;
30
            void loadCharacter() override;
31
```

```
Wizard::Wizard(string name) {
      this->name = name;
      level = 1:
      hp = maxHp = 60;
      atk = 10:
      exp = 0:
      plusHp = 10;
      plusAtk = 6;
      defense = -5:
      plusDefense = 0;
      gold = 0;
      type = "[마법사] ";
void Wizard::levelUp() {
      cout << "레벨업: " << level << " -> " << level + 1 << '\h'
           << "최대 체력: " << maxHp << " -> " << maxHp + plusHp << '\m'
           << "체력: " << hp << " -> " << hp + plusHp << '\" '\"
           << "공격력: " << atk << " -> " << atk + plusAtk << '\m\m';
      level++;
      hp += plusHp;
      maxHp += plusHp;
      atk += plusAtk;
```

Wizard::Wizard() ': Wizard("무명") {}

16

17

18 19

20

21

24

25

26

28

29

30

31

32

33

34

35

37

38

39

40

41

## Wizard

```
void Wizard::loadCharacter() {
                                                                           try {
Implementation of Wizard Class
                                                                              ifstream ifs("character.txt");
                                                                69
                                                                              ifs >> type >> name >> level >> hp >> maxHp >> atk >> exp >> gold >> defense;
                                                                70
                                                                              type += ' ';
        void Wizard::displayInfo() const {
  43
               cout << "----- " << type << name << " ------\#n"
  44
                                                                              if(ifs.fail()) {
                    << "레벨: " << level << "\m"
                                                                                  cout << "캐릭터 불러오기 실패!₩n";
                                                                74
                    << "경험치: " << exp << " / 50\m"
                                                                                  throw runtime_error("파일 읽기 실패");
                                                               75
                    << "체력: " << hp << " / " << maxHp << "\m"
                    << "공격력: " << atk << "\n"
                                                                              cout << "캐릭터 불러오기 완료!₩n";
                    << "방어력: " << defense << "\n"
                                                                           } catch(const exception&) {
                    << "골드: " << gold << "\m\m";
                                                                              throw runtime_error("저장된 게임이 없습니다.");
                                                                79
                                                                80
  51
                                                                81
  52
        ∨ void Wizard::saveCharacter() {
  53
               ofstream ofs("character.txt");
  54
  55
               ofs << type
                   << name << ' '
                   << level << ' '
                   << hp << ' '
                   << maxHp << ' '
                   << atk << ' '
                   << exp << ' '
                   << gold << ' '
                   << defense << ' ';
               ofs.close();
  64
```

### Medicine

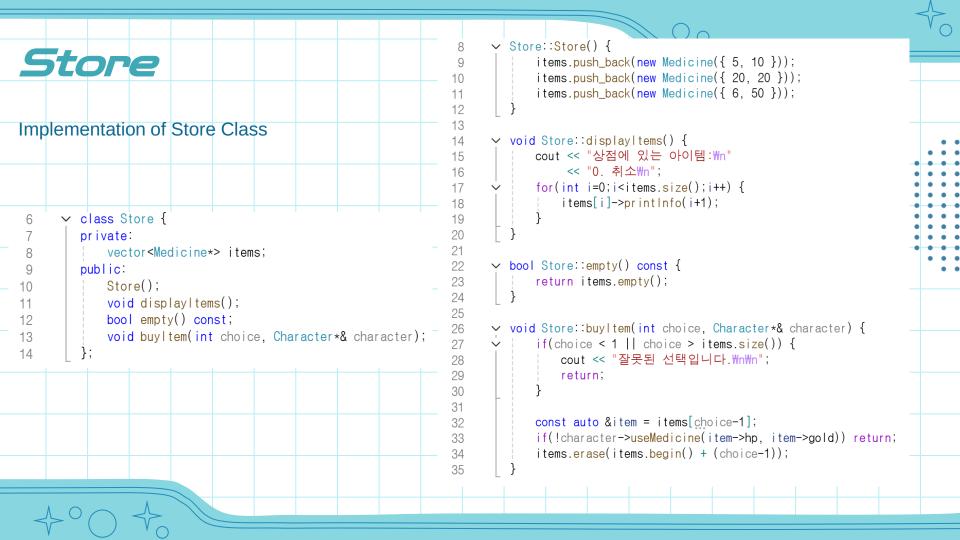
```
Implementation of Medicine Class
                                                 Medicine::Medicine(int gold, int hp) {
                                                      this->name = "회복 물약(" + to_string(hp) + ")";
                                                      this->gold = gold;

✓ class Medicine {
                                                      this->hp = hp;
           private:
               string name;

    void Medicine::printInfo(int i) const {
                                         14
           public:
                                                      cout << i << ". " << name << ": " << gold << "골드\n"
                                         15
                l* 가격: gold
                * 회복량: hp
                int gold, hp;
```



14 15 Medicine(int gold, int hp);
void printlnfo(int i) const;



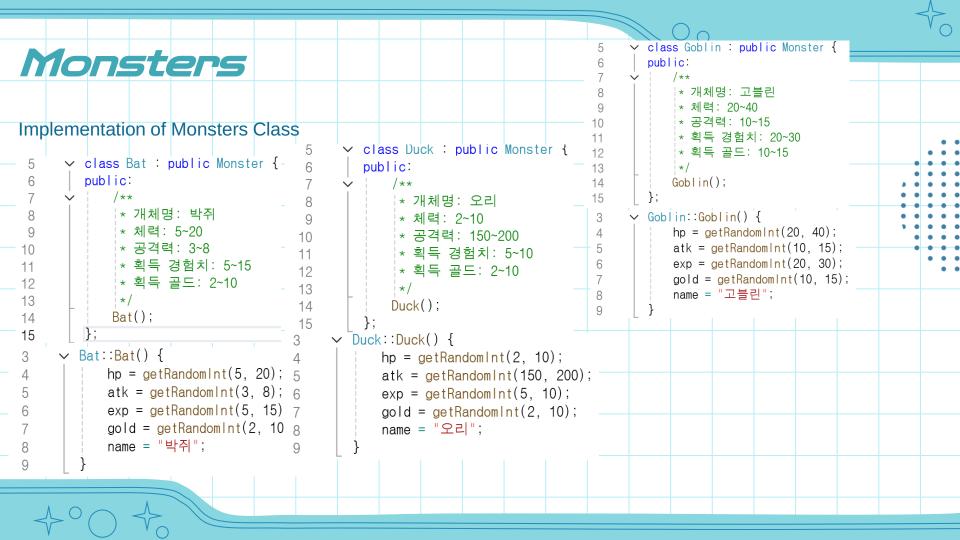
```
✓ Monster::Monster()
 Monster
                                                                      hp = 50:
                                                                      atk = 10;
                                                                      exp = 20;
                                                                      type = "[몬스터] ";
                                                                      name = "몬스터";
 Implementation of Monster Class
                                                                      gold = 10;
                                                           14
                                                           15

✓ class Monster : public Character {
                                                           16
                                                           17

∨ void Monster::displayMonster() {
          public:
                                                           18
                                                                      cout << type << name << '\m';
                                                           19
               ¦* 몬스터 초기 설정
                                                           20

∨ void Monster::saveMonster(ofstream& ofs) {
                * 체력: 50
                                                                      ofs << name << ' '
                * 공격력: 10
                                                                         << hp << ' '
                ├* 획득 경험치: 20
                                                                         << atk << ' '
                                                                         << gold << ' '
                * 획득 골드: 10
                                                                         << exp << '\m';
13
               i * /
                                                           27
               Monster();

∨ void Monster::loadMonster(ifstream& ifs) {
               void displayMonster();
15
                                                           30
                                                                      ifs >> name >> hp >> atk >> gold >> exp;
               void saveMonster(ofstream& ofs);
16
                                                           31
                                                                      if(ifs.fail()) {
                                                           32
               void loadMonster(ifstream& ifs);
                                                                         cout << "몬스터 불러오기 실패!₩n";
18
                                                           34
                                                                         throw runtime_error("파일 읽기 실패");
                                                           35
```



#### Monsters class Scorpion : public Monster { class Orc : public Monster { public: public: ★ 개체명: 전갈 Implementation of Monsters Class \* 개체명: 오크 \* 체력: 25~50 \* 체력: 50~80 \* 공격력: 12~20 ✓ class Imp : public Monster { \* 공격력: 15~25 \* 획득 경험치: 30~45 public: \* 획득 경험치: 40~60 \* 획득 골드: 20~35 \* 획득 골드: 30~50 13 13 ├\* 개체명: 임프 Scorpion(); Orc(); 14 \* 체력: 20~35 15 \* 공격력: 10~18 ∨ Orc::Orc() { Scorpion::Scorpion() { \* 획득 경험치: 20~35 hp = getRandomInt(50, 80);hp = getRandomInt(25, 50);\* 획득 골드: 15~25 atk = getRandomInt(15, 25); atk = getRandomInt(12, 20); 13 \*/ exp = getRandomInt(40, 60);exp = getRandomInt(30, 45);14 Imp(); gold = getRandomInt(30, 50); gold = getRandomInt(20, 35); 15 name = "오크": name = "전갈": | mp::|mp() | { hp = getRandomInt(20, 35),atk = getRandomInt(10, 18); exp = getRandomInt(20, 35);gold = getRandomInt(15, 25); name = "임프":

#### Monsters ✓ class Wolf : public Monster { ∨ class Snake : public Monster { public: public: Implementation of Monsters Class ├\* 개체명: 늑대 \* 개체명: 뱀 \* 체력: 25~45 \* 체력: 15~30 ✓ class Slime : public Monster \* 공격력: 12~18 \* 공격력: 5~12 public: ├\* 획득 경험치: 25~35 l\* 획득 경험치: 15~25 \* 획득 골드: 12~18 \* 획득 골드: 5~10 ├★ 개체명: 슬라임 13 \* 체력: 10~30 Wolf(); Snake(); 14 └\* 공격력: 5~15 ├\* 획득 경험치: 10~20 Wolf::Wolf() { \* 획득 골드: 5~15 ✓ Snake::Snake() { hp = getRandomInt(25, 45);13 \*/ hp = getRandomInt(15, 30);Slime(); atk = getRandomInt(12, 18); 14 atk = getRandomInt(5, 12); exp = getRandomInt(25, 35);15 exp = getRandomInt(15, 25)gold = getRandomInt(12, 18); Slime::Slime() { gold = getRandomInt(5, 10) name = "박쥐": hp = getRandomInt(10, 30); gname = "뱀"; atk = getRandomInt(5, 15); 9 exp = getRandomInt(10, 20);gold = getRandomInt(5, 15); name = "슬라임";

