**Санкт-Петербургский**

**Государственный электротехнический университет**

**Факультет Компьютерных Технологий и Информатики**

**Кафедра МО ЭВМ**

****

*Учебная практика.*

*4 семестр.*

**Выполняли: Комаров Д.Н.**

**Климук К.П.**

**Лихачев А.А.**

**Проверил: Рябиков Э.М.**

***Санкт-Петербург***

***2014 год.***

1. **Постановка задачи:**

Реализовать пошаговую игру жанра Fighting с элементами RPG на языке программировании Java и использованием GUI.

1. **Описание игры:**

Игроку дается по два очка защиты и атаки. Каждое очко тратится на выбор, какую часть защищать или атаковать. При удачной защите урона нет, иначе случайным образом в зависимости от параметров противников рассчитывается урон. Каждая часть тел имеет собственные жизни. Если сломаны рука или нога, то теряется одно очко защиты или атаки. Для выигрыша требуется одно из условий:

- нанесено достаточное кол-во урона.

- сломана голова противника

- сломаны все руки и ноги противника

- сломано “туловище” противника

Игра ведется против бота, который случайным образом выбирает куда бить и что защищать.

1. **Описание системы классов:**

Для реализации данной задачи были созданы три группы классов, между которыми происходит взаимодействие.

Первая группа – это классы, моделирующие самих игроков. Сюда относится класс Character, который содержит общую информацию об игроках, как HP, атака, защита и т.п. и классы Bot и Player, унаследованные от Character и содержащую информацию, касающуюся только отдельно их. К примеру, класс Bot имеет функции для случайного выбора защиты\атаки, которые не нужны Player, и функцию изменения параметров и генерирования боевого лога. Класс Player же не нуждается в рандомизации и содержит помимо основной информации в себе информацию об опыте персонажа, функции изменения параметров и генерации боевого лога.

Вторая группа – это классы форм интерфейса. Это такие классы, как Battle, Menu и LevelUp.

Вся эта группа относится к GUI и содержит функции, которые пересылают информацию о событиях в третью группу классов. К примеру, Menu отправляет сообщение, что игрок хочет сохранить\загрузить своего персонажа, Battle пересылает сообщения о выборе игрока во время боя, а LevelUp – какую характеристику улучшить.

Третья группа – это связующее звено между первыми двумя. Сюда относятся классы Server и Message. Класс Message – это класс, содержащий в себе всю передаваемую от интерфейсов информацию. Класс Server же обрабатывает эти сообщения и говорит классам Player и Bot, что от них требуется. К примеру, Menu отправило сообщение об сохранении игрока, сообщение приходит на Server и он просит Player передать ему все данные для сохранения в файл. Или Battle переслало сообщение об выборе игрока и Server говорит Player и Bot, как именно требуется изменить свои параметры.

**4. Текст программы:**

Server.java

**package** gameServer**;**

**import** сharacter.Player**;**

**import** сharacter.Bot**;**

**import** gameForms.LevelUp**;**

**import** gameForms.Battle**;**

**import** gameForms.Menu**;**

**import** java.io.\***;**

**import** javax.swing.JOptionPane**;**

**public** **class** **Server**

**{**

**private** Player player**;**

**private** Bot bot**;**

**private** Battle battleRoom**;**

**private** LevelUp levelUpRoom**;**

**private** Menu menuRoom**;**

**public** **Server(){**

        player **=** **new** **Player();**

**}**

**public** **final** Bot **getBot(){**

**return** bot**;**

**}**

**public** **final** Player **getPlayer(){**

**return** player**;**

**}**

**public** String **translateMessage(**Message msg**)** **throws** java**.**io**.**IOException**{**

**if** **(**msg **==** **null)**

**return** "NullMessage"**;**

        String newMessage **=** **new** **String();**

**switch(**msg**.**getType**())**

**{**

**case(**"choice"**):**

                    newMessage **=** choice**(**msg**.**getChoice**());**

**break;**

**case(**"battleStep"**):**

                    newMessage **=** battleStep**(**msg**.**getPlayerAttack**(),**

                                            msg**.**getPlayerDefense**());**

**break;**

**case(**"create"**):**

**if(**msg**.**getChoice**()** **==** 0**)**

                        createBattleRoom**();**

**if(**msg**.**getChoice**()** **==** 1**)**

                        createMenuRoom**();**

**if(**msg**.**getChoice**()** **==** 2**)**

                        createLevelUpRoom**();**

**break;**

**case(**"save"**):**

                    savePlayer**();**

**break;**

**case(**"load"**):**

                    loadPlayer**();**

**break;**

**}**

**return** newMessage**;**

**}**

**private** String **battleStep(final** **boolean[]** attack**,** **final** **boolean[]** defense**){**

        String battleLog **=** **new** **String();**

        battleLog **+=** bot**.**battleStep**(**attack**,** bot**.**choiceDefense**(),**

                                            player**.**getAttack**());**

        battleLog **+=** "\n"**;**

        battleLog **+=** player**.**battleStep**(**bot**.**choiceAttack**(**player**.**getBreakParts**()),**

                                       defense**,**

                                       bot**.**getAttack**());**

        battleLog **+=** "\n"**;**

**for** **(int** i **=** 0**;** i **<** 6**;** **++**i**)**

**{**

**if(**player**.**getBreakParts**()[**0**][**i**]** **&&** **!**player**.**getBreakParts**()[**1**][**i**])**

**{**

**for** **(int** j **=** 0**;** j **<** 6**;** **++**j**)**

**{**

**if** **(**battleRoom**.**attackList**.**get**(**j**).**isSelected**())** battleRoom**.**attackList**.**get**(**j**).**doClick**();**

**if** **(**battleRoom**.**defenseList**.**get**(**j**).**isSelected**())** battleRoom**.**defenseList**.**get**(**j**).**doClick**();**

**}**

                    player**.**setInvalidBreakPart**(**i**);**

**}**

**if(**bot**.**getBreakParts**()[**0**][**i**]** **&&** **!**bot**.**getBreakParts**()[**1**][**i**])**

**{**

                    battleRoom**.**attackList**.**get**(**i**).**doClick**();**

                    battleRoom**.**attackList**.**get**(**i**).**setEnabled**(false);**

                    bot**.**setInvalidBreakPart**(**i**);**

**}**

**}**

**return** battleLog**;**

**}**

**private** String **choice(final** **int** choice**){**

**if** **(**choice **==** 0**)**

            player**.**upMaxHP**(**60**);**

**else** **if(**choice **==** 1**)**

            player**.**upAttack**(**2**);**

**else**

            player**.**upDefense**(**2**);**

        player**.**UpdateHp**();**

**return** **null;**

**}**

**private** **void** **createBattleRoom(){**

**try** **{**

**for** **(**javax**.**swing**.**UIManager**.**LookAndFeelInfo info **:** javax**.**swing**.**UIManager**.**getInstalledLookAndFeels**())** **{**

**if** **(**"Nimbus"**.**equals**(**info**.**getName**()))** **{**

                    javax**.**swing**.**UIManager**.**setLookAndFeel**(**info**.**getClassName**());**

**break;**

**}**

**}**

**}** **catch** **(**ClassNotFoundException ex**)** **{**

            java**.**util**.**logging**.**Logger**.**getLogger**(**Battle**.**class**.**getName**()).**log**(**java**.**util**.**logging**.**Level**.**SEVERE**,** **null,** ex**);**

**}** **catch** **(**InstantiationException ex**)** **{**

            java**.**util**.**logging**.**Logger**.**getLogger**(**Battle**.**class**.**getName**()).**log**(**java**.**util**.**logging**.**Level**.**SEVERE**,** **null,** ex**);**

**}** **catch** **(**IllegalAccessException ex**)** **{**

            java**.**util**.**logging**.**Logger**.**getLogger**(**Battle**.**class**.**getName**()).**log**(**java**.**util**.**logging**.**Level**.**SEVERE**,** **null,** ex**);**

**}** **catch** **(**javax**.**swing**.**UnsupportedLookAndFeelException ex**)** **{**

            java**.**util**.**logging**.**Logger**.**getLogger**(**Battle**.**class**.**getName**()).**log**(**java**.**util**.**logging**.**Level**.**SEVERE**,** **null,** ex**);**

**}**

        bot **=** **new** **Bot(**player**.**getLevel**());**

        bot**.**UpdateHp**();**

        player**.**UpdateHp**();**

**final** Server server **=** **this;**

        java**.**awt**.**EventQueue**.**invokeLater**(new** **Runnable()** **{**

            @Override

**public** **void** **run()** **{**

                battleRoom **=** **new** **Battle(**server**);**

                battleRoom**.**setVisible**(true);**

**}**

**});**

        menuRoom **=** **null;**

        levelUpRoom **=** **null;**

**}**

**private** **void** **createMenuRoom(){**

**final** Server server **=** **this;**

        java**.**awt**.**EventQueue**.**invokeLater**(new** **Runnable()** **{**

            @Override

**public** **void** **run()** **{**

                menuRoom **=** **new** **Menu(**server**);**

                menuRoom**.**setVisible**(true);**

**}**

**});**

        battleRoom **=** **null;**

        levelUpRoom **=** **null;**

**}**

**private** **void** **createLevelUpRoom(){**

**try** **{**

**for** **(**javax**.**swing**.**UIManager**.**LookAndFeelInfo info **:** javax**.**swing**.**UIManager**.**getInstalledLookAndFeels**())** **{**

**if** **(**"Nimbus"**.**equals**(**info**.**getName**()))** **{**

                            javax**.**swing**.**UIManager**.**setLookAndFeel**(**info**.**getClassName**());**

**break;**

**}**

**}**

**}** **catch** **(**ClassNotFoundException ex**)** **{**

                    java**.**util**.**logging**.**Logger**.**getLogger**(**LevelUp**.**class**.**getName**()).**log**(**java**.**util**.**logging**.**Level**.**SEVERE**,** **null,** ex**);**

**}** **catch** **(**InstantiationException ex**)** **{**

                    java**.**util**.**logging**.**Logger**.**getLogger**(**LevelUp**.**class**.**getName**()).**log**(**java**.**util**.**logging**.**Level**.**SEVERE**,** **null,** ex**);**

**}** **catch** **(**IllegalAccessException ex**)** **{**

                    java**.**util**.**logging**.**Logger**.**getLogger**(**LevelUp**.**class**.**getName**()).**log**(**java**.**util**.**logging**.**Level**.**SEVERE**,** **null,** ex**);**

**}** **catch** **(**javax**.**swing**.**UnsupportedLookAndFeelException ex**)** **{**

                    java**.**util**.**logging**.**Logger**.**getLogger**(**LevelUp**.**class**.**getName**()).**log**(**java**.**util**.**logging**.**Level**.**SEVERE**,** **null,** ex**);**

**}**

**final** Server serv **=** **this;**

        java**.**awt**.**EventQueue**.**invokeLater**(new** **Runnable()** **{**

            @Override

**public** **void** **run()** **{**

                levelUpRoom **=** **new** **LevelUp(**serv**);**

                levelUpRoom**.**setVisible**(true);**

**}**

**});**

        battleRoom **=** **null;**

        menuRoom **=** **null;**

**}**

**private** **void** **savePlayer()** **throws** java**.**io**.**IOException**{**

        String fileName **=** JOptionPane**.**showInputDialog**(**menuRoom**,** "Введите название файла."**);**

        PrintWriter out **=** **new** **PrintWriter(new** **BufferedWriter**

**(new** **FileWriter(**"characters/" **+** fileName **+** ".chr"**)));**

        out**.**println**(**Integer**.**toString**(**player**.**getAttack**()));**

        out**.**println**(**Integer**.**toString**(**player**.**getCurExp**()));**

        out**.**println**(**Integer**.**toString**(**player**.**getDefense**()));**

        out**.**println**(**Integer**.**toString**(**player**.**getExpForLevel**()));**

        out**.**println**(**Integer**.**toString**(**player**.**getLevel**()));**

        out**.**println**(**Integer**.**toString**(**player**.**getMaxHP**()));**

**for** **(int** i **=** 0**;** i **<** 6**;** **++**i**)**

            out**.**println**(**Integer**.**toString**(**player**.**getMaxHpParts**()[**i**]));**

        out**.**flush**();**

**}**

**private** **void** **loadPlayer()** **throws** java**.**io**.**IOException**{**

         String fileName **=** JOptionPane**.**showInputDialog**(**menuRoom**,** "Введите название файла."**);**

         player **=** **new** **Player(**"characters/" **+** fileName **+** ".chr"**);**

**}**

**}**

Message.java

**package** gameServer**;**

**public** **class** **Message**

**{**

**private** **final** String type**;**

**private** **final** **int** choice**;**

**private** **final** **boolean[]** playerAttack**;**

**private** **final** **boolean[]** playerDefense**;**

**public** **final** **boolean[]** **getPlayerAttack(){**

**return** playerAttack**;**

**}**

**public** **final** **boolean[]** **getPlayerDefense(){**

**return** playerDefense**;**

**}**

**public** **final** **int** **getChoice(){**

**return** choice**;**

**}**

**public** **final** String **getType(){**

**return** type**;**

**}**

**public** **Message(final** String type**){**

**this.**type **=** type**;**

        choice **=** 0**;**

**this.**playerAttack **=** **null;**

**this.**playerDefense **=** **null;**

**}**

**public** **Message(final** String type**,** **final** **int** choice**){**

**this.**choice **=** choice**;**

**this.**type **=** type**;**

**this.**playerAttack **=** **null;**

**this.**playerDefense **=** **null;**

**}**

**public** **Message(final** **boolean[]** playerAttack**,** **final** **boolean[]** playerDefense**)** **{**

**this.**playerAttack **=** playerAttack**;**

**this.**playerDefense **=** playerDefense**;**

        type **=** "battleStep"**;**

        choice **=** 0**;**

**}**

**}**

**Player.java**

**package** сharacter**;**

**import** java.util.Random**;**

**import** java.io.\***;**

**public** **class** **Player** **extends** Character**{**

**private** **int** currentExp**;**

**private** **int** expForLevel**;**

**private** **int** level**;**

**public** **Player(){**

        maxHp **=** 200**;**

        currentHp **=** maxHp**;**

        attack **=** 7**;**

        defense **=** 5**;**

        currentExp **=** 0**;**

        expForLevel **=** 10**;**

        level **=** 1**;**

        maxHpParts **=** **new** **int[**6**];**

        currentHpParts **=** **new** **int[**6**];**

**for(int** i **=** 2**;** i **<** 6**;** **++**i**)**

**{**

            maxHpParts**[**i**]** **=** 75**;**

            currentHpParts**[**i**]** **=** 75**;**

**}**

        maxHpParts**[**0**]** **=** 150**;**

        maxHpParts**[**1**]** **=** 200**;**

        currentHpParts**[**0**]** **=** 150**;**

        currentHpParts**[**1**]** **=** 200**;**

        countDefensePoints **=** 2**;**

        countAttackPoints **=** 2**;**

        breakParts **=** **new** **boolean[**2**][];**

        breakParts**[**0**]** **=** **new** **boolean[**6**];**

        breakParts**[**1**]** **=** **new** **boolean[**6**];**

**}**

**public** **Player(final** String fileName**)** **throws** java**.**io**.**IOException **{**

        BufferedReader in **=** **new** **BufferedReader(new** **FileReader(new** **File(**fileName**)**

**.**getAbsoluteFile**()));**

        attack **=** Integer**.**parseInt**(**in**.**readLine**());**

        currentExp **=** Integer**.**parseInt**(**in**.**readLine**());**

        defense **=** Integer**.**parseInt**(**in**.**readLine**());**

        expForLevel **=** Integer**.**parseInt**(**in**.**readLine**());**

        level **=** Integer**.**parseInt**(**in**.**readLine**());**

        maxHp **=** Integer**.**parseInt**(**in**.**readLine**());**

        currentHp **=** maxHp**;**

        maxHpParts **=** **new** **int[**6**];**

        currentHpParts **=** **new** **int[**6**];**

**for(int** i **=** 0**;** i **<** 6**;** **++**i**)**

**{**

            maxHpParts**[**i**]** **=** Integer**.**parseInt**(**in**.**readLine**());**

            currentHpParts**[**i**]** **=** maxHpParts**[**i**];**

**}**

        countDefensePoints **=** 2**;**

        countAttackPoints **=** 2**;**

        breakParts **=** **new** **boolean[**2**][];**

        breakParts**[**0**]** **=** **new** **boolean[**6**];**

        breakParts**[**1**]** **=** **new** **boolean[**6**];**

**}**

**public** **final** **int** **getLevel(){**

**return** level**;**

**}**

**public** **final** **int** **getExpForLevel(){**

**return** expForLevel**;**

**}**

**public** **final** **int** **getCurExp(){**

**return** currentExp**;**

**}**

**public** **final** **int** **getAttackPoints(){**

**return** countAttackPoints**;**

**}**

**public** **final** **int** **getDefensePoints(){**

**return** countDefensePoints**;**

**}**

**public** **void** **giveExp(){**

        currentExp **+=** 15**;**

**}**

**public** **void** **levelUp(){**

        expForLevel **<<=** 1**;**

**++**level**;**

**}**

    @Override

**public** String **battleStep(boolean[]** enemyAttack**,**

**boolean[]** playerDefense**,**

**int** enemyAttackPoints**)**

**{**

        String battle\_log **=** "Ход бота:\n"**;**

        Random rand **=** **new** **Random();**

**int** at1 **=** **-**1**,** at2 **=** **-**1**;**

**int** def1 **=** **-**1**,** def2 **=** **-**1**;**

**boolean** at **=** **false,** def **=** **false;**

**for** **(int** i **=** 0**;** i **<** 6**;** **++**i**)**

**{**

**if** **(**enemyAttack**[**i**])**

**{**

**if** **(!**at**)**

**{**

                    at1 **=** i**;**

                    at **=** **true;**

**}**

**else**

                    at2 **=** i**;**

**}**

**if** **(**playerDefense**[**i**])**

**if** **(!**def**)**

**{**

                    def1 **=** i**;**

                    def **=** **true;**

**}**

**else**

                    def2 **=** i**;**

**}**

**int** damage **=** 0**;**

**if** **(**def1 **!=** **-**1 **&&** def2 **!=** **-**1 **&&** def1 **==** at1 **||** def2 **==** at1**)**

            battle\_log **+=** "Игрок парировал удар противника.\n"**;**

**else** **if(**at1 **!=** **-**1**)**

**{**

            damage **=** rand**.**nextInt**(**enemyAttackPoints**)** **-** rand**.**nextInt**(**defense**);**

**switch(**at1**){**

**case(**0**):**

                    damage **+=** rand**.**nextInt**(**5**);**

**break;**

**case(**1**):**

                    damage **+=** rand**.**nextInt**(**3**);**

**break;**

**case(**2**):**

                    damage **+=** rand**.**nextInt**(**2**);**

**break;**

**}**

**if** **(**damage **<** 0**)**

                damage **=** 0**;**

            currentHpParts**[**at1**]** **-=** damage**;**

            currentHp **-=** damage**;**

            battle\_log **+=** "Бот нанес игроку " **+** Integer**.**toString**(**damage**)** **+** " урона.\n"**;**

**if** **(**currentHpParts**[**at1**]** **<=** 0**)**

**{**

                battle\_log **+=** "Бот сломал игроку" **+** deadParts**(**at1**);**

                breakParts**[**0**][**at1**]** **=** **true;**

**if** **(**at1 **>** 1**)**

**{**

**if** **(**rand**.**nextInt**(**2**)** **==** 0**)**

**if** **(**countAttackPoints **!=** 0**)**

**--**countAttackPoints**;**

**else**

**--**countDefensePoints**;**

**else**

**if** **(**countDefensePoints **!=** 0**)**

**--**countDefensePoints**;**

**else**

**--**countAttackPoints**;**

**}**

**}**

**}**

**if** **(**def1 **!=** **-**1 **&&** def2 **!=** **-**1 **&&** def1 **==** at2 **||** def2 **==** at2**)**

            battle\_log **+=** "Игрок парировал удар противника.\n"**;**

**else** **if** **(**at2 **!=** **-**1**)**

**{**

            damage **=** rand**.**nextInt**(**enemyAttackPoints**)** **-** rand**.**nextInt**(**defense**);**

**switch(**at2**){**

**case(**0**):**

                    damage **+=** rand**.**nextInt**(**5**);**

**break;**

**case(**1**):**

                    damage **+=** rand**.**nextInt**(**3**);**

**break;**

**case(**2**):**

                    damage **+=** rand**.**nextInt**(**2**);**

**break;**

**}**

**if** **(**damage **<** 0**)**

                damage **=** 0**;**

            currentHpParts**[**at2**]** **-=** damage**;**

            currentHp **-=** damage**;**

            battle\_log **+=** "Бот нанес игроку " **+** Integer**.**toString**(**damage**)** **+** " урона.\n"**;**

**if** **(**currentHpParts**[**at2**]** **<=** 0**)**

**{**

                battle\_log **+=** "Бот сломал игроку" **+** deadParts**(**at2**);**

                breakParts**[**0**][**at2**]** **=** **true;**

**if** **(**at2 **>** 1**)**

**{**

**if** **(**rand**.**nextInt**(**2**)** **==** 0**)**

**if** **(**countAttackPoints **!=** 0**)**

**--**countAttackPoints**;**

**else**

**--**countDefensePoints**;**

**else**

**if** **(**countDefensePoints **!=** 0**)**

**--**countDefensePoints**;**

**else**

**--**countAttackPoints**;**

**}**

**}**

**}**

**return** battle\_log**;**

**}**

**}**

 Character.java

**package** сharacter**;**

**public** **abstract** **class** **Character{**

**protected** **int** maxHp**;**

**protected** **int** currentHp**;**

**protected** **int[]** maxHpParts**;**

**protected** **int[]** currentHpParts**;**

**protected** **int** attack**;**

**protected** **int** defense**;**

**protected** **int** countAttackPoints**;**

**protected** **int** countDefensePoints**;**

**protected** **boolean[][]** breakParts**;**

**public** **int** **getMaxHP(){**

**return** maxHp**;**

**}**

**public** **int** **getCurHP(){**

**return** currentHp**;**

**}**

**public** **int** **getAttack(){**

**return** attack**;**

**}**

**public** **int** **getDefense(){**

**return** defense**;**

**}**

**public** **int[]** **getMaxHpParts(){**

**return** maxHpParts**;**

**}**

**public** **int[]** **getCurrentHpParts(){**

**return** currentHpParts**;**

**}**

**public** **boolean[][]** **getBreakParts(){**

**return** breakParts**;**

**}**

**public** **void** **setInvalidBreakPart(final** **int** part**){**

**this.**breakParts**[**1**][**part**]** **=** **true;**

**}**

**public** **boolean** **isDead(){**

**return** breakParts**[**0**][**0**]** **||** breakParts**[**0**][**1**]** **||**

**(**breakParts**[**0**][**2**]** **&&** breakParts**[**0**][**3**]** **&&**

               breakParts**[**0**][**4**]** **&&** breakParts**[**0**][**5**])** **||**

               currentHp **<=** 0**;**

**}**

**public** **void** **upMaxHP(final** **int** value**){**

        maxHp **+=** value**;**

        maxHpParts**[**0**]** **+=** value**/**12**;**

**for** **(int** i **=** 2**;** i **<** 6**;** **++**i**)**

            maxHpParts**[**i**]** **+=** value**/**6**;**

        maxHpParts**[**1**]** **+=** value**\***7**/**12**;**

**}**

**public** **void** **upAttack(final** **int** value**){**

        attack **+=** value**;**

**}**

**public** **void** **upDefense(final** **int** value**){**

        defense **+=** value**;**

**}**

**public** **void** **UpdateHp(){**

        currentHp **=** maxHp**;**

**for** **(int** i **=** 0**;** i **<** 6**;** **++**i**)**

            currentHpParts**[**i**]** **=** maxHpParts**[**i**];**

**}**

**protected** String **deadParts(final** **int** value**){**

        String str **=** **new** **String();**

**switch(**value**){**

**case(**0**):**

                str **=** " голову.\n"**;**

**break;**

**case(**2**):**

                str **=** " правую руку.\n"**;**

**break;**

**case(**3**):**

                str **=** " левую руку.\n"**;**

**break;**

**case(**4**):**

                str **=** " правую ногу.\n"**;**

**break;**

**case(**5**):**

                str **=** " левую ногу.\n"**;**

**break;**

**}**

**return** str**;**

**}**

**public** **abstract** String **battleStep(boolean[]** enemyАttack**,** **boolean[]** playerDefense**,** **int** enemyAttackPoints**);**

**}**

Bot.java

**package** сharacter**;**

**import** java.util.Random**;**

**public** **class** **Bot** **extends** Character**{**

**public** **Bot(int** level**){**

        maxHp **=** 200**;**

        attack **=** 7**;**

        defense **=** 5**;**

        maxHpParts **=** **new** **int[**6**];**

        currentHpParts **=** **new** **int[**6**];**

**for(int** i **=** 2**;** i **<** 6**;** **++**i**)**

**{**

            maxHpParts**[**i**]** **=** 75**;**

            currentHpParts**[**i**]** **=** 75**;**

**}**

        maxHpParts**[**0**]** **=** 150**;**

        maxHpParts**[**1**]** **=** 200**;**

        Random rand **=** **new** **Random();**

**for** **(int** i **=** 0**;** i **<** level**;** **++**i**)**

**{**

**int** r **=** rand**.**nextInt**(**3**);**

**if** **(**r **==** 0**)**

**this.**upMaxHP**(**60**);**

**else** **if** **(**r **==** 1**)**

**this.**upAttack**(**2**);**

**else** **if** **(**r **==** 2**)**

**this.**upDefense**(**2**);**

**}**

        countDefensePoints **=** 2**;**

        countAttackPoints **=** 2**;**

        breakParts **=** **new** **boolean[**2**][];**

        breakParts**[**0**]** **=** **new** **boolean[**6**];**

        breakParts**[**1**]** **=** **new** **boolean[**6**];**

**}**

**public** **boolean[]** **choiceAttack(boolean[][]** breakParts**)** **{**

**boolean[]** attack **=** **new** **boolean[**6**];**

        Random rand **=** **new** **Random();**

**if** **(**countAttackPoints **!=** 0**)**

**{**

**int[]** at **=** **new** **int[**countAttackPoints**];**

            at**[**0**]** **=** rand**.**nextInt**(**6**);**

**while** **(**breakParts**[**0**][**at**[**0**]])**

**{**

                at**[**0**]** **=** rand**.**nextInt**(**6**);**

**}**

**if** **(**countAttackPoints **==** 2**)**

**{**

                at**[**1**]** **=** rand**.**nextInt**(**6**);**

**while** **(**at**[**0**]** **==** at**[**1**]** **||** breakParts**[**0**][**at**[**0**]]** **||** breakParts**[**0**][**at**[**1**]])**

**{**

                    at**[**0**]** **=** rand**.**nextInt**(**6**);**

                    at**[**1**]** **=** rand**.**nextInt**(**6**);**

**}**

                attack**[**at**[**1**]]** **=** **true;**

**}**

            attack**[**at**[**0**]]** **=** **true;**

**}**

**return** attack**;**

**}**

**public** **boolean[]** **choiceDefense()** **{**

**boolean[]** defense **=** **new** **boolean[**6**];**

        Random rand **=** **new** **Random();**

**if** **(**countDefensePoints **!=** 0**)**

**{**

**int[]** def **=** **new** **int[**countDefensePoints**];**

            def**[**0**]** **=** rand**.**nextInt**(**6**);**

**while(**breakParts**[**0**][**def**[**0**]])**

                def**[**0**]** **=** rand**.**nextInt**(**6**);**

**if** **(**countDefensePoints **==** 2**)**

**{**

                def**[**1**]** **=** rand**.**nextInt**(**6**);**

**while** **(**def**[**0**]** **==** def**[**1**]** **||** breakParts**[**0**][**def**[**0**]]**

**||** breakParts**[**0**][**def**[**1**]])**

**{**

                    def**[**0**]** **=** rand**.**nextInt**(**6**);**

                    def**[**1**]** **=** rand**.**nextInt**(**6**);**

**}**

                defense**[**def**[**1**]]** **=** **true;**

**}**

            defense**[**def**[**0**]]** **=** **true;**

**}**

**return** defense**;**

**}**

    @Override

**public** String **battleStep(boolean[]** enemyAttack**,** **boolean[]** playerDefense**,** **int** enemyAttackPoints**)**

**{**

        String battle\_log **=** "Ход игрока:\n"**;**

        Random rand **=** **new** **Random();**

**int** at1 **=** **-**1**,** at2 **=** **-**1**;**

**int** def1 **=** **-**1**,** def2 **=** **-**1**;**

**boolean** at **=** **false,** def **=** **false;**

**for** **(int** i **=** 0**;** i **<** 6**;** **++**i**)**

**{**

**if** **(**enemyAttack**[**i**])**

**{**

**if** **(!**at**)**

**{**

                    at1 **=** i**;**

                    at **=** **true;**

**}**

**else**

                    at2 **=** i**;**

**}**

**if** **(**playerDefense**[**i**])**

**if** **(!**def**)**

**{**

                    def1 **=** i**;**

                    def **=** **true;**

**}**

**else**

                    def2 **=** i**;**

**}**

**int** damage **=** 0**;**

**if** **(**def1 **!=** **-**1 **&&** def2 **!=** **-**1 **&&** def1 **==** at1 **||** def2 **==** at1**)**

            battle\_log **+=** "Бот парировал удар игрока.\n"**;**

**else** **if(**at1 **!=** **-**1**)**

**{**

            damage **=** rand**.**nextInt**(**enemyAttackPoints**)** **-** rand**.**nextInt**(**defense**);**

**switch(**at1**){**

**case(**0**):**

                    damage **+=** rand**.**nextInt**(**5**);**

**break;**

**case(**1**):**

                    damage **+=** rand**.**nextInt**(**3**);**

**break;**

**case(**2**):**

                    damage **+=** rand**.**nextInt**(**2**);**

**break;**

**}**

**if** **(**damage **<** 0**)**

                damage **=** 0**;**

            currentHpParts**[**at1**]** **-=** damage**;**

            currentHp **-=** damage**;**

            battle\_log **+=** "Игрок нанес боту " **+** Integer**.**toString**(**damage**)** **+** " урона.\n"**;**

**if** **(**currentHpParts**[**at1**]** **<=** 0**)**

**{**

                battle\_log **+=** "Игрок сломал боту" **+** deadParts**(**at1**);**

                breakParts**[**0**][**at1**]** **=** **true;**

**if** **(**at1 **>** 1**)**

**{**

**if** **(**rand**.**nextInt**(**2**)** **==** 0**)**

**if** **(**countAttackPoints **!=** 0**)**

**--**countAttackPoints**;**

**else**

**--**countDefensePoints**;**

**else**

**if** **(**countDefensePoints **!=** 0**)**

**--**countDefensePoints**;**

**else**

**--**countAttackPoints**;**

**}**

**}**

**}**

**if** **(**def1 **!=** **-**1 **&&** def2 **!=** **-**1 **&&** def1 **==** at2 **||** def2 **==** at2**)**

            battle\_log **+=** "Бот парировал удар игрока.\n"**;**

**else** **if(**at2 **!=** **-**1**)**

**{**

            damage **=** rand**.**nextInt**(**enemyAttackPoints**)** **-** rand**.**nextInt**(**defense**);**

**switch(**at2**){**

**case(**0**):**

                    damage **+=** rand**.**nextInt**(**5**);**

**break;**

**case(**1**):**

                    damage **+=** rand**.**nextInt**(**3**);**

**break;**

**case(**2**):**

                    damage **+=** rand**.**nextInt**(**2**);**

**break;**

**}**

**if** **(**damage **<** 0**)**

                damage **=** 0**;**

            currentHpParts**[**at2**]** **-=** damage**;**

            currentHp **-=** damage**;**

            battle\_log **+=** "Игрок нанес боту " **+** Integer**.**toString**(**damage**)** **+** " урона.\n"**;**

**if** **(**currentHpParts**[**at2**]** **<=** 0**)**

**{**

                battle\_log **+=** "Игрок сломал боту" **+** deadParts**(**at2**);**

                breakParts**[**0**][**at2**]** **=** **true;**

**if** **(**at2 **>** 1**)**

**{**

**if** **(**rand**.**nextInt**(**2**)** **==** 0**)**

**if** **(**countAttackPoints **!=** 0**)**

**--**countAttackPoints**;**

**else**

**--**countDefensePoints**;**

**else**

**if** **(**countDefensePoints **!=** 0**)**

**--**countDefensePoints**;**

**else**

**--**countAttackPoints**;**

**}**

**}**

**}**

**return** battle\_log**;**

**}**

**}**

Player.java

**package** сharacter**;**

**import** java.util.Random**;**

**import** java.io.\***;**

**public** **class** **Player** **extends** Character**{**

**private** **int** currentExp**;**

**private** **int** expForLevel**;**

**private** **int** level**;**

**public** **Player(){**

        maxHp **=** 200**;**

        currentHp **=** maxHp**;**

        attack **=** 7**;**

        defense **=** 5**;**

        currentExp **=** 0**;**

        expForLevel **=** 10**;**

        level **=** 1**;**

        maxHpParts **=** **new** **int[**6**];**

        currentHpParts **=** **new** **int[**6**];**

**for(int** i **=** 2**;** i **<** 6**;** **++**i**)**

**{**

            maxHpParts**[**i**]** **=** 75**;**

            currentHpParts**[**i**]** **=** 75**;**

**}**

        maxHpParts**[**0**]** **=** 150**;**

        maxHpParts**[**1**]** **=** 200**;**

        currentHpParts**[**0**]** **=** 150**;**

        currentHpParts**[**1**]** **=** 200**;**

        countDefensePoints **=** 2**;**

        countAttackPoints **=** 2**;**

        breakParts **=** **new** **boolean[**2**][];**

        breakParts**[**0**]** **=** **new** **boolean[**6**];**

        breakParts**[**1**]** **=** **new** **boolean[**6**];**

**}**

**public** **Player(final** String fileName**)** **throws** java**.**io**.**IOException **{**

        BufferedReader in **=** **new** **BufferedReader(new** **FileReader(new** **File(**fileName**)**

**.**getAbsoluteFile**()));**

        attack **=** Integer**.**parseInt**(**in**.**readLine**());**

        currentExp **=** Integer**.**parseInt**(**in**.**readLine**());**

        defense **=** Integer**.**parseInt**(**in**.**readLine**());**

        expForLevel **=** Integer**.**parseInt**(**in**.**readLine**());**

        level **=** Integer**.**parseInt**(**in**.**readLine**());**

        maxHp **=** Integer**.**parseInt**(**in**.**readLine**());**

        currentHp **=** maxHp**;**

        maxHpParts **=** **new** **int[**6**];**

        currentHpParts **=** **new** **int[**6**];**

**for(int** i **=** 0**;** i **<** 6**;** **++**i**)**

**{**

            maxHpParts**[**i**]** **=** Integer**.**parseInt**(**in**.**readLine**());**

            currentHpParts**[**i**]** **=** maxHpParts**[**i**];**

**}**

        countDefensePoints **=** 2**;**

        countAttackPoints **=** 2**;**

        breakParts **=** **new** **boolean[**2**][];**

        breakParts**[**0**]** **=** **new** **boolean[**6**];**

        breakParts**[**1**]** **=** **new** **boolean[**6**];**

**}**

**public** **final** **int** **getLevel(){**

**return** level**;**

**}**

**public** **final** **int** **getExpForLevel(){**

**return** expForLevel**;**

**}**

**public** **final** **int** **getCurExp(){**

**return** currentExp**;**

**}**

**public** **final** **int** **getAttackPoints(){**

**return** countAttackPoints**;**

**}**

**public** **final** **int** **getDefensePoints(){**

**return** countDefensePoints**;**

**}**

**public** **void** **giveExp(){**

        currentExp **+=** 15**;**

**}**

**public** **void** **levelUp(){**

        expForLevel **<<=** 1**;**

**++**level**;**

**}**

    @Override

**public** String **battleStep(boolean[]** enemyAttack**,**

**boolean[]** playerDefense**,**

**int** enemyAttackPoints**)**

**{**

        String battle\_log **=** "Ход бота:\n"**;**

        Random rand **=** **new** **Random();**

**int** at1 **=** **-**1**,** at2 **=** **-**1**;**

**int** def1 **=** **-**1**,** def2 **=** **-**1**;**

**boolean** at **=** **false,** def **=** **false;**

**for** **(int** i **=** 0**;** i **<** 6**;** **++**i**)**

**{**

**if** **(**enemyAttack**[**i**])**

**{**

**if** **(!**at**)**

**{**

                    at1 **=** i**;**

                    at **=** **true;**

**}**

**else**

                    at2 **=** i**;**

**}**

**if** **(**playerDefense**[**i**])**

**if** **(!**def**)**

**{**

                    def1 **=** i**;**

                    def **=** **true;**

**}**

**else**

                    def2 **=** i**;**

**}**

**int** damage **=** 0**;**

**if** **(**def1 **!=** **-**1 **&&** def2 **!=** **-**1 **&&** def1 **==** at1 **||** def2 **==** at1**)**

            battle\_log **+=** "Игрок парировал удар противника.\n"**;**

**else** **if(**at1 **!=** **-**1**)**

**{**

            damage **=** rand**.**nextInt**(**enemyAttackPoints**)** **-** rand**.**nextInt**(**defense**);**

**switch(**at1**){**

**case(**0**):**

                    damage **+=** rand**.**nextInt**(**5**);**

**break;**

**case(**1**):**

                    damage **+=** rand**.**nextInt**(**3**);**

**break;**

**case(**2**):**

                    damage **+=** rand**.**nextInt**(**2**);**

**break;**

**}**

**if** **(**damage **<** 0**)**

                damage **=** 0**;**

            currentHpParts**[**at1**]** **-=** damage**;**

            currentHp **-=** damage**;**

            battle\_log **+=** "Бот нанес игроку " **+** Integer**.**toString**(**damage**)** **+** " урона.\n"**;**

**if** **(**currentHpParts**[**at1**]** **<=** 0**)**

**{**

                battle\_log **+=** "Бот сломал игроку" **+** deadParts**(**at1**);**

                breakParts**[**0**][**at1**]** **=** **true;**

**if** **(**at1 **>** 1**)**

**{**

**if** **(**rand**.**nextInt**(**2**)** **==** 0**)**

**if** **(**countAttackPoints **!=** 0**)**

**--**countAttackPoints**;**

**else**

**--**countDefensePoints**;**

**else**

**if** **(**countDefensePoints **!=** 0**)**

**--**countDefensePoints**;**

**else**

**--**countAttackPoints**;**

**}**

**}**

**}**

**if** **(**def1 **!=** **-**1 **&&** def2 **!=** **-**1 **&&** def1 **==** at2 **||** def2 **==** at2**)**

            battle\_log **+=** "Игрок парировал удар противника.\n"**;**

**else** **if** **(**at2 **!=** **-**1**)**

**{**

            damage **=** rand**.**nextInt**(**enemyAttackPoints**)** **-** rand**.**nextInt**(**defense**);**

**switch(**at2**){**

**case(**0**):**

                    damage **+=** rand**.**nextInt**(**5**);**

**break;**

**case(**1**):**

                    damage **+=** rand**.**nextInt**(**3**);**

**break;**

**case(**2**):**

                    damage **+=** rand**.**nextInt**(**2**);**

**break;**

**}**

**if** **(**damage **<** 0**)**

                damage **=** 0**;**

            currentHpParts**[**at2**]** **-=** damage**;**

            currentHp **-=** damage**;**

            battle\_log **+=** "Бот нанес игроку " **+** Integer**.**toString**(**damage**)** **+** " урона.\n"**;**

**if** **(**currentHpParts**[**at2**]** **<=** 0**)**

**{**

                battle\_log **+=** "Бот сломал игроку" **+** deadParts**(**at2**);**

                breakParts**[**0**][**at2**]** **=** **true;**

**if** **(**at2 **>** 1**)**

**{**

**if** **(**rand**.**nextInt**(**2**)** **==** 0**)**

**if** **(**countAttackPoints **!=** 0**)**

**--**countAttackPoints**;**

**else**

**--**countDefensePoints**;**

**else**

**if** **(**countDefensePoints **!=** 0**)**

**--**countDefensePoints**;**

**else**

**--**countAttackPoints**;**

**}**

**}**

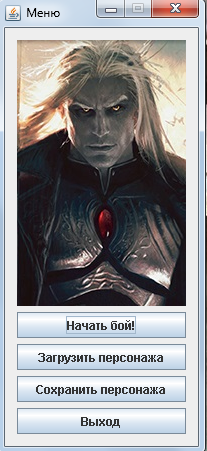
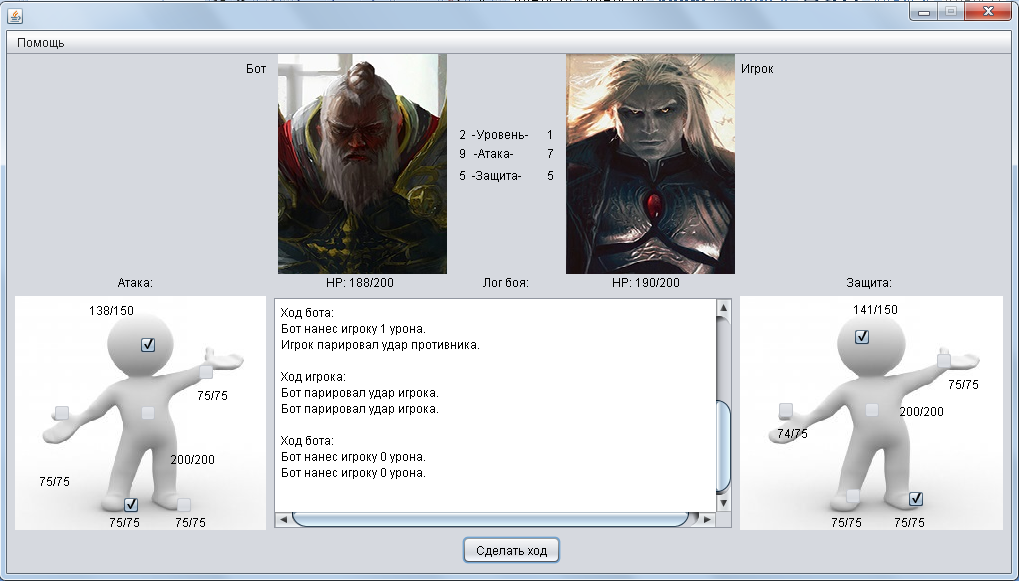
**}**

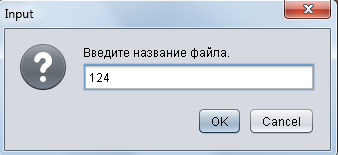
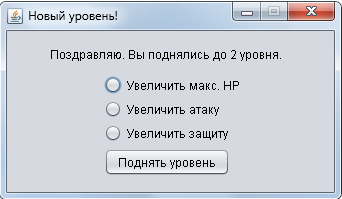
**return** battle\_log**;**

**}**

**}**

1. **Работа программы**

** **

** **