

Плюсы:

- +1 Использование объектно-ориентированного подхода
- +2 Информативное название полей (в большинстве случаев)
- +3 Отсутствие закомментированных участков кода

Минусы:

- 1 Описание всех классов в одном файле, а всех взаимодействий с ними в другом.
- 2 Логика приложения и взаимодействия с графическим интерфейсом не разделены.
- 3 Именованние методов, состоящих из 2х слов без применения регистрового либо символьного разделения.

```
public:
-- int course(int time,int (*gamers)[N-1],i
-- 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 способн
-- int conformality;---// Конформность
-- int status;-----// Статус в игре
-- Role *role;-----// Роль в игре
```

- 4 Отсутствие комментариев в теле методов

```
// Распределение ролей
void MainWindow::casting()
{
-- int qmaf;
-- int *quantm;
-- bool f;
-- qmaf=Quantity/3;
-- quantm=new int[qmaf];
-- srand(time(NULL));
-- quantm[0]=rand() % Quantity;
-- for(int i=0;i<qmaf;i++)
-- {
-- do{
-- quantm[i]=rand() % Quantity;
-- f=true;
-- for(int j=0;j<i;j++)
-- if(quantm[i]==quantm[j])
-- f=false;
-- }while(f==false);
-- }
-- for(int i=0;i<qmaf;i++)
-- Gamers[quantm[i]].amendrole();
}
```

- 5 Ширина кода достигает 160 и более символов, что усложняет чтение и восприятие.

```

for (int i = 0; i < Quantity; i++)
{
    QBrush br(QColor(205, 0, 0));
    QPen pen(QBrush(QColor(205, 0, 0)), 2.0);
    pn.setBrush(br);
    // КРУЖОК
    pn.drawEllipse(ctr + QPointF(125.0 * cos(2.0 * pi * i / (float)Quantity), 125.0 * sin(2.0 * pi * i / (float)Quantity)), 15.0, 15.0);
    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)) + QPointF(-4.0, 4.0), QString::number(i+1));
}

```

Итоговая оценка: 4/10