Практическая работа №6.

1. Работа с графическими изображениями.

1.1. **Задача:** Изучение теоретического материала. Закрепление полученных знаний путем разработки приложения на языке программирования Java в среде разработки eclipse.

1.2.**Тип занятий:** исследовательский.

1.3.**Содержание:** теоретический материал, задания для самостоятельного выполнения.

2.Теоретический материал к уроку

2.1. **Таймер**

Timer имя\_таймера = **new** Timer(частота выполнения,**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

блок команд;

}

});

Пример: Timer timer = **new** Timer(частота выполнения,**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

System.out.println("WOW!");

}

});

Запуск таймера осуществляется следующей командой: имя таймера.start();

Пример: timer.start();

3. Задания

3.1. уровень 1

1. Откройте ваш проект Game.
2. Добавьте метод в класс Pole после метода paintComponent для добавления предметов на игровое поле:

// Метод для проверки и добавление подарков на игровое поле

**private** **void** updateStart()

{

**int** kol=0; // Переменная для подсчета подарков на игровом поле

**for** (**int** i=0;i<7;i++) // Цикл перебора всех подарков массива

{

**if** (gamePredmet[i].act==**false**) // Если подарок не на игровом поле

{

**if** (kol<slogn) // Если текущее количество менее номера сложности (от 1 до 7)

{

gamePredmet [i].start(); // Активизация подарка на игровом поле, вывод его сверху игрового поля

**break**; // Прерывание цикла

}

}

**else** kol++; // Если подарок на игровом поле

}

}

1. В конструктор класса Pole добавьте таймер, который будет раз в три секунды(частота выполнения 3000) проверять и добавлять предметы на игровое поле В блоке команд будет метод updateStart();// Метод для проверки и добавление подарков на игровое поле.
2. Запустите таймер на выполнение.
3. Закомментируйте вызов метода start() в конструкторе класса Predmet, т.к. теперь метод у нас вызывается в методе updateStart.
4. Запустите программу на выполнение, у вас должны появляться предметы на экране с задержкой в три секунды.
5. Объявите в начале класса Predmet переменную Timer timer3;
6. В класс Predmet добавьте метод, осуществляющий движение подарка вниз.

**public** **void** vniz()

{

**if** (act==**true**) // Если подарок активен на игровом поле

{

y+=6; // Увеличение отступа сверху на 6 пикселей

}

**if** ((y+120)>=470) // Если подарок достиг нижней границы

{

timer3.stop(); // Остановка таймера

}

}

1. В конструктор класса Predmet добавьте таймер.

timer3 = **new** Timer(500,**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

vniz(); // Метод, осуществляющий движение подарка вниз

}

});

1. А в метод start класса Predmet добавьте

timer3.setDelay(300); // Установка временной задержки для таймера

timer3.start(); // Запуск таймера

1. Запустите программу на выполнение. Предметы должны двигаться к вашему ловцу.
2. Изменим метод paintComponent класса Pole,внутри цикла for изменим команды, если ругается на имена переменных таймера, вынесите их объявление из конструктора в начало класса, только объявление, присвоение значения оставьте в конструкторе.

gamePredmet [i].draw(gr); // Отображение подарка

**if** (gamePredmet [i].act==**true**) // Если подарок из массива подарков активен

{

**if** ((gamePredmet [i].y+120)>=470) // Если подарок достиг нижней границы {

**if** (Math.*abs*(gamePredmet [i].x - x) > 120)// Если подарок пропущен

{

gr.drawImage(endGame, 0,0, 800, 600, **null**); // Вывод картинки Окончания игры

!это делаем сами

//здесь останавливаем оба ваших таймера с вашими названиями, после имени таймера ставим точку и пишем stop();

**break**; // Прерывание цикла

}

**else** gamePredmet[i].act=**false**; // Снятие подарка с игрового поля, если он пойман шапкой

}

}

15) запускаем, играем1.