

Федеральное государственное автономное образовательное учреждение
высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа по программированию №3-4

Вариант 31012

Выполнил

Пчелкин Илья Игоревич

Р3106

Проверил

Вербовой А. А.

Санкт-Петербург 2024

ОГЛАВЛЕНИЕ

<i>Задание</i>	3
<i>UML – Диаграмма</i>	5
<i>Код программы.....</i>	6
<i>Результат работы программы.....</i>	7

Задание

В соответствии с выданным вариантом на основе предложенного текстового отрывка из литературного произведения создать объектную модель реального или воображаемого мира, описываемого данным текстом. Должны быть выделены основные персонажи и предметы со свойственным им состоянием и поведением. На основе модели написать программу на языке Java.

Описание предметной области, по которой должна быть построена объектная модель:

Вместо того чтобы идти играть в городки или футбол, Незнайка садился за стол и принимался за чтение. Читал он каждый день по страничке, но и от этого была, конечно, большая польза. Иногда он читал даже по две странички: за сегодняшний день и за завтрашний. Покончив с чтением, он брал тетрадочку и начинал писать. Писал он уже не печатными буквами, а письменными, но сначала они получались у него не очень красиво. Первое время у него в тетради вместо букв выходили какие-то несообразные кривульки и кренделя, но Незнайка очень старался и постепенно выучился писать красивые буквы, и большие, то есть заглавные, и маленькие. Гораздо хуже у него обстояло дело с кляксами. Незнайка часто сажал кляксы в тетради. И к тому же как только посадит кляксу, так сейчас же слизнет ее языком. От этого кляксы у него получались с длинными хвостами. Такие хвостатые кляксы Незнайка называл кометами. Эти "кометы" были у него чуть ли не на каждой страничке. Но Незнайка не унывал, так как знал, что терпение и труд помогут ему избавиться и от "комет".

Этапы выполнения работы:

1. Получить вариант
2. Нарисовать UML-диаграмму, представляющую классы и интерфейсы объектной модели и их взаимосвязи;
3. Придумать сценарий, содержащий действия персонажей, аналогичные приведенным в исходном тексте;
4. Согласовать диаграмму классов и сценарий с преподавателем;
5. Написать программу на языке Java, реализующую разработанные объектную модель и сценарий взаимодействия и изменения состояния объектов. При запуске программа должна проигрывать сценарий и выводить в стандартный вывод текст, отражающий изменение состояния объектов, приблизительно напоминающий исходный текст полученного отрывка.
6. Продемонстрировать выполнение программы на сервере helios.
7. Ответить на контрольные вопросы и выполнить дополнительное задание.

Текст, выводимый в результате выполнения программы не обязан дословно повторять текст, полученный в исходном задании. Также не обязательно реализовывать грамматическое согласование форм и падежей слов выводимого текста.

Стоит отметить, что цель разработки объектной модели состоит не в выводе текста, а в эмуляции объектов предметной области, а именно их состояния (поля) и поведения (методы). Методы в разработанных классах должны изменять состояние объектов, а выводимый текст должен являться побочным эффектом, отражающим эти изменения.

Требования к объектной модели, сценарию и программе:

1. В модели должны быть представлены основные персонажи и предметы, описанные в исходном тексте. Они должны иметь необходимые атрибуты и характеристики (состояние) и уметь выполнять свойственные им действия (поведение), а также должны образовывать корректную иерархию наследования классов.
2. Объектная модель должна реализовывать основные принципы ООП - инкапсуляцию, наследование и полиморфизм. Модель должна соответствовать принципам SOLID, быть расширяемой без глобального изменения структуры модели.
3. Сценарий должен быть вариативным, то есть при изменении начальных характеристик персонажей, предметов или окружающей среды, их действия могут изменяться и отклоняться от базового сценария, приведенного в исходном тексте. Кроме того, сценарий должен поддерживать элементы случайности (при генерации персонажей, при задании исходного состояния, при выполнении методов).
4. Объектная модель должна содержать как минимум один корректно использованный элемент каждого типа из списка:
 - абстрактный класс как минимум с одним абстрактным методом;
 - интерфейс;
 - перечисление (enum);
 - запись (record);
 - массив или ArrayList для хранения однотипных объектов;
 - проверяемое исключение.
5. В созданных классах основных персонажей и предметов должны быть корректно переопределены методы equals(), hashCode() и toString(). Для классов-исключений необходимо переопределить метод getMessage().
6. Созданные в программе классы-исключения должны быть использованы и обработаны. Кроме того, должно быть использовано и обработано хотя бы одно unchecked исключение (можно свое, можно из стандартной библиотеки).
7. При необходимости можно добавить внутренние, локальные и анонимные классы.

[illegible]

Код программы

<https://github.com/ipka23/ITMO/tree/main/Labs/Prog/Lab3/lab3>

Результат работы программы

Незнайка не играет в футбол

Незнайка не играет в городки

Незнайка сел за стол, и начал читать.

Читал он каждый день по страничке, но сегодня прочитал две и от этого была, очень большая польза.

Покончив с чтением, Незнайка начинал писать в тетрабочка.

Писал Незнайка не письменная буква, а некрасивая письменная буква.

Первое время у Незнайка выходили кривулька и крендель, но Незнайка очень старался и постепенно выучился писать заглавная красивая и маленькая красивая буквы.

Незнайка не поставил кляксу на странице № 2.

Незнайка поставил кляксу на странице № 1.

Незнайка слизал комета языком, из-за этого клякса получилась с длинным хвостом.

Кляксы с длинным хвостом он называл комета.

Такие кометы были у него почти на каждой странице.

У Незнайка было хорошее настроение, потому что поставил только одну кляксу

терпение и труд помогут избавиться от комет.