```
class Person {
                                                   class Character {
 private:
                                                   private:
         Character* has character;
                                                            Person* is attached to:
         unsigned int* hit points;
                                                            void SetId(PlayerType id);
         bool is dead;
                                                   protected:
 public:
                                                            void SetHP(int hp);
         Character* hit(Character*);
                                                            PlayerType character id;
         Person(Character *);
                                                            Character& ConvertTo(PlayerType T);
         Person(PlayerType);
                                                            void Damage(int);
         int GetHP();
                                                            unsigned int hit points;
         void SetHP(int);
                                                   public:
                                                            Character (PlayerType id);
         ~Person();
                                                   virtual Character* Attack(Character&);
 };
                                                            virtual int GetHP();
                                                            virtual int GetId();
                                                   virtual void GetCharacterType();
                                                            void PrintData();
                                                            virtual void Heal();
 class Good: public Character {
                                                            virtual ~Character();
private:
                                                            //friend functions and classes
         operator Zombie();
                                                            friend Character* Convert(Character*, PlayerType);
public:
                                                            friend class Zombie;
         Good():
                                                            friend class Good;
         Good(Character&);
                                                            friend class Bad;
         Character* Attack(Character&);
                                                            friend class Person;
};
                                                            friend class World;
                                                   };
                                                             class Zombie: public Character {
class Bad: public Character {
                                                            private:
private:
                                                                     operator Good():
        operator Zombie();
                                                                     operator Bad();
public:
                                                            public:
        Bad();
                                                                     Zombie();
        Bad(Character&);
                                                                     Zombie(Character&);
        Character* Attack(Character&);
                                                                     Character* Attack(Character&);
};
                                                            };
                               class World {
                               private:
```

```
PlayerList players;
        ScoreList scores;
        bool UpdateScores();
        int round:
        void KillPlayer(int pos);
public:
        World(PlayerList init config);
        World(int number of players,
        bool generate random);
        ~World();
        void PlayRound();
        void PlayWorld();
        void DisplayWorld();
        static World Default();
        static Character*
        SpawnPlayer(PlayerType);
};
```