Министерство образования и науки РФ

Санкт-Петербургский государственный политехнический университет

Институт компьютерных наук и технологий

Кафедра «Системный анализ и управления»

УТВЕРЖДАЮ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_ г.

ОТЧЕТ

по дисциплине "Системный подход в разработке программного обеспечения"

**Лабораторная работа №2**

**Выполнил:**

студент гр. 3530902/00201 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О. В. Позолотин

подпись, дата

**Проверил**

доцент, к.т.н. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. А. Нестеров

подпись, дата

Санкт-Петербург

2022

**Ход работы**

**Версия от 16.08.2013**

Данная версия успешно запустилась (рисунок 1).

Изображение выглядит как текст

Автоматически созданное описаниеВ этой версии присутствуют несколько объектов, управляемых пользователем: серый передвижной блок, два слота для этого блока, управляемый вращающийся блок, неуправляемый вращающийся блок, положение которого зависит от положения первого.

Рисунок 1 – первая версия игры

**Версия от 22.08.2013**

Версия успешно запустилась (рисунок 2).

Добавлено: 2 вращающихся диска разных цветов и содержащие разные изображения в центре, 4 подвижных блока, слоты для блоков на дисках и красный вращающийся блок, связанный с красным диском.

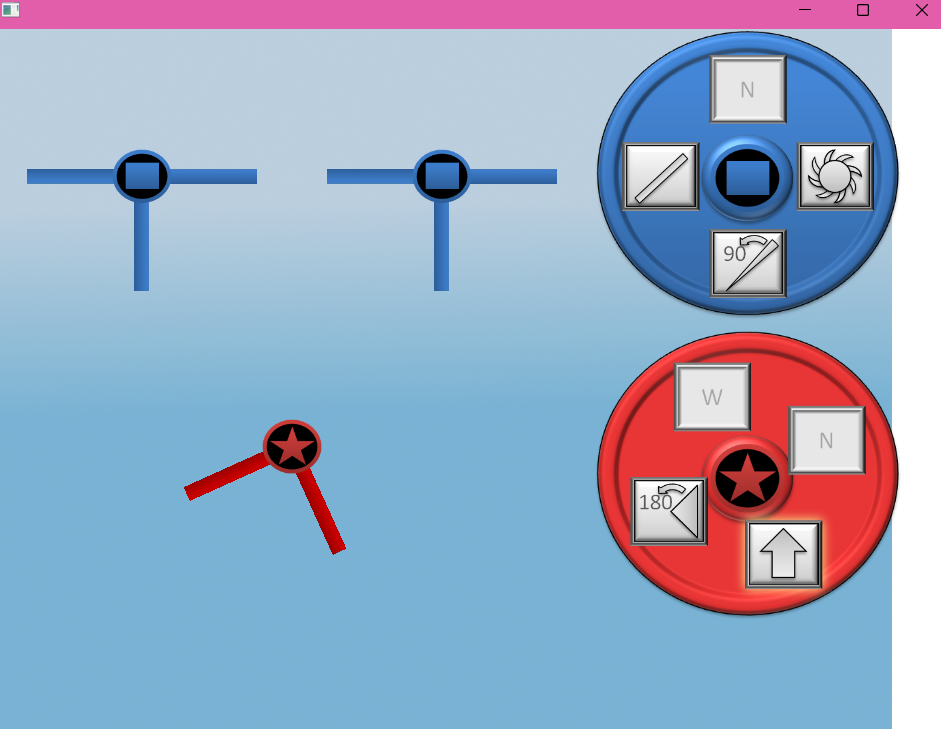
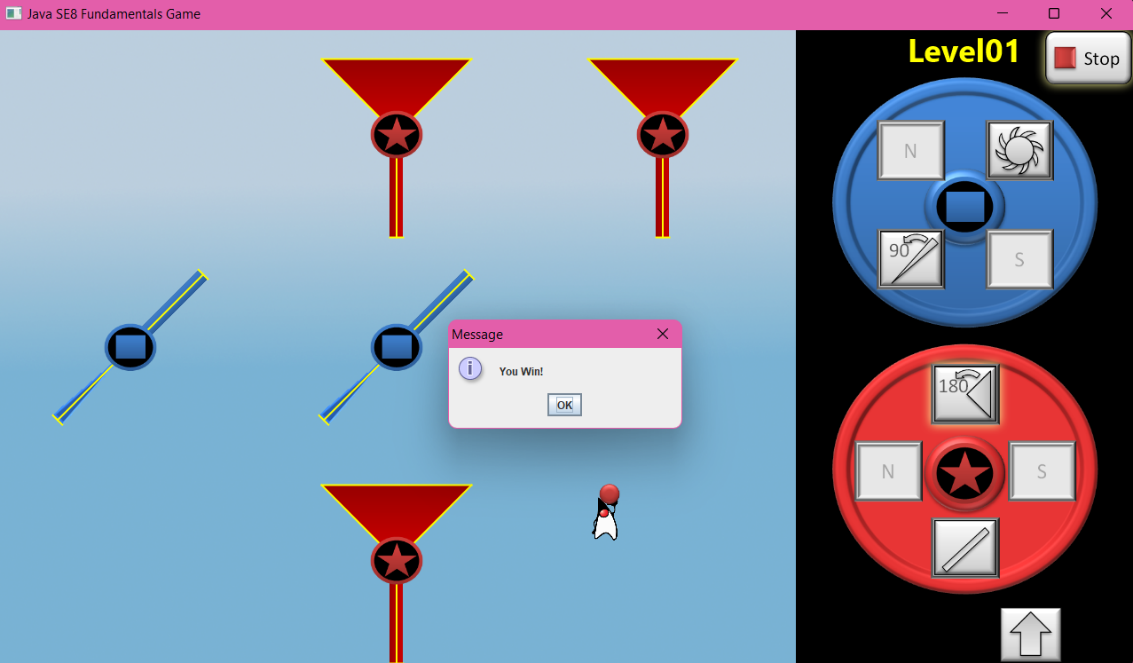


Рисунок 2 – вторая версия игры

**Версия от 27.09.2013**

Версия 3 успешно запустилась (рисунок 3).

В этой версии добавлены новые объекты: шарик, финиш, кнопка запуска игры. Значительно обновлен интерфейс игры. Добавлен новый функционал для ранее созданных объектов из прошлых версий игры.

Рисунок 3 – 3 версия игры (успешное прохождение уровня)**Версия от 16.10.2013**

Версия успешно запустилась (рисунок 4).

В данной версии реализована функция выбора уровня. Добавлены новые уровни и свойства объектов. Добавлен тестовый уровень, на котором можно наблюдать как шарик взаимодействует с окружением.

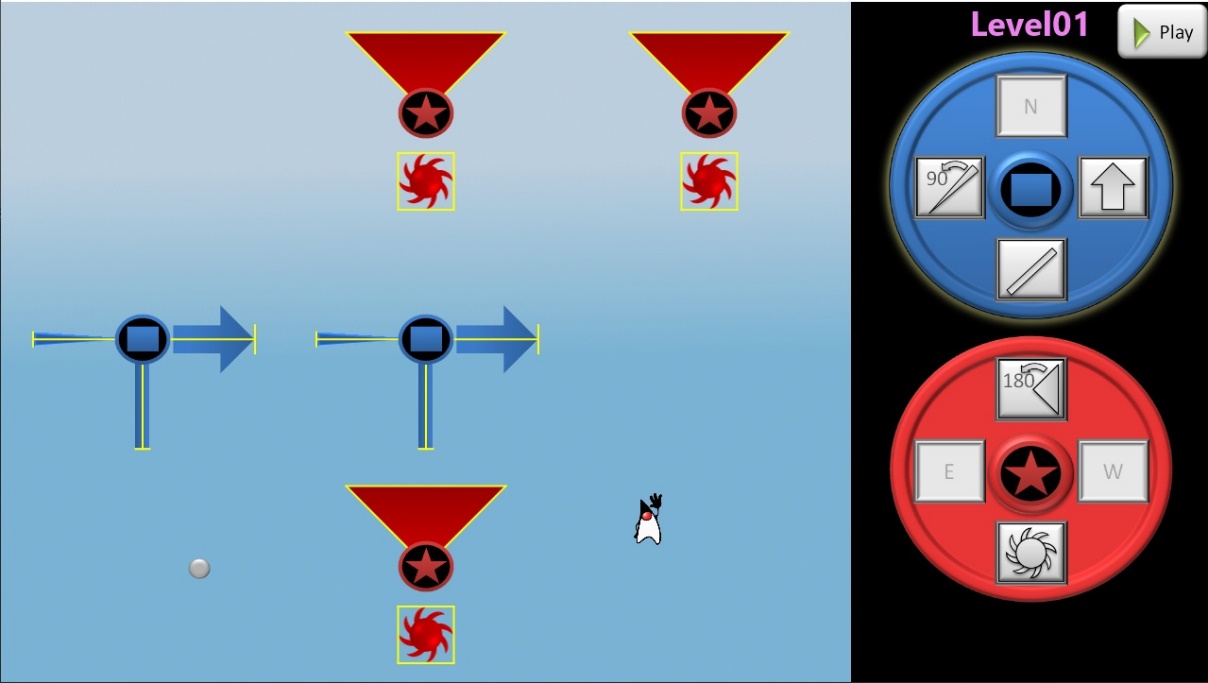


Рисунок 4 – 4 версия игры

**Версия от 21.11.2013**

Версия игры успешно запустилась (рисунок 5).

Добавлена возможность выбрать уровень из предложенного списка. Добавлены новые уровни, содержащие новое окружение. Заметно изменен дизайн игры.

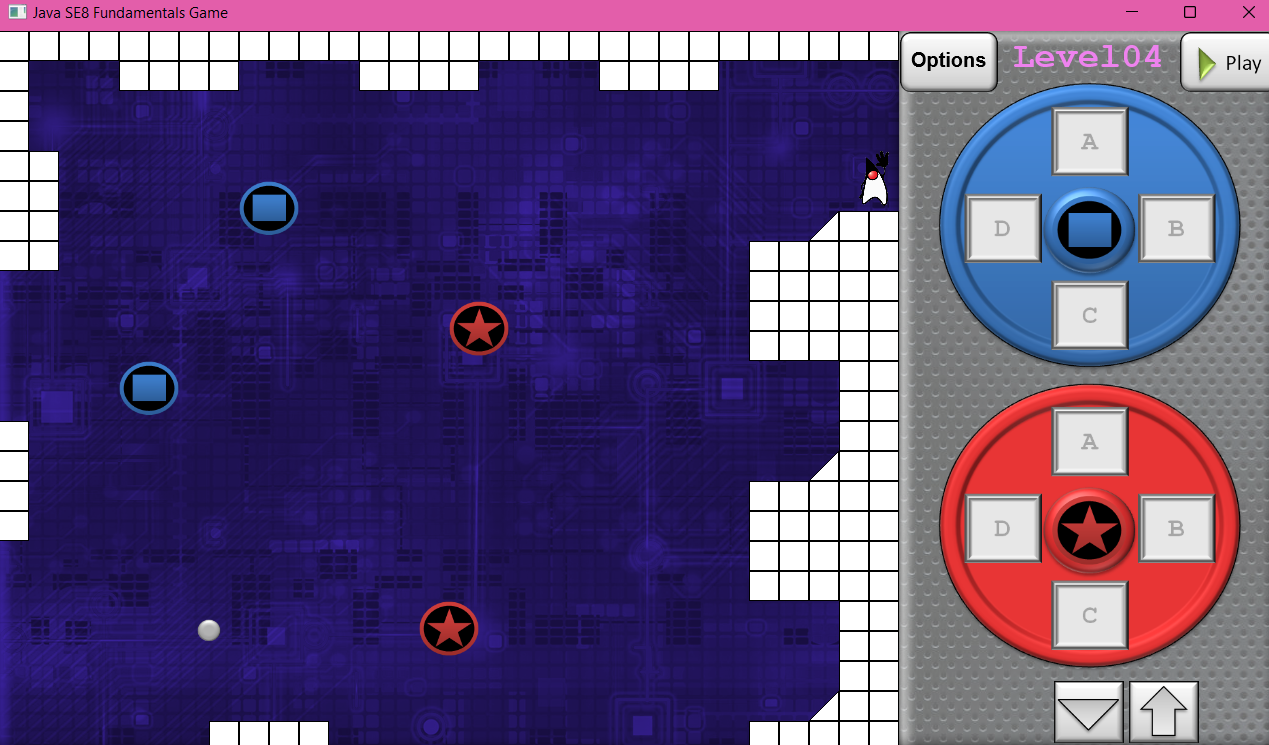


Рисунок 5 – пятая версия игры **Финальная версия**

Финальная версия игры успешно запустилась (рисунок 6).  
 Добавлены визуальные эффекты, изменена заставка, добавлено множество новых уровней. Можно выбирать уровни прямо в игре (рисунок 7).

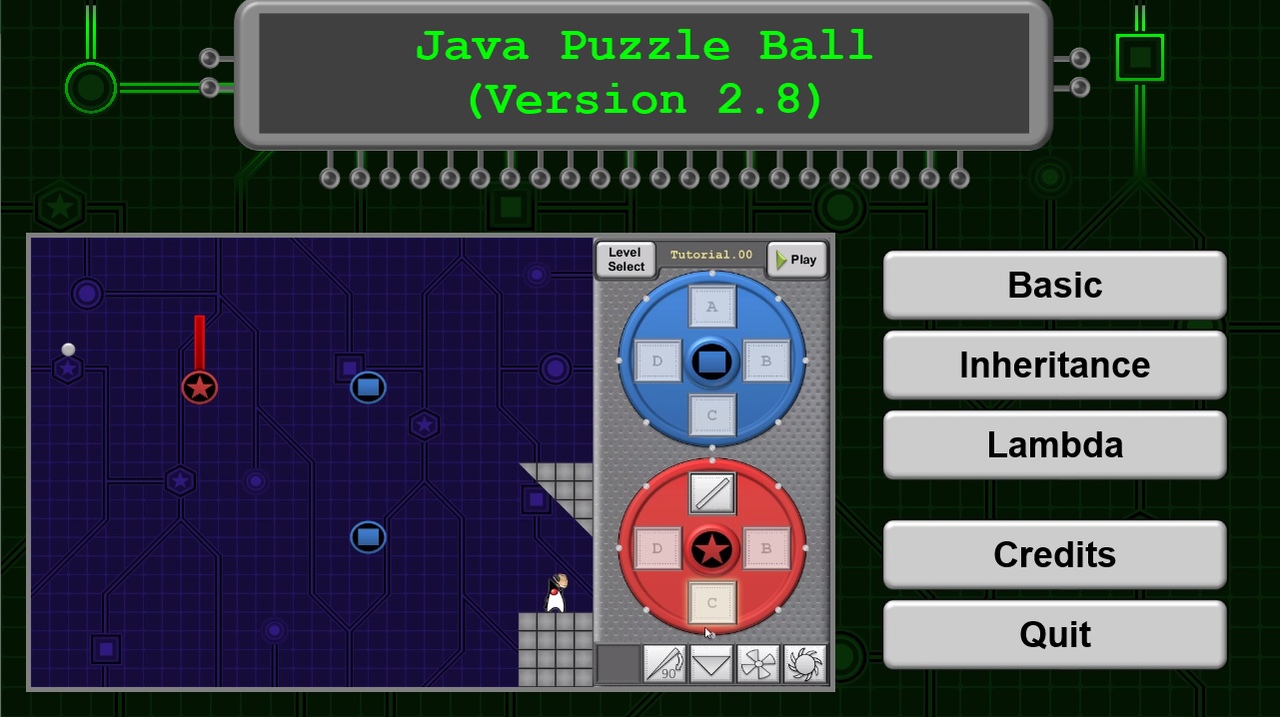


Рисунок 6 -финальная версия игры

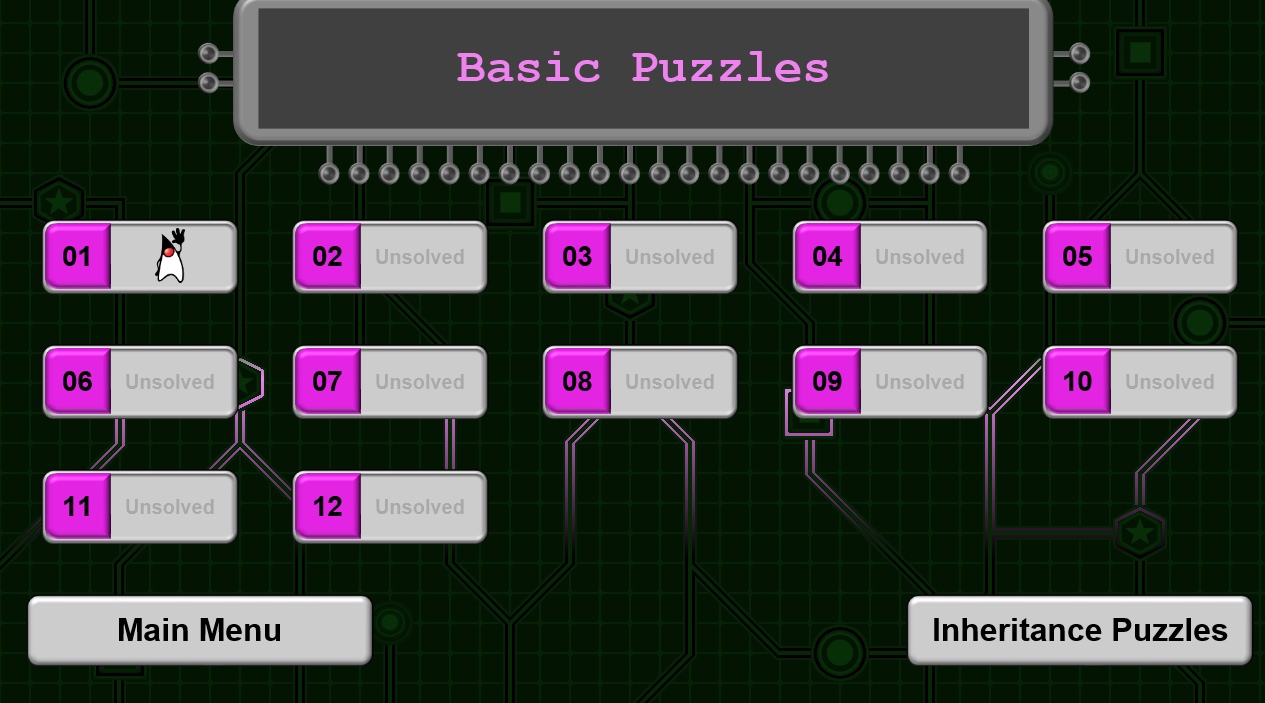


Рисунок 7 -возможность выбора уровней прямо в игре

На уровнях появились подсказки, помогающие понять смысл игры( рисунок 8).

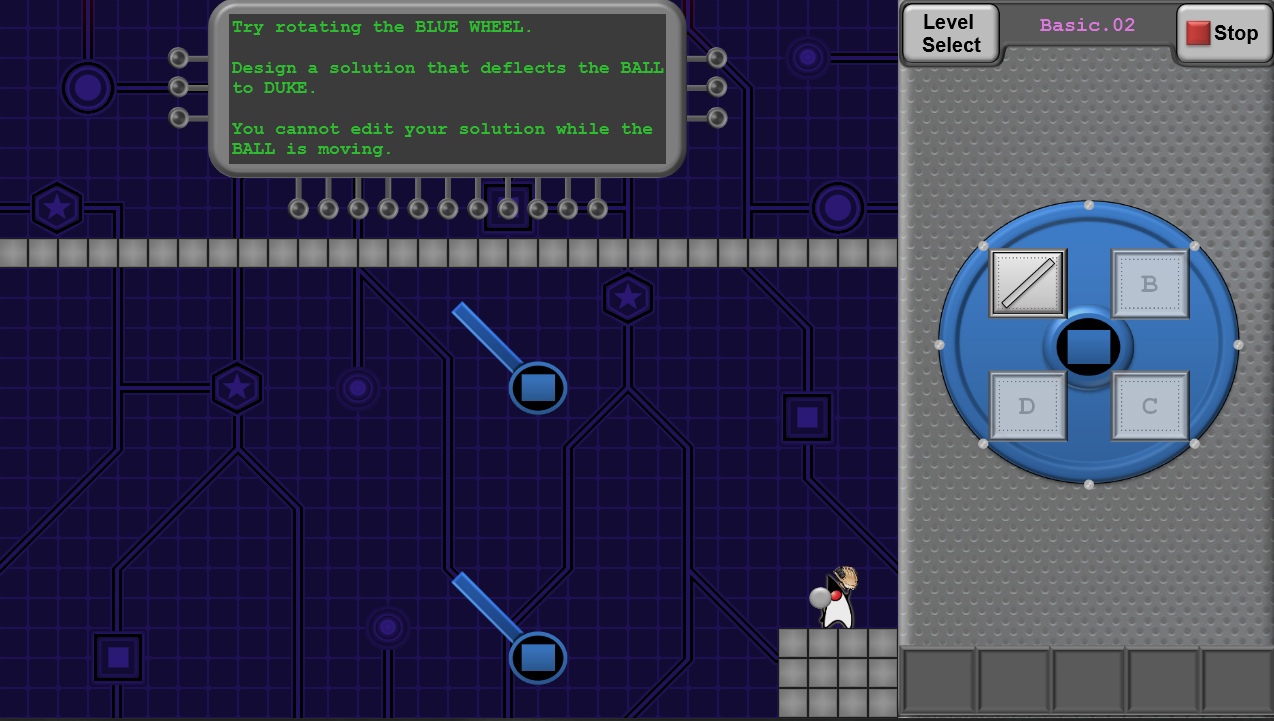


Рисунок 8 – подсказки в игре

# Простая ASCII-графика

**Задача.** Воспроизвести приведенное выше улыбающееся лицо, используя 8 операторов вывода. Помимо пробела ваш рисунок будет основан на одном символе, таком как X или #.

**Решение:**

Для выполнения лабораторной работы используем приведенный проект ProblemSet2\_1.

Написанный код для решения задачи и результат его работы представлены на рисунке 9.

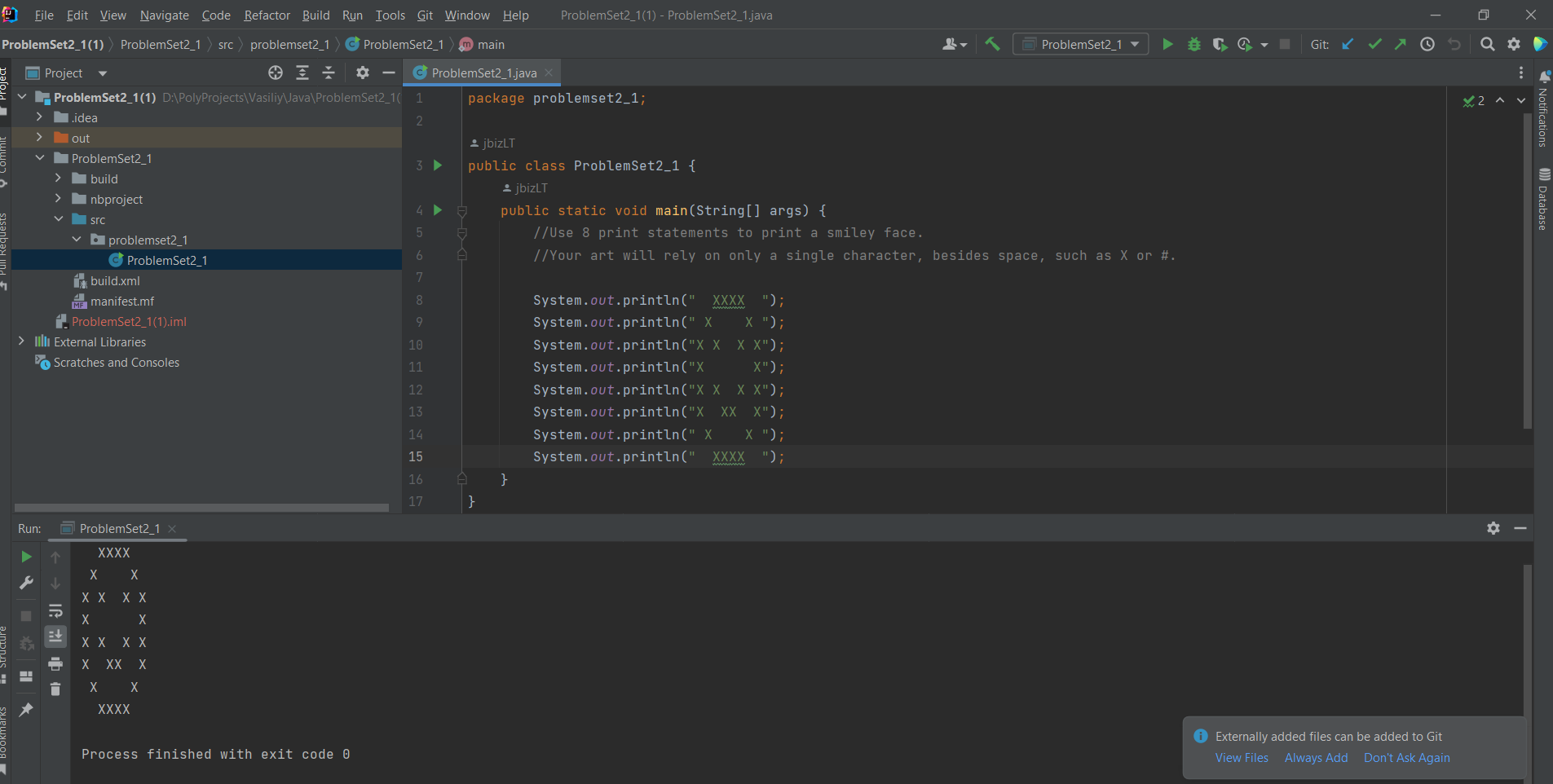


Рисунок 9 – рабочий код и результат работы программы

**Задача.** Создать собственную оригинальную ASCII-графику, используя операторы вывода. С помощью комментариев опишите, что показано на изображении.

Помимо пробела рисунок может быть основан на одном символе, таком как X или #. Однако рекомендуется использовать несколько разных символов, как в рассмотренном на занятии примере кота.

Рисунок также должен:

• использовать не менее 8 операторов вывода;

• быть в ширину не меньше 8 символов;

• использовать не менее 20 символов, отличных от пробела.

**Решение:**

Для выполнения лабораторной работы используем приведенный проект ProblemSet2\_2.

Дописываем код так, чтоб на экран выводилось изображение свиньи и оленя, составленных из символов. Рабочий код представлен на рисунке 10, результат работы – на рисунке 11.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10 – рабочий код

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11 – результат работы программы

**The Snake Box Factory**

**Обзор.**

Уважаемый разработчик программного обеспечения,

Наша всемирно известная компания Snake Box Factory поставляет клиентам высококачественные картонные коробки с индивидуальным дизайном. Наши ящики заполнены змеями самого высокого качества, подобранными по специальному заказу. Мы обслуживаем тысячи клиентов по всему миру, а наш показатель удовлетворенности клиентов составляет 98%. Однако в настоящее время заказы оформляются в письменном виде на картоне, а затем передаются между отделами с помощью змей-курьеров. Мы полагали, что это будет хороший способ показать уверенность в качестве и пользе нашей продукции. Наша компания продолжает расти, и мы осознали, что это была не самая лучшая идея. Мы думаем, что пришло время внедрить более традиционные и цифровые операции. Вы могли бы помочь нам разработать соответствующее программное обеспечение?

С уважением,

Президент Джордж Джонсон

The Snake Box Factory

**Задачи**. Прочтите сценарий в обзоре и подумайте, какие объекты можно смоделировать в рамках создания программного решения. Определите 3 объекта в этом сценарии (помните, объекты могут быть материальными или абстрактными). Для каждого объекта перечислите 3 свойства и 3 варианта поведения.

Оформите свое решение в виде документа, а не файла .java.

**Решение:**

|  |  |  |
| --- | --- | --- |
| **Объекты** | **Свойства** | **Варианты поведения** |
| Доставщик | Стаж  Почасовая ставка  Номер телефона | Доставить товар  Провести оплату на месте  Забрать товар со склада |
| Склад | Номер склада  Площадь склада  Содержимое склада | Выдать товар  Принять товар  Ревизия товара |
| Клиент | Имя  Телефон  Возраст | Оформление заказа  Оплата товара  Обратная связь |

**Изучение режима отладки**

**Задачи.** В данном задании необходимо открыть предложенный проект Variables02, внести правки в соответствии с комментариями, проверить результат. Поставить точки останова, при выполнении в режиме отладки, показать для нескольких точек, какие переменные с какими значениями.

**Решение:**

Измененный код представлен на рисунке 12.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12 – измененный код

Поставим точки останова в строке 15 и запустим режим отладки (рисунок 13)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 13 – точка останова на 15 строке

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 14 – после выполнения шага значение переменно х поменялось

У отладчика есть 5 действий: “Step over”, “Step into”, “Force step into”, “Step out”, “Run to cursor”.

Step over – происходит переход к следующей строке.

Step into – переход по строкам, но с заходом в метод, если таковой встречается по пути отладки

Force step into – заход в Java-библиотеки

Step out – выход из метода

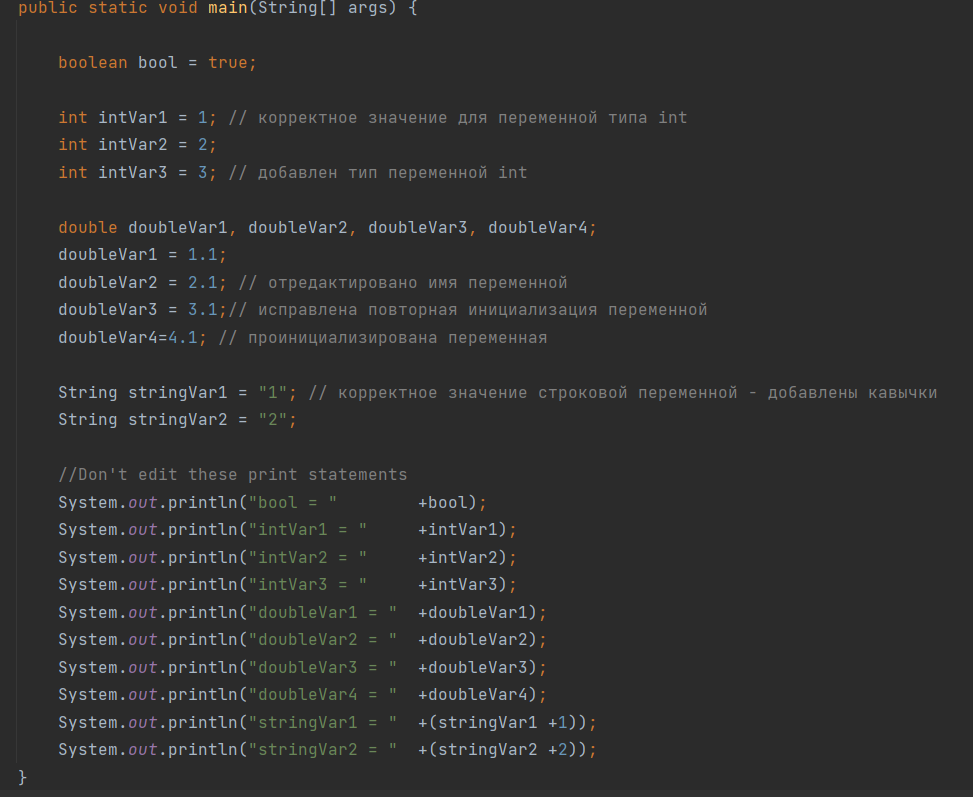
Run to cursor – переход к месту, на котором стоит курсор

**Исправление ошибок в программе**

**Задачи.** Открыть проект Variables03, исправить ошибки, поставить комментарии относительно сделанных исправлений.

**Решение:**

Редактируем код программы для корректной работы (рисунок 15).

Рисунок 15 – отредактированный текст

**Заключение**

В ходе работы я ознакомился с различными версиями игры JavaPuzzleBall. Работал с выводом в консоль, построчно выводя в консоль изображения смайлика, свинья и оленя, составленные из ASCII символов. Также я пользовался инструментами отладки, редактировал некорректный код.