

## ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ ПАРАДИГМА КАК МЕТОД ФОРМАЛИЗАЦИИ СВОЙСТВ ПЕРСОНАЖЕЙ В ВЫМЫШЛЕННЫХ ВСЕЛЕННЫХ (НА ПРИМЕРЕ ВОПЛОЩЕНИЙ ОБРАЗА НЕХЛЮДОВА ИЗ РОМАНА Л. Н. ТОЛСТОГО «ВОСКРЕСЕНИЕ»)

Григорий Скоморовский

Институт языков и культур Балтийского региона

Вильнюсский университет, Литва

E-mail: [grigory.skomorovskiy@flf.stud.vu.lt](mailto:grigory.skomorovskiy@flf.stud.vu.lt)

<https://orcid.org/0009-0008-3044-0601>

**Резюме.** В статье рассмотрена возможность применения объектно-ориентированной парадигмы (термин компьютерной науки) для формализации различных имплементаций персонажа в вымышленных вселенных. Приведены примеры на основе воплощений образа Нехлюдова из романа Л. Н. Толстого «Воскресение». Рассмотрены возможные приложения обсуждаемого подхода для анализа взаимосвязей различных имплементаций одного и того же персонажа, а также для генерации новых персонажей на основе индивидуальных предпочтений читателя (зрителя, пользователя).

**Ключевые слова:** вымышленная вселенная, формализация персонажей, объектно-ориентированная парадигма, экранизация, роман «Воскресение».

## OBJECT-ORIENTED PARADIGM AS A METHOD FOR FORMALIZING CHARACTER PROPERTIES IN FICTIONAL UNIVERSES (ON THE EXAMPLE OF THE CHARACTER REPRESENTATION OF NEKHLYDOV IN LEO TOLSTOY'S NOVEL *RESURRECTION*)

Grigory Skomorovskiy

Institute for the Languages and Cultures of the Baltic

Vilnius University, Lithuania

**Abstract.** The article explores the possibility of applying the object-oriented paradigm (a term sourced from computer science) to formalize various implementations of characters in fictional universes. Examples are provided based on the implementations of the character of Nekhlyudov from Tolstoy's novel *Resurrection*. Possible applications of the discussed approach are examined for analyzing the relationships between different implementations of the same character, as well as for generating new characters based on the individual preferences of the reader (or viewer, user).

**Keywords:** fictional universe, character formalization, object-oriented paradigm, adaptation, novel *Resurrection*, cinematographic adaptation.

## Введение

Внедрение компьютерных методов происходит повсеместно, в том числе в гуманитарные области исследований (*Digital Humanities*). Существующие работы по цифровой филологии фокусируются, в основном, на компьютерной лингвистике, статистических методах работы с корпусами текстов, цифровизации источников, реже – на представлении знаний в форме семантических сетей (Гомонова, 2020; Berti, 2019). В то же время потенциал использования компьютерных методов очевидно шире: несмотря на то, что программирование в его современном виде постепенно уходит в прошлое (Welsh, 2023), в информатике накоплено множество подходов для формального представления связанных между собой объектов. В качестве таких объектов могут выступать, в том числе, литературные персонажи. Настоящая статья фокусируется на возможности применения объектно-ориентированной парадигмы для формализации (литературных) персонажей «вымышленных вселенных», имеющих множество «воплощений» и, как следствие, обнаруживающих такое явление, как «наследование свойств». Отметим, что данная статья не претендует на всеобъемлющий анализ толстовского образа Нехлюдова, но демонстрирует возможности применения определенной методологии программирования для анализа корпуса всевозможных имплементаций (литературного) образа.

## Персонаж. Сеть. Вселенная

Персонаж (в самом широком смысле) «существует» в различных имплементациях<sup>1</sup>: от представления в сознании автора, через образ, запечатленный в тексте и отраженный затем в сознании читателя, и далее – в отдельных случаях – к сценическому, экранному или даже игровому воплощению<sup>2</sup>. Совокупность всех таких воплощений составляет *корпус всевозможных имплементаций образа*, мощность (объем) которого практически неограничен за счет несчетного множества читателей (зрителей, пользователей). При этом определенный персонаж

- 1 Под имплементацией здесь и далее понимается *реализация, воплощение*. Данный термин используется в компьютерной науке для обозначения очередной реализации некоторой одной сущности.
- 2 Например, персонаж «*Сталкера*» братьев Стругацких: от повести «Пикник на обочине» до серии компьютерных игр «S.T.A.L.K.E.R.».

становится связующим звеном, объединяя произведения разных видов и жанров в своеобразную «сеть». Такая «сеть», или «объединенная сеть фильмов, телевизионных программ, комиксов, игр и интернет-контента»<sup>3</sup> (Navard et al., 2019, p. 10) в медийном дискурсе все чаще фигурирует как «вымышленная вселенная» (*fictional universe*) или просто «вселенная». Впервые термин «вселенная» в обсуждаемом контексте ввел, по-видимому, Д. Маркштейн<sup>4</sup>, большой энтузиаст и историк комиксов и анимации. Сегодня вымышленные вселенные как системы все чаще становятся объектами исследований. Однако работ, посвященных именно персонажам вселенных, не так много, что связано, по мнению редакторов сборника «Персонажи вымышленных вселенных», с «разрывом между дисциплинами, [а также с тем, что] при более внимательном рассмотрении персонажи оказываются очень сложными объектами во многих отношениях» (Eder et al., 2010, p. 3).

Можно допустить, что сложность персонажа связана не только, например, с драматургическими аспектами, но также с тем, что образ имеет разную природу (медиатекст / ментальное представление в сознании), индивидуален и подвержен влиянию множества факторов. Рассмотрим в качестве примера образ Нехлюдова из романа Льва Толстого «Воскресение», который породил одну из первых «вселенных»<sup>5</sup>. Образ Нехлюдова интересен тем, что автор весьма скупо описывает его

- 3 Предположим, что в этот список можно добавить также театральные постановки, существующие в медиапространстве в виде записи (видеoverсии) или прямого эфира (стрима).
- 4 Имеется в виду текст Дональда Маркштейна из «фанзина» (*fanzine*) 1970 г., который издавался энтузиастами из ассоциации любительской прессы CAPA-alpha, объединявшей любителей комиксов. В архиве онлайн-энциклопедии Don Markstein's Toonopedia сохранился след этого текста: <https://www.toonopedia.com/universe.htm>, который начинается с фразы (перевод мой по смыслу) «Прослеживать “вселенные” персонажей комиксов – увлекательное занятие. Это делается путем построения сети пересечений, начиная с основного персонажа. Далее проводится наблюдение: сколько еще персонажей сосуществуют с ним в одном мире». Утверждение об авторстве термина может оказаться справедливым, если учесть характер деятельности Маркштейна как «собираателя» и исследователя персонажей, сюжетов и интертекстуальных пересечений из различных комиксов и мультфильмов.
- 5 Роман «Воскресение» стал основой театральных и оперных постановок, более чем 30 экранизаций и фильмов «по мотивам», большинство из которых создано за пределами России: от немой короткометражки «En Opstandelse» 1907 г. датчанина Вигго Ларсена до итальянского «Resurrezione» 2019 г. режиссера Тонино Де Бернарди.

внешность (буквально несколько слов в третьей главе первой части романа), и читателю остается только *догадываться* о том, *как мог выглядеть, каким мог быть* такой герой, как Нехлюдов.

Как известно, существовал реальный «прототип» Нехлюдова, о котором Толстому рассказал А. Ф. Кони. Итак, в сознании Кони «существовала» первая имплементация обсуждаемого образа (если не считать реальности «прототипа»). Можно предположить, что в сознании Толстого-слушателя было сформировано свое представление о герое (устного) рассказа Кони, *повлиявшее* на тот авторский, синтетический образ Нехлюдова, который был постепенно «рожден» в сознании Толстого-писателя и воплощен затем в тексте романа. В свою очередь, в сознании каждого читателя романа «Воскресения» «рисует» свой Нехлюдов – образ, формирующийся под влиянием как текста (например, из описания внешности Нехлюдова, из последовательности его поступков и реакций на события), так и личных представлений читателя о том *каким может быть* русский дворянин (как некоторый архетип) той эпохи, *каким может быть* литературный герой Толстого и т.д.<sup>6</sup> Иллюстраторы многочисленных изданий «Воскресения» имели свои собственные представления. У сценаристов, режиссеров, операторов, гримеров, актеров многочисленных экранизаций «Воскресения» также были свои внутренние репрезентации образа Нехлюдова, которые, в свою очередь, оказали влияние на соответствующие экранные воплощения.

Брайан Ричардсон в своем разностороннем исследовании «транстекстуальных персонажей» в вымышленных вселенных, опираясь на критерий Роланда Харвега, отмечает, что «информация из более поздних произведений *дополнит наши знания о персонаже*» (Richardson, 2010, с. 540. Курсив наш. – Г.С.). Можно уточнить, что это «дополнение» есть некоторое *наследование* свойств персонажа, и мы имеем дело со своеобразной «сетью», в которой каждая последующая имплементация образа «наследует» свойства предыдущей, испытывая при этом влияние многочисленных «личных» представлений (предубеждений, стереотипов) читателя или зрителя. Итоговый образ (будь то ментальный образ у читателя или экранный образ) обладает некоторыми суммарными свойствами, которые лишь частично «наследуют» свойства своего литературного «предка» в их первоизданном виде.

6 Вплоть до того, как, в конце концов, выглядит человек.

Формализовать такую сеть – чтобы иметь возможность, например, проследить взаимосвязи между имплементациями образа, личными представлениями и другими влияющими на образ факторами – можно как графически, в виде схемы, так и с помощью объектно-ориентированной парадигмы в форме декларирования данных на языке программирования высокого уровня. Второй способ может быть более продуктивен, так как его применению сопутствует процесс дигитализации, а результат компьютеризирован и может быть использован в компьютерно-ориентированных приложениях<sup>7</sup>.

## ООП и образ Нехлюдова

Объектно-ориентированная парадигма (или объектно-ориентированное программирование; далее ООП) – известная в компьютерной науке методология представления данных в виде классов и объектов, основанная на трех базовых принципах: наследование (наследование функциональности класса-предка или объекта-предка), инкапсуляция (объединение данных и программного кода, изоляция «собственных» данных класса от использования извне) и полиморфизм (способность той или иной функции получать в качестве аргументов данные разных типов). Общеизвестно, что ООП удобна для представления иерархических зависимостей и сущностей из разнообразных предметных областей. При этом существует два подхода к паре понятий класс-объект (Gunter, Mitchell, 1994): первый, реализованный в большинстве современных языков программирования (*class-based* языки) полагает класс как образец (прототип) для клонирования (порождения) объектов (при этом класс, выступая как абстракция, как своеобразная модель, может быть наследником другого класса или классов<sup>8</sup>); второй подход, на котором базируются *delegation-based* языки, допускает создание одного объекта путем формирования другого объекта, имитируя «наследование».

Для представления образа в такой системе, как *корпус всевозможных имплементаций образа* в его подмножестве – в «вымышленной

7 Например, в производстве компьютерных игр на основе того или иного персонажа: в играх, как и в 3D-моделировании, персонаж описывается на соответствующем языке программирования именно как объект с множеством *свойств*. См., напр.: Максименкова, Веселко, 2022.

8 Если класс является наследником более одного класса, говорят о «множественном наследовании».

вселенной», интересен второй (*delegation-based*) подход<sup>9</sup>. При этом мы выводим за скобки понятие *класс*, а под понятием *объект* подразумеваем как сам образ (литературный, ментальный, экранный), так и различные личные стереотипы читателя (зрителя). Таким образом, мы унифицируем *образ* и *личные стереотипы*, что позволяет описать влияние одного на другое как своеобразное *наследование* свойств от одного объекта к другому. Так, например, представление образа Нехлюдова читателем является *объектом*, который «наследует» свойства<sup>10</sup>, как минимум, двух других *объектов*: того образа, который описан в романе, а также стереотипного (личного) представления читателем образа русского дворянина XIX в. На объектно-ориентированном языке программирования последний тезис может быть записан следующим образом (в общем виде):

```
Объект «Ментальный образ участника реальных событий, свидетелем которых был А. Ф. Кони, в сознании самого Кони» {
    . . .
};
```

```
Объект «Герой устного рассказа А. Ф. Кони»
есть «наследник» объекта «Ментальный образ участника реальных событий, свидетелем которых был А. Ф. Кони, в сознании самого Кони» {
    . . .
};
```

```
Объект «Ментальный образ героя устного рассказа А. Ф. Кони в сознании Толстого»
есть «наследник» объекта «Герой устного рассказа А. Ф. Кони» {
    . . .
};
```

```
Объект «Нехлюдов как ментальный образ в сознании Толстого как автора романа»
есть «наследник» объекта «Ментальный образ героя устного рассказа А. Ф. Кони в сознании Толстого» {
    . . .
};
```

- 9 *Delegation-based* подход используется здесь для теоретического представления имплементаций образа – для предварительного анализа. Для практического компьютерного эксперимента автором статьи был использован *class-based* подход и язык программирования высокого уровня Java.
- 10 В данном контексте *имитирует наследование* – формируется за счет других объектов, используя их свойства.

**Объект** «Нехлюдов как литературный образ, описанный в тексте Толстого»

**есть «наследник» объекта** «Нехлюдов как ментальный образ в сознании Толстого как автора романа» {

. . .  
};

**Объект** «Стереотип внешности человека у читателя X» {

. . .  
};

**Объект** «Стереотип русского дворянина у читателя X» {

. . .  
};

**Объект** «Образ Нехлюдова у читателя X»

**есть «наследник» объекта** «Стереотип внешности человека у читателя X»,

**есть «наследник» объекта** «Стереотип русского дворянина у читателя X», **есть «наследник» объекта** «Нехлюдов как литературный образ, описанный в тексте Толстого» {

. . .  
};

Приведенная последовательность деклараций представляет собой лишь «верхний» слой абстракции, отражает цепочку наследования (влияния), но не содержит деталей – конкретной информации о том или ином свойстве. Для более детальной формализации на данном этапе предлагается пойти на некоторое *упрощение*, чтобы посмотреть, как рассматриваемый подход работает для частного массива свойств, описывающих *внешность* персонажа, оставляя за скобками более общие и сложные массивы, связанные с оценочными суждениями и эмоциональным отношением читателя (зрителя) к персонажу.

## ООП и внешность Нехлюдова

Толстой описывает внешность Нехлюдова очень кратко:

*[...] спустив с кровати гладкие белые ноги, нашел ими туфли, накинул на полные плечи шелковый халат [...]. Вымыл душистым мылом руки, старательно вычистив щетками отпущенные ногти и обмыл у*

*большого мраморного умывальника себе лицо и толстую шею, он пошел еще в третью комнату у спальни, где приготовлен был душ. Обмыв там холодной водой мускулистое, обложившееся жиром белое тело и вытершись лохматой простыней, он надел чистое выглаженное белье, как зеркало, вычищенные ботинки и сел перед туалетом расчесывать двумя щетками небольшую курчавую бороду и поредевшие на передней части головы вьющиеся волосы. (Толстой, 1996, часть I, глава 3).*

Формализуя *внешность героя* с использованием ООП, можно записать следующее:

**Объект** «Нехлюдов как литературный образ, описанный в тексте Толстого»  
**есть «наследник» объекта** «Нехлюдов как ментальный образ в сознании Толстого как автора романа» {

Ноги: «гладкие, белые»;

Плечи: «полные»;

Ногти: «отпущенные»;

Шея: «толстая»;

Тело: «мускулистое, обложившееся жиром, белое»;

Борода: «небольшая, черная, курчавая»;

Волосы: «поредевшие на передней части головы, вьющиеся»;

};

Очевидно, что имеющихся данных о внешности недостаточно: из текста мы не знаем, например, цвета глаз и «вьющихся» волос Нехлюдова, формы его лица, размера рук, роста и т.д. Поэтому читатель вынужден «домысливать» Нехлюдова, основываясь на собственных представлениях (стереотипах) о том, как выглядит человек и как может выглядеть дворянин (второй половины) XIX в. Таким образом, параметров (или полей) в соответствующем объекте<sup>11</sup> гораздо больше, и они «заполняются» значениями, которые наследуются из собственных стереотипов:

**Объект** «Образ Нехлюдова у читателя X»  
**есть «наследник» объекта** «Стереотип внешности человека у

11 Строго говоря, можно выделить отдельный класс (*интерфейс*), который является моделью человека в самом общем смысле и уже содержит соответствующие поля, которые «заполняются» при очередной итерации наследования.



читателя X»,  
**есть «наследник» объекта «Стереотип русского дворянина у читателя X», есть «наследник» объекта «Нехлюдов как литературный образ, описанный в тексте Толстого» {**

Рост: [наследуемое значение];  
Форма лица: [наследуемое значение];  
Цвет глаз: [наследуемое значение];  
Руки: [наследуемое значение];  
Ноги: «гладкие, белые»;  
Плечи: «полные»;  
Ногти: «отпущенные»;  
Шея: «толстая»;  
Тело: «мускулистое, обложившееся жиром, белое»;  
Борода: «небольшая, черная, курчавая»;  
Волосы: «поредевшие на передней части головы, вьющиеся»;  
Цвет волос: [наследуемое значение];  
. . .  
};

В реальной декларации объекта каждый параметр (в данном случае – внешности) должен иметь свой «тип», который является набором (множеством) точно определенных значений – примитивов. Введем, например, тип «ЦветГлаз» и определим его следующим образом:

**Тип «ЦветГлаз» = {Синие, Голубые, Серые, Зеленые, Желтые, Карие, Черные};**

Таким образом, например, значение «ЦветГлаз.Синие» является *примитивом*, который индивидуален для каждого читателя (зрителя, пользователя). Очевидно, что каждый примитив связан с определенным опытом и сформированными стереотипами. Но каким образом значение в объекте-потомке замещается конкретным примитивом? Этот вопрос особенно актуален в случае множественного наследования: так, например, индивид 1) сначала прочел роман; в котором 2) увидел иллюстрацию; а затем 3) посмотрел экранизацию. Какой теперь цвет глаз будет иметь этот «суммарный» Нехлюдов в сознании индивида? Как работает *предпочтительный выбор* нового значения? Механизм такого выбора строго индивидуален. Назовем этот механизм *индивидуальной функцией замещения*. Определить такую функцию можно только

опытным путем, построив максимально полную «сеть» из объектов (образов и стереотипов) для конкретного индивида (читателя, зрителя, пользователя).

## Выводы

Автором статьи предпринята практическая попытка формализовать с помощью ООП внешность Нехлюдова для «вселенной», сформированной на основе непосредственно романа и одной черно-белой экранизации. Эксперимент выполнялся на языке программирования Java с использованием интерфейсов (имитирующих множественное наследование) и объектов-синглтонов<sup>12</sup>. В то же время, описанный метод может быть использован не только для анализа взаимосвязей различных имплементаций персонажа и исследования соответствующих «маршрутов» формирования образа. Зная, например, как работает функция замещения для конкретного индивида, можно выявить необходимую информацию для генерирования новых персонажей на основе индивидуальных предпочтений. Иными словами, имея информацию (например, из big data), что конкретный читатель (зритель, пользователь) предпочитает замещать неизвестный цвет глаз, например, карими глазами, можно использовать эти данные для создания новой «версии» персонажа в режиме индивидуального медиа (когда контент адаптируется к конкретному пользователю). А зная дополнительно, что индивид имеет положительное отношение к персонажам с голубыми глазами, можно генерировать – наоборот – отрицательного героя с голубыми глазами, чтобы временно скрыть его негативные качества. Предположительно, такие приемы могут стать особенно актуальными в наступающую эпоху искусственного интеллекта и индивидуальных медиа.

## Литература

- Гомонова, И., Серикова, И., 2020. *Компьютерная филология: практическое руководство*. Гомель: ГГУ им. Ф. Скорины.
- Максименкова, О., Веселко, Н., 2022. *Программирование в Unreal Engine 5 для начинающего игродела. Основы визуального языка Blueprint*. Москва: Эксмо.

12 Синглтон – один из паттернов проектирования, термин из программной инженерии. См., напр.: Gamma et al., 1994.

- Толстой, Л.Н., 1996. Воскресение. *Собрание сочинений в 8 т.* Москва: Лексика. Т. 6. Режим доступа: <https://ilibrary.ru/text/1462/index.html> [см. 25 12 2023].
- Berti, M., ed., 2019. *Digital Classical Philology. Ancient Greek and Latin in the Digital Revolution*. Berlin/Boston: Walter de Gruyter GmbH.
- Eder, J., Jannidis, F. and Schneider, R., 2010. Characters in Fictional Worlds. An Introduction. In: Eder, J., Jannidis, F. and Schneider, R., eds. *Characters in fictional worlds: understanding imaginary beings in literature, film, and other media*. Berlin/New York: Walter de Gruyter, pp. 3–64.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. and Booch, G., 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. Westford, MA: Addison-Wesley Professional.
- Gunter, C. and Mitchell, J., eds., 1994. *Theoretical Aspects of Object-Oriented Programming: Types, Semantics, and Language Design*. Cambridge, MA: The MIT Press.
- Havard, C., Fuller, R., Ryan, T. and Grieve, F., 2019. Using the Marvel Cinematic Universe to Build a Defined Research Line. *Transformative Works & Cultures*. Vol 30, p. 10. Available at: <https://journal.transformativeworks.org/index.php/twc/article/view/1837/2195> [Accessed 25 December 2023].
- Richardson, B., 2010. Transtextual Characters. In: Eder, J., Jannidis, F. and Schneider, R., eds. *Characters in fictional worlds: understanding imaginary beings in literature, film, and other media*. Berlin/New York: Walter de Gruyter, pp. 527–541.
- Welsh, M., 2023. The End of Programming. *Communications of the ACM*. Vol. 66. No. 1, pp. 34–35.

## References

- Berti, M., ed., 2019. *Digital Classical Philology. Ancient Greek and Latin in the Digital Revolution*. Berlin/Boston: Walter de Gruyter GmbH.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. and Booch, G., 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. Westford, MA: Addison-Wesley Professional.
- Gomonova, I. and Serikova, I., 2020. *Komp'yuternaya filologiya: prakticheskoe rukovodstvo*. Gomel': GGU im. F. Skoriny.
- Gunter, C. and Mitchell, J., eds., 1994. *Theoretical Aspects of Object-Oriented Programming: Types, Semantics, and Language Design*. Cambridge, MA: The MIT Press.
- Havard, C., Fuller, R., Ryan, T. and Grieve, F., 2019. Using the Marvel Cinematic Universe to Build a Defined Research Line. *Transformative Works & Cultures*. Vol 30, p. 10. Available at: <https://journal.transformativeworks.org/index.php/twc/article/view/1837/2195> [Accessed 25 December 2023].
- Eder, J., Jannidis, F. and Schneider, R., 2010. Characters in Fictional Worlds. An

- Introduction. In: Eder, J., Jannidis, F. and Schneider, R., eds. *Characters in fictional worlds: understanding imaginary beings in literature, film, and other media*. Berlin/New York: Walter de Gruyter, pp. 3–64.
- Maksimenkova, O. and Veselko, N., 2022. *Programmirovaniye v Unreal Engine 5 dlya nachinayushchego igrodela. Osnovy vizual'nogo yazyka Blueprint*. Moskva: Eksmo.
- Richardson, B., 2010. Transtextual Characters. In: Eder, J., Jannidis, F. and Schneider, R., eds. *Characters in fictional worlds: understanding imaginary beings in literature, film, and other media*. Berlin/New York: Walter de Gruyter, pp. 527–541.
- Tolstoi, L.N., 1996. Voskresenie. In: *Sobranie sochinenii v 8 t.* Moskva: Leksika. Vol. 6. <https://ilibrary.ru/text/1462/index.html> [sm. 25 12 2023].
- Welsh, M., 2023. The End of Programming. *Communications of the ACM*. Vol. 66. No. 1, pp. 34–35.