МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОСИЙСКОЙ ФЕДЕРАЦИИ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**Лабораторная работа №3**

**По дисциплине «Технология программирования»**

**на тему**

**Классы-коллекции**

Группа: АВТ-808

Студент: Попов Д.А.

Новосибирск, 2020

***Задание:***

1. Изучить особенности реализации классов-коллекций в Java.

2. Доработать программу, созданную в лабораторной работе № 2:

1) добавить генерируемым объектам понятия «время рождения» и «время жизни». Время рождения устанавливается в момент генерации объекта, и по значению соответствует времени, прошедшему от начала симуляции. Время жизни – время, через которое объект должен исчезнуть, считая от времени рождения;

2) вынести установку параметров времени жизни объектов в пользовательский интерфейс. Для каждого типа объекта должно задаваться собственное время. Рекомендуется использовать текстовые поля, но следуют помнить о проверке на ввод некорректных данных;

3) добавить генерируемым объектам уникальные целочисленные идентификаторы (случайные числа), которые назначаются при генерации объекта. Для хранения сгенерированных идентификаторов используйте коллекцию удобную для поиска по варианту;

4) использовать коллекции по варианту. При генерации объекта происходит добавление его в коллекцию (в класс добавить поле идентификатора), а во вторую коллекцию: идентификаторы существующих объектов, в третью идентификатор + время рождения. При возникновении события по таймеру обойдите коллекцию и удалите все объекты, время жизни которых истекло, а также все данные во вспомогательных коллекциях; Для обеспечения доступа к коллекциям используйте структурный паттерн Singleton.

5) добавьте в панель управления кпопку «Текущие объекты». По нажатию на эту кнопку появляется модальное диалоговое окно, содержащее список всех «живых» объектов на момент нажатия со временем их рождения (время рождения – ключ). В класс диалогового окна должна передаваться коллекция с хранением объектов по времени рождения. Типы коллекций задаются вариантом.

Вариант 7

Коллекция для хранения объектов: Vector

Коллекция для хранения и поиска уникальных идентификаторов: HashSet

Коллекция для хранения времени рождения объектов: TreeMap

***Описание доработки программы:***

В класс House были добавлены поля:

private long timeOfBirth;

protected double id;

В классе MyComponent добавлено удаление объектов:

for(int i = 0; i < *count*; i++) {  
 if(singleton.getHouse(i) instanceof CapitalHouse ){  
 if(*time* > singleton.getTimeOfBirth(i) + CapitalHouse.*getLifetime*()){  
 singleton.destroyHouse(i);  
 *count*--;  
 }  
 }  
 else if(singleton.getHouse(i) instanceof WoodenHouse ){  
 if(*time* > singleton.getTimeOfBirth(i) + WoodenHouse.*getLifetime*()){  
 singleton.destroyHouse(i);  
 *count*--;  
 }

Изменения в классе Singleton:

Добавлены поля-хранилища домов, идентификаторов и времен рождения:

private Vector<House> houseVector;  
private HashSet<Double> id;  
private TreeMap<Double, Long> timesOfBirth;

Генерация дома перенесена также в этот класс:

public void setHouse(TypeOfHouse typeOfHouse){  
 AbstractFactory abstractFactory = ConcreteFactory.*concreteFactory*(typeOfHouse);  
 House house = abstractFactory.createHouse();  
 house.setTimeOfBirth(MyComponent.*getTime*());  
 houseVector.add(house);  
 id.add(house.getId());  
 timesOfBirth.put(house.getId(), MyComponent.*getTime*());  
}

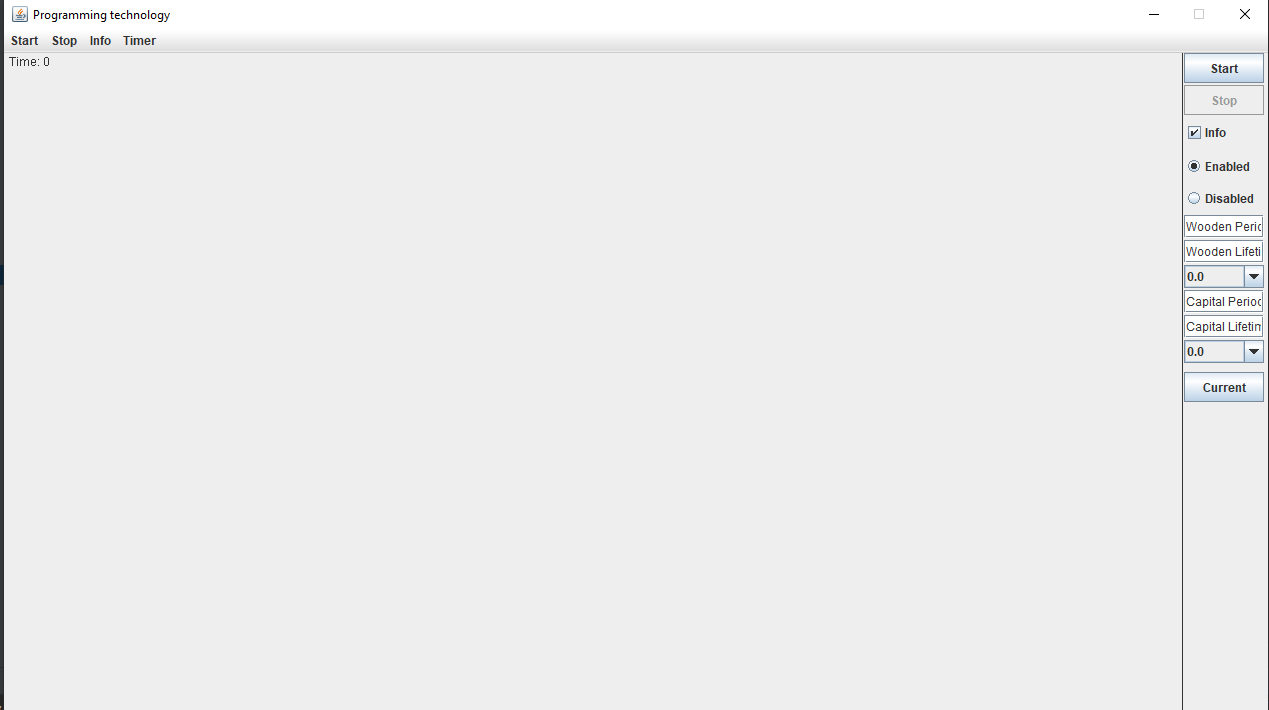
Уничтожение дома:

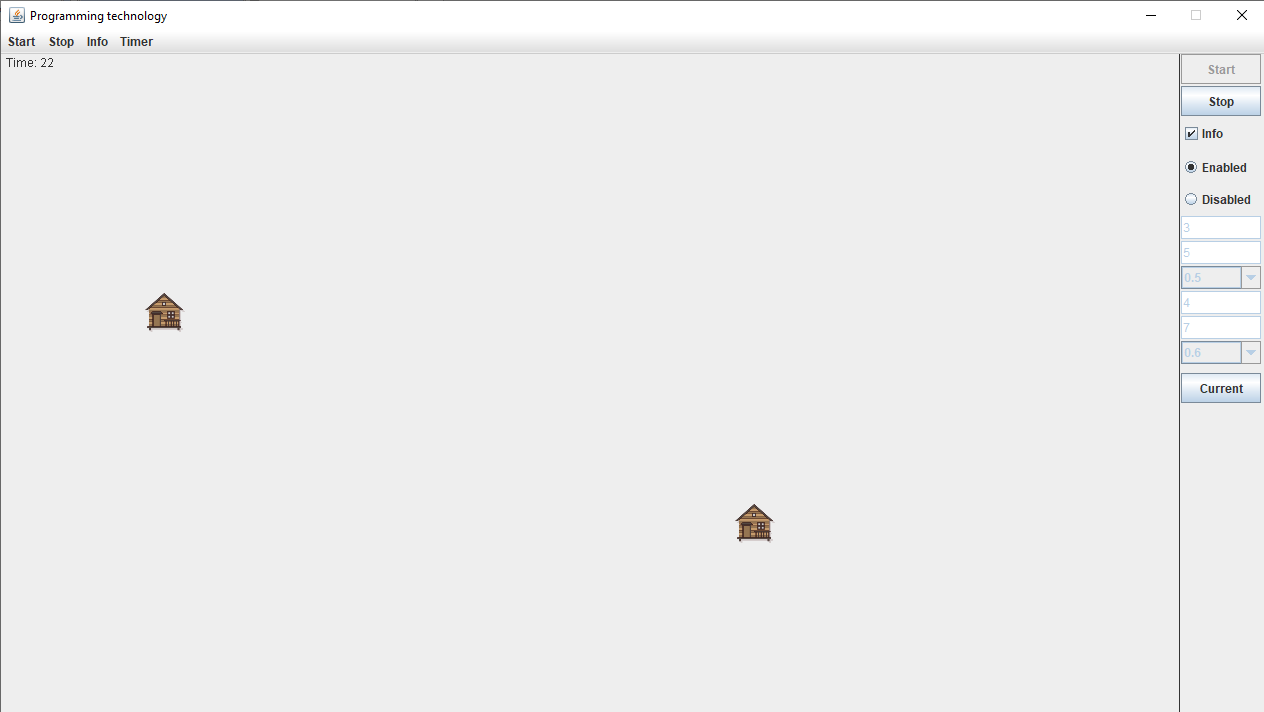
public void destroyHouse(int i) {  
 id.remove(houseVector.get(i).getId());  
 timesOfBirth.remove(houseVector.get(i).getId());  
 houseVector.remove(i); }

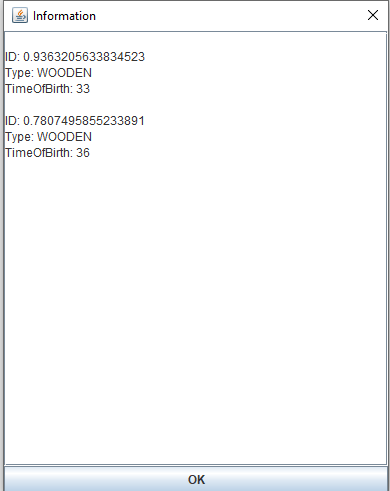
Добавлен класс, для отображения диалогового окна с текущими объектами на экране:

public class InformationDialog extends JDialog {  
  
 JButton jButtonOK = new JButton("OK");  
 JTextArea jTextArea = new JTextArea();  
 JScrollPane scrollPane = new JScrollPane(jTextArea);  
  
 InformationDialog(JFrame owner){  
 super(owner, "Information", true);  
 setBounds(730, 300, 400, 500);  
 add(jButtonOK, BorderLayout.*SOUTH*);  
 jTextArea.setEditable(false);  
 add(scrollPane);  
 }  
  
  
  
 public void showDialog(Vector<House> houseVector){  
  
 jButtonOK.addMouseListener(new MouseAdapter() {  
 @Override  
 public void mouseClicked(MouseEvent e) {  
 setVisible(false);  
 }  
 });  
  
 String string = "";  
  
 for(House it : houseVector){  
 string+="\nID: " + it.getId() +  
 "\nType: " + it.getTypeOfHouse() +  
 "\nTimeOfBirth: " + it.getTimeOfBirth() + "\n";  
 }  
  
 jTextArea.setText(string);  
  
 setVisible(true);  
  
  
 }  
}

Добавлено время жизни для объектов, а также возможность установки его из пользовательского интерфейса.







*Рис.1-3 – Демонстрация работы программы*

**Вывод:**

В ходе лабораторной работы были изучены особенности реализации классов-коллекций в Java.

**Программный код:**

https://github.com/AVT-808/AVT-808/tree/Popov-Dmitry-7