НИУ ИТМО

Факультет программной инженерии и компьютерной техники

Отчет по Лабораторной работе №3    
по курсу “Программирование”  
Вариант № 1057894

Выполнила:

Студентка группы P3110

Бегинина Анастасия Алексеевна   
Преподаватель:

Сорокин Роман Борисович

Санкт-Петербург

2021

Оглавление

[Текст задания 3](#_Toc88998625)

[Диаграмма классов 4](#_Toc88998626)

[Исходный код 4](#_Toc88998627)

[Результат работы программы 4](#_Toc88998628)

[Main.java 4](#_Toc88998629)

[Выводы по работе 7](#_Toc88998630)

# Текст задания

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

*Он привязал шар веревкой к ореховому кусту, чтобы не унесло ветром, после чего поделил малышей на два отряда. Одному отряду он велел собирать шелковичные коконы, чтобы размотать их и наделать шелковых нитей.Из этих нитей он велел им сплести огромную сетку. Другому отряду Знайка велел сделать большую корзину из тонкой березовой коры.  
 Пока Знайка со своими товарищами занимался этой работой, все жители Цветочного города приходили и смотрели на огромнейший шар,  
 который был привязан к ореховому кусту.Каждому хотелось потрогать шар руками, а некоторые даже пытались его приподнять.*

**Программа должна удовлетворять следующим требованиям:**

1. Доработанная модель должна соответствовать [принципам SOLID](https://en.wikipedia.org/wiki/SOLID_(object-oriented_design)).
2. Программа должна содержать как минимум два интерфейса и один абстрактный класс (номенклатура должна быть согласована с преподавателем).
3. В разработанных классах должны быть переопределены методы equals(), toString() и hashCode().
4. Программа должна содержать как минимум один перечисляемый тип (enum).

**Порядок выполнения работы:**

1. Доработать объектную модель приложения.
2. Перерисовать диаграмму классов в соответствии с внесёнными в модель изменениями.
3. Согласовать с преподавателем изменения, внесённые в модель.
4. Модифицировать программу в соответствии с внесёнными в модель изменениями.

**Отчёт по работе должен содержать:**

1. Текст задания.
2. Диаграмма классов объектной модели.
3. Исходный код программы.
4. Результат работы программы.
5. Выводы по работе.

**Вопросы к защите лабораторной работы:**

1. Принципы объектно-ориентированного программирования SOLID и STUPID.
2. Класс Object. Реализация его методов по умолчанию.
3. Особенности реализации наследования в Java. Простое и множественное наследование.
4. Понятие абстрактного класса. Модификатор abstract.
5. Понятие интерфейса. Реализация интерфейсов в Java, методы по умолчанию. Отличия от абстрактных классов.
6. Перечисляемый тип данных (enum) в Java. Особенности реализации и использования.
7. Методы и поля с модификаторами static и final.
8. Перегрузка и переопределение методов. Коварианты возвращаемых типов данных.
9. Элементы функционального программирования в синтаксисе Java. Функциональные интерфейсы, лямбда-выражения. Ссылки на методы.

# Диаграмма классов

Graphical user interface

Description automatically generated

# Исходный код

Ссылка на Git-репозиторий: https://github.com/nastyabeggin/ITMO\_pe\_first\_term/tree/main/programming/lab3

# Результат работы программы

## Main.java

public class Main {  
 public static void main(String[] args){  
 LittleBrainiac brainiac = new LittleBrainiac("Знайка");  
 Rope rope = new Rope();  
 Bush bush = new Bush();  
 Ball ball = new Ball();  
 brainiac.bind(rope, bush, ball);  
 System.*out*.println(brainiac.describe() + ' ' + ball.describe());  
 Wind wind = new Wind();  
 wind.blow(ball.is\_tied);  
 System.*out*.println(wind.describe());  
 Babies babies = new Babies();  
 Squad squad1 = new Squad("1");  
 Squad squad2 = new Squad("2");  
 brainiac.separate(babies, squad1, squad2);  
  
 System.*out*.println(brainiac.describe() + ' ' + babies.describe());  
  
 Pods pods = new Pods();  
 Thread thread = new Thread();  
 Net net = new Net();  
 System.*out*.println(squad1.getName() + " Отряд");  
 brainiac.order(squad1, OrderType.*СОБИРАТЬ*);  
 System.*out*.println(brainiac.describe() + ' ' + pods.describe() + squad1.describe() + " " + pods.describe());  
 brainiac.order(squad1, OrderType.*РАЗМАТЫВАТЬ*);  
 System.*out*.println(brainiac.describe() + ' ' + pods.describe() + squad1.describe() + " " + pods.describe());  
 brainiac.order(squad1, OrderType.*СДЕЛАТЬ*);  
 System.*out*.println(brainiac.describe() + ' ' + thread.describe() + squad1.describe() + ' ' + thread.describe());  
 brainiac.order(squad1, OrderType.*СПЛЕСТИ*);  
 System.*out*.println(brainiac.describe() + ' ' + net.describe() + squad1.describe() + ' ' + net.describe());  
  
 Basket basket = new Basket();  
 Crust crust = new Crust();  
 System.*out*.println(squad2.getName() + " Отряд");  
 brainiac.order(squad2, OrderType.*СДЕЛАТЬ*);  
 System.*out*.println(brainiac.describe() + ' ' + basket.describe() + " из " + crust.describe());  
 System.*out*.println(squad2.describe() + basket.describe() + " из " + crust.describe());  
  
 brainiac.refresh();  
 brainiac.work();  
 System.*out*.println(brainiac.describe());  
  
 Residents residents = new Residents("Все");  
 residents.come();  
 residents.watch(ball);  
 System.*out*.println(residents.getName() + residents.describe());  
  
 Residents residents\_2 = new Residents("Каждому");  
 residents\_2.touch(ball);  
 System.*out*.println(residents\_2.getName() + residents\_2.describe());  
  
  
 Residents residents\_3 = new Residents("Некоторые");  
 residents\_3.triedToLift(ball);  
 System.*out*.println(residents\_3.getName() + residents\_3.describe());  
  
 LittleBrainiac new\_one = new LittleBrainiac("Вася");  
 LittleBrainiac sec\_new\_one = new LittleBrainiac("Вася");  
 System.*out*.println(new\_one.hashCode());  
 System.*out*.println(sec\_new\_one.hashCode());  
  
 System.*out*.println(new\_one.equals(sec\_new\_one));  
 System.*out*.println(new\_one.toString());  
 }  
}

Знайка привязывает Шар привязан к ореховый куст с помощью Веревка привязана к ореховый куст. Шар не может быть унесен ветром

Ветер дует Ветер не может унести шар

Знайка разделил Малыши поделены на Отряд ничего не делает Отряд ничего не делает

1 Отряд

Знайка велит собирать шелковые коконы Отряд занимается работой Отряд собирать шелковые коконы

Знайка велит разматывать шелковые коконы Отряд занимается работой Отряд разматывать шелковые коконы

Знайка велит сделать шелковые нити Отряд занимается работой Отряд сделать шелковые нити

Знайка велит сплести огромная сетка Отряд занимается работой Отряд сплести огромная сетка

2 Отряд

Знайка велит сделать большая корзина из тонкая, берёзовая кора

Отряд занимается работой Отряд сделатьбольшая корзина из тонкая, берёзовая кора

Знайка занимается работой

Все Жители приходили Жители смотрели на Шар привязан к ореховый куст с помощью Веревка привязана к ореховый куст. Шар не может быть унесен ветром

Каждому Жители хотели потрогать шар

Некоторые Жители пытались приподнять шар

32107276

32107276

true

Этого сказочного персонажа зовут Вася

# Выводы по работе

Во время выполнения лабораторной работы я укрепила свои знания в ООП, получила опыт в применении enum, интерфейсов и абстрактных классов. Также изучила методы класса Object (toString, equals, hashCode) и научилась их переопределять. Познакомилась с принципами обьектно-ориентированного программирования SOLID.