Classes

enum class Type {NONE,TROLL, ORC};

enum class Attack {BAT,SWORD, SHIELD, ARMOUR, AXE};

class Character {

public:

virtual void walkIntoFight();

virtual void winMessage() = 0;

virtual Type getType()=0;

virtual void choose();

virtual void chooseWeapon();

virtual void chooseBarrier();

virtual void autoChoose();

bool getAlive();

void adjustHealth(int t\_value);

Weapon\* getWeapon();

int getHealth();

void increaseRoundWeapons();

private:

bool m\_alive=true;

int m\_health=100;

Weapon\* m\_weapon=nullptr;

Bat m\_bat;

Sword m\_sword;

Shield m\_shield;

Armour m\_armour;

Axe m\_axe;

};

class Orc : public Character{

public:

void barrelRoll();

void walkIntoFight();

void winMessage();

Type getType();

};

class Troll : public Character{

public:

void winMessage();

void walkIntoFight();

Type getType();

void motivation();

};

class Weapon {

public:

Weapon(int t\_coolDown=0, int t\_unusedRounds=0);

virtual Attack getWeaponType()=0;

virtual bool canUse();

void increaseUnusedRounds();

void setUnusedRounds(int t\_value);

int getUnusedRounds();

private:

int m\_coolDown;

int m\_unusedRounds;

};

class Bat:public Weapon

{

public:

Bat();

Attack getWeaponType();

bool canUse();

};

class Sword:public Weapon

{

public:

Sword();

Attack getWeaponType();

};

class Shield:public Weapon

{

public:

Shield();

Attack getWeaponType();

};

class Armour:public Weapon

{

public:

Armour();

Attack getWeaponType();

};

class Axe:public Weapon

{

public:

Axe();

Attack getWeaponType();

};