

Плюсы:

1. Проект организован в виде классов
2. Код читаемый и написан в одном определенном стиле
3. Используются стандартные классы Qt

Минусы:

1. Недостаток комментариев
2. Константы прописаны прямо в функциях

```
        break;
    }
    case 2:
    {
        // сброс посуды
        pnt->fillRect(QRectF(geom.x() + geom.width() * 0.2, geom.y() + geom.height() * 0.2, geom.width() * 0.6, geom.height() * 0.6), QBrush(Qt::white));
        break;
    }
    case 3:
    {
        // раковина
        pnt->fillRect(QRectF(geom.x() + geom.width() * 0.2, geom.y() + geom.height() * 0.2, geom.width() * 0.6, geom.height() * 0.6), QBrush(Qt::white));
        break;
    }
    case 4:
    {
        // плита
        pnt->setBrush(QBrush(QColor(128, 128, 128)));
        pnt->drawEllipse(QPointF(geom.x() + geom.width() * 0.25, geom.y() + geom.height() * 0.25), geom.width() * 0.5, geom.height() * 0.5);
        pnt->drawEllipse(QPointF(geom.x() + geom.width() * 0.25, geom.y() + geom.height() * 0.75), geom.width() * 0.5, geom.height() * 0.5);
        pnt->drawEllipse(QPointF(geom.x() + geom.width() * 0.75, geom.y() + geom.height() * 0.25), geom.width() * 0.5, geom.height() * 0.5);
        pnt->drawEllipse(QPointF(geom.x() + geom.width() * 0.75, geom.y() + geom.height() * 0.75), geom.width() * 0.5, geom.height() * 0.5);
        break;
    }
    case 5:
    {
        // витрина
        pnt->fillRect(QRectF(geom.x(), geom.y() + geom.height() * 0.1, geom.width(), geom.height() * 0.8), QBrush(Qt::white));
        break;
    }
}
```

3. Излишние фигурные скобки (пример выше)
4. Множество стандартно сгенерированных методов и названий переменных/методов

```
void DialogDish::on_btnOK_clicked()
{
    if (ui->editName->text().isEmpty())
    {
        return;
    }
    if (editMode)
        dish->replaceData(ui->editName->text());
    else
        dish = new Dish(ui->editName->text());
    accept();
}

void DialogDish::on_btnCancel_clicked()
{
    reject();
}
```

5. Пустые функции

```
void Person::Act()  
{  
  
}  

```

6. Слишком большие и нагроможденные методы (свитчи внутри свитчей)

```
void Employee::Act()  
{  
    do  
    {  
        switch (role)  
        {  
            case 0:  
            {  
                switch (actionStep)  
                {  
                    case 0:  
                    {  
                        if (personToInteract == 0)  
                        {  
                            if (!dynamic_cast<Person*>(&personToInteract))  
                                return;  
                        }  
                        this->thread()->sendMessage(personToInteract);  
                        break;  
                    }  
                    default:  
                        break;  
                }  
                break;  
            }  
            case 1:  
            {  
                switch (actionStep)  
                {  
                    case 0:  
                    {  
                        do  

```

Финальная оценка 6 из 10