

## Плюсы:

- Наличие организации проекта в виде классов.
- Код написан единообразно, в едином стиле
- Наименования классов, полей, методов, параметров - «говорящие»
- Не используются краткие названия полей, переменных, методов
- Не используются глобальные переменные
- Отсутствуют методы большого размера
- Не используется множественное наследование


## Минусы:

- Отсутствуют комментарии к реализации методов
- Много закомментированных участков кода, например в Program.cs

```
43     private static void Tr_MessageSticker(object sendr, MessageSticker<User, Chat> e)
44     {
45         //Console.WriteLine("ID сообщения:{0}\nID отправителя:{1}\nНик отправителя:{2}\nИмя:{3} Фамилия:{4}\nДата:{5}\nШирина стикера:{6}\nВысота стикера:{7}\nЭмодзи:{8}\nТекст стикера:{9}\n",
46         //    e.message_id, e.from.id, e.from.username, e.from.first_name, e.from.last_name, e.date, e.width, e.height, e.emoji, e.thun
47         //Console.WriteLine();
48     }
49     private static void Tr_MessagePhoto(object sendr, MessagePhoto<User, Chat> e)
50     {
51         //Console.WriteLine("ID сообщения:{0}\nID отправителя:{1}\nНик отправителя:{2}\nИмя:{3} Фамилия:{4}\nДата:{5}\nКомментарий к фс:{6}\n",
52         //    e.message_id, e.from.id, e.from.username, e.from.first_name, e.from.last_name, e.date, e.caption);
53         //for (int i = 0; i < e.photo.Count; i++)
54         //{
55             //    Console.WriteLine("Фото №{0}", i + 1);
56             //    Console.WriteLine("ID Файла:{0}\nРазмер файла:{1}байт\nШирина:{2} Высота:{3}\n",
57             //        e.photo[i].file_id, e.photo[i].file_size, e.photo[i].width, e.photo[i].height);
58         //}
59         //MethodBot m = new MethodBot(Settings.Default.Token);
60         //m.SendMessage("Вы нам прислали фотографию", 243746390);
61         //m.SendPhotoLink(243746390, e.photo[e.photo.Count - 1].file_id, e.caption);
62     }
63     private static void Tr_MessageVideo(object sendr, MessageVideo<User, Chat> e)
64     {
65         //Console.WriteLine("ID сообщения:{0}\nID отправителя:{1}\nНик отправителя:{2}\nИмя:{3} Фамилия:{4}\nДата:{5}\nID видео:{6}\nНШВ:{7}\n",
66         //    e.message_id, e.from.id, e.from.username, e.from.first_name, e.from.last_name, e.date, e.file_id, e.width, e.height, e.c
67         //Console.WriteLine();
68     }
```

- Код не очень удобочитаем, встречаются очень длинные строчки кода, приходится использовать горизонтальную прокрутку

```
private static void Tr_MessageText(object sendr, MessageText<User, Chat> e)
{
    Console.WriteLine("ID сообщения:{0}\nID отправителя:{1}\nНик отправителя:{2}\nИмя:{3} Фамилия:{4}\nДата:{5}\nТекст сообщения:{6}\n",
        e.DefaultMessageInfo.MessageId, e.DefaultMessageInfo.From.Id, e.DefaultMessageInfo.From.Username,
        e.DefaultMessageInfo.From.First_name, e.DefaultMessageInfo.From.Last_name,
        e.DefaultMessageInfo.DateReveiver, e.Text);
}
```



- Встречаются «магические» константы, например в SimpleJSON.cs

```
769         public override string ToJSON(int prefix)
770     {
771         string s = new string(' ', (prefix + 1) * 2);
772         string ret = "[";
773         foreach (JSONNode n in m_List)
774         {
775             if (ret.Length > 3)
776                 ret += ", ";
777             ret += "\n" + s;
778             ret += n.ToJSON(prefix + 1);
779         }
780     }
```

Оценка: 7/10