

Плюсы:

1. Организация проектов в виде классов
2. Отсутствие глобальных переменных
3. Отсутствие закомментированных участков кода
4. Код написан в одном стиле

Минусы:

1. Строки длиной больше 80 символов

```
if(Gamers[i].getStatus()==0)
{
    s="Житель "+QString::fromStdString(Gamers[i].getName())+" порололовал за "+QString::fromStdString(Gamers[vict[i]].
    ui->listWidget->addItem(s);
}
```

2. Объявление нескольких классов в одном файле

```

class Role
{
public:
    virtual int course(int time,int (*gamers)[N-1],int quantity,int gamer,int f)=0;
    virtual int getrole();
};
// Класс игрок
class Gamer
{
public:
    int course(int time,int (*gamers)[N-1],int quantity,int gamer,int f);
    void amendrole();
    void amendstatus(int time);
    Gamer(string N,int o,int l,int a,int c);
    Gamer();
    ~Gamer();
    string getName();
    int getoratory();
    int getluck();
    int getacting();
    int getconformality();
    int getstatus();
    int getrole();
private:
    string name;           // имя
    int oratory;           // красноречие
    int luck;              // удача
    int acting;            // актерские способности
    int conformality;      // конформность
    int status;            // статус в игре
    Role *role;            // роль в игре
};

```

3. Функции большого размера

```

bool fl=true;
int *victim;
int v, f;
int vict[16];
int gamers[16][N-1];
int max,imax,omax,iomax;
QString s;
victim=new int[Quantity];
for(int i=0;i<Quantity;i++)
{
    for(int j=0;j<N-1;j++)
        switch (j)
        {
            case 0:{
                gamers[i][0]=Gamers[i].getoratory();
                break;
            }
            case 1:{
                gamers[i][1]=Gamers[i].getluck();
                break;
            }
            case 2:{
                gamers[i][2]=Gamers[i].getacting();
                break;
            }
            case 3:{
                gamers[i][3]=Gamers[i].getconformality();
                break;
            }
            case 4:{
                gamers[i][4]=Gamers[i].getrole();
                break;
            }
            case 5:{
                gamers[i][5]=Gamers[i].getstatus();
                break;
            }
            default: break;
        }
}

```

4. Использование непонятных констант

```

pn.drawEllipse(ctr + QPointF(125.0 * cos(2.0 * pi * i / (float)Quantity), 125.0 * sin(2.0 * pi * i / (float)Quantity)), 15.
pn.setPen(QPen(QBrush(Qt::white), 1.0));
// и текст в нем (т. е. ключ)
pn.drawText(ctr + QPointF(125.0 * cos(2.0 * pi * i / (float)Quantity), 125.0 * sin(2.0 * pi * i / (float)Quantity)) + QPoint

```

5. Переменные с неясным значением

```

bool fl=true;
int *victim;
int v, f;
int vict[16];
int gamers[16][N-1];
int max,imax,omax,iomax;
QString s;

```

6. Название переменной с большой буквы

```

int Quantity;
int timed;

```

7. Использование генерируемых методов

```

ui->pushButton_2->setEnabled(false);

```

8. Кривое форматирование и использование одного куска кода без минимальных изменений

```

if(difference()==1)
{
    QMessageBox mbox;
    mbox.setText("Победила мафия");
    mbox.exec();
    ui->pushButton_2->setEnabled(false);
}
else
    if(difference()==2)
    {
        QMessageBox mbox;
        mbox.setText("Победили мирные жители");
        mbox.exec();
        ui->pushButton_2->setEnabled(false);
    }
}

```

9. Не закрыт поток ввода

```

void MainWindow::Load()
{
    int num, num2, num3, num4, num5;
    string s;
    ifstream inp("C:\\My Proj\\Practika\\Gamer.txt");
    if (inp.is_open())
    {
        if (inp >> num)
        {
            Quantity=num;
            for (int i = 0; i < num; i++)
            {
                if ((inp >> s)&&(inp >> num2) && (inp >> num3) && (inp >> num4) && (inp >> num5))
                {
                    if((num2>Min && num2<Max) && (num3>Min && num3<Max)&&(num4>Min && num4<Max)&&(num5>Min && num5<Max))
                    {
                        new( &Gamers[i])Gamer(s,num2,num3,num4,num5);
                    }
                    else
                    {
                        QMessageBox mbox;
                        mbox.setText("Ошибка. Неверная характеристика игрока.");
                        mbox.exec();
                        delete ui;
                    }
                }
                else return;
            }
        }
        else return;
    }
    else return;
}

```

10. Плохое наименование переменных, передаваемых в конструктор

```

Gamer::Gamer(string N,int o,int l,int a,int c)
{
    name=N;
    oratory=o;
    luck=l;
    acting=a;
    conformality=c;
    status=0;
    role=new Civilian;
}

```

11. Метод с неясным значением

```

int Civilian::getrole()
{
    return 0;
}

```

12. Нет запятых при перечисление

```

int Civilian::vote(int (*gamers)[N-1],int Quantity,int gamer,int f)
{

```

Оценка: 3 из 10