МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

ЛУГАНСКОЙ НАРОДНОЙ РЕСПУБЛИКИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ЛУГАНСКОЙ НАРОДНОЙ РЕСПУБЛИКИ

«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ ТАРАСА ШЕВЧЕНКО»

Институт физики, математики и информационных технологий

Кафедра ИТС

**КУРСОВАЯ РАБОТА**

По дисциплине: **Объектно-ориентированное программирование**

на тему: **Написание игры «Hungry Panda»**

Выполнила:

Студентка ИФМИТ

2 курс ПИ

Антонюк Е. В.

Руководитель: Шкандыбин Ю. А.

Луганск

2016

СОДЕРЖАНИЕ

Введение

Глава І. Java

1.1. История развития

1.2. Объектно – ориентированное программирование

Розділ ІІ. Анализ технического развития

Розділ ІІІ. Обоснование алгоритма и структуры программы

3.1. Основной алгоритм игрового процесса

3.2. Алгоритм функционирования программы

Розділ ІV. Разработка программы

4.1. Разработка системы классов

Розділ V. Тестирование игровой программы

5.1. Разработка системы классов

Заключение

Список используемой литературы

**Введение**

Игра - непременный спутник развития человечества. Они использовались для социализации подрастающего поколения, для подготовки к коллективной охоте, для тренировки. Но учебно-тренировочные функции были не главными в древней игровой деятельности; главное поле игры — это праздники, ритуалы, первобытное искусство (танцы, музыка, мифы). Все эти занятия представляют собой археокультурную творческую и коммуникативную деятельность.

Вместе с появлением компьютеров появились компьютерные игры, которые сразу же нашли массу поклонников. Эти игры с детских лет сопровождают подрастающее поколение.

Развитие и совершенствование игр тесно связано с развитием компьютерного обеспечения и технологий. Сейчас многие составные части компьютеров разрабатываются, чуть ли не специально для игр. Например, дорогие видеокарты, стоимость которых доходит до половины стоимости удовлетворительного компьютера для работы в офисе. Все игры разрабатываются с учётом последних новинок компьютерной техники, реагируя на все достижения и всё ближе подходя к реальности изображения и звука. На сегодняшний день существуют поражающие своей правдоподобностью игры с хорошим графическим и звуковым оформлением, почти полностью имитирующим жизнь. Имеется и постоянно возникает огромное количество фирм, которые предоставляют всё новые и новые игры крайне разнообразного характера.

**Классификация игр по жанру:**

Четкая классификация затруднена из-за того, что подчас трудно отнести игру к каком-нибудь конкретному жанру. Игра может представлять собой как смешение существующих жанров, так и не относиться ни к одному из них. Несмотря на это, в ходе развития компьютерных игр сложилась следующая классификация:

1. 3D Shooter (3D-шутеры, "бродилки")

Основной принцип состоит в изображении виртуального пространства и предметов посредством игровой программы, исполняемой на компьютере. При этом игрок может воздействовать на виртуальную игровую среду. Применяется для обозначения всех видов компьютерных игр, содержащих элементы боя в виртуальном трехмерном пространстве.

1. Arcade (аркада)

Игры, в которых игроку приходится действовать быстро, полагаясь в первую очередь на свои рефлексы и реакцию. Аркады характеризуются развитой системой бонусов: начисление очков, постепенно открываемые элементы игры и т.д.

1. Fighting (Драки)

В драках два персонажа дерутся на арене, применяя различные удары, броски и комбинации. Характеризуется большим количеством персонажей (бойцов) и ударов (иногда больше ста для каждого персонажа).

1. Platformer (Платформеры)

Понятие платформеров пришло с игровых приставок. Именно там этот жанр наиболее популярен. Основной задачей игрока является преодоление препятствий (ям, шипов, врагов и т.д.) с помощью прыжков. Зачастую приходится прыгать по абстрактно расставленным в воздухе "палочкам" (т.н. платформам), отсюда и пошло название жанра.

1. Scrollers (Скроллеры)

В скроллерах экран непрерывно движется в одну из сторон, а игроку предлагается уничтожать появляющихся врагов и собирать появляющиеся бонусы. По направлению движения различают вертикальные и горизонтальные скроллеры.

1. Simulation (симуляторы)

Игра-симуляция. При помощи компьютера, как можно более полно, имитируется управление каким-либо сложным технической системы (например: боевым истребителем, автомобилем и т.д.).

1. Strategy (стратегии)

Игра требующая выработки стратегии, например для победы в военной операции. Игрок управляет не одним персонажем, а целым подразделением, предприятием или даже вселенной.

1. Sport (спортивные)

Как и следует из названия - имитация какой-либо спортивной игры, например футбола.

1. Puzzle (головоломки, логические)

В некомпьютерной головоломке роль арбитра, следящего за соблюдением правил, играет или сам игрок (пасьянс), или некоторое механическое устройство (кубик Рубика). С появлением компьютеров возможности головоломок расширились, так как написать компьютерную программу проще, чем сконструировать механическое устройство. Головоломки, как правило, не требуют реакции от игрока (однако многие ведут счёт времени, потраченного на решение).

1. Traditional (традиционные) и board (настольные)

Компьютерная реализация настольных игр, например шахмат.

Компьютерные игры занимают важную часть реальной жизни в нашем НЕ виртуальном пространстве. Для кого-то это просто способ скоротать свободное время, для кого-то – ежедневный выброс эмоций после трудового и учебного дня, а для кого-то - смысл жизни. Каждому - свое. Но для меня гораздо важнее лишний час свободного времени посветить себе или друзьям, выйти погулять. Там небо измеряется не в пикселях и разрешение не повлияет на качество картинки. Да, нельзя выйти из игры и сохранить нужный момент, но так даже интереснее.

**Глава І. Java**

* 1. **История развития**

Язык программирования Java – это высокоуровневый объектно-ориентированный язык, разработанный в компании Sun Microsystems.

С момента появления языка Java его возможности и сферы использования значительно увеличились. В настоящее время принято говорить о Java не как об отдельном языке программирования, а как о целом семействе технологий.

Изначально язык назывался Oak («Дуб») разрабатывался Джеймсом Гослингом для программирования бытовых электронных устройств. Впоследствии он был переименован в Java и стал использоваться для написания клиентских приложений и серверного программного обеспечения. Назван в честь марки кофе Java, которая, в свою очередь, получила наименование одноимённого острова (Ява), поэтому на официальной эмблеме языка изображена чашка с горячим кофе. Существует и другая версия происхождения названия языка, связанная с аллюзией на кофе-машину как пример бытового устройства, для программирования которого изначально язык создавался. В соответствии с этимологией, в русскоязычной литературе с конца двадцатого и до первых лет двадцать первого века название языка нередко переводилось как Ява, а не транскрибировалось, как это стало общепринятым позднее.

Приблизительно в 1990 Джеймс Гослинг, Билл Джой, Патрик Ногтон и другие в Sun Microsystems начали разрабатывать язык по имени Oak. Прежде всего, они видели применение Java для встроенных микрокомпьютерных модулей бытовой техники, в видеомагнитофонах, тостерах, а также для PDA (personal data assistants).

Чтобы решать эти задачи, Oak должен был быть:

* Независимым от платформы (с тех пор как продукцию стали вовлекать многие изготовители);
* Чрезвычайно надежным;
* Компактным.

В основе Java лежат языки С и C++. Его синтаксис получается из С, а ориентированные на объект особенности влияет C++.

Java – полностью объектно-ориентированный язык программирования. В Java отсутствует понятие процедур. С помощью Java мы можем решить различные задачи и тот же самый круг проблем, что и на других языках программирования. Java может использоваться для создания двух типов программ: Приложений и Апплетов.

Приложение - программа, которая выполняется на нашем компьютере, под его операционной системой. Приложения Java могут быть непосредственно выполнены, используя интерпретатор Java.

Апплет - небольшая программа работающая с окнами, которые внедрены в страницу HTML. Чтобы выполнить Java апплеты, нужна поддержка Java Web-браузером, то есть Internet Explorer, Netscape Navigator, Hot Java и т.д. или средство просмотра апплета. Также Java допускал другие средства, с помощью которых браузер мог выполнить программу Java на нашей системе.   
  
Java – это интерпретируемый и компилированный язык программирования. Исходный текст (файлы с расширением a Java) откомпилирован со справкой компилятора Java (javac), который преобразовывает исходный текст в байт-код (файлы с расширением a.class). Цель проектировщиков Java состояла в том, чтобы разработать язык, посредством которого программист мог записать код, который мог бы выполняться всегда, в любое время

**Сферы применения** **Java-технологий**

В настоящее время Java – технологии находят широкое применение в различных сферах. Это, в первую очередь, вызвано особенностями создания и запуска Java-приложений.

1. Разработка приложений (application)

Язык Java используется для разработки обычных прикладных программ, которые запускаются пользователем и имеют доступ ко всем ресурсам компьютера наравне с любыми другими программами. Для запуска Java-приложений в операционной системе должна быть установлена среда исполнения Java (JRE);

1. Разработка мидлетов (midlet)

Язык Java используется для разработки специализированных программ, предназначенных для использования на мобильных устройствах. Структура и возможности мидлетов обусловлены техническими особенностями мобильных устройств. Для запуска мидлетов мобильное устройство должно поддерживать соответствующую возможность;

1. Разработка апплетов (applet)

Язык Java используется для разработки специализированных программ, обычно небольшого размера, запускаемых браузером внутри web-документа для вывода динамического содержимого и/или интерактивного взаимодействия с пользователем. Для запуска апплетов браузер должен содержать соответствующий модуль (plug-in);

1. Разработка JSP-страниц (Java Server Pages)

JSP-страницы представляют собой HTML-документы со вставками на языке Java, которые используются для анализа пользовательских запросов и динамического формирования web-документов на основе результатов обработки этих запросов. Для обработки JSP-страниц Web-сервер должен содержать соответствующий модуль;

1. Разработка сервлетов (servlet)

На языке Java разрабатываются специализированные программы-сервлеты, функционирующие в рамках Web-сервера и имеющие доступ к его ресурсам: файлам, базам данных и т.п. Для запуска сервлетов используется Web-сервер со специальным модулем (контейнером сервлетов) или специальный сервер приложений.

**Достоинства и недостатки** **Java-технологий**

Java-технологии имеют много особенностей, отличающие их от других технологий разработки программного обеспечения.

1. Переносимость

Программы, написанные на языке Java, после однократной трансляции в байт-код могут быть исполнены на любой платформе, для которой реализована виртуальная Java-машина. Наиболее эффективно возможности реального компьютера может использовать только программа, написанная с использованием «родного» машинного кода.

1. Безопасность

Функционирование программы полностью определяется (и ограничивается) виртуальной Java-машиной. Отсутствуют указатели и другие механизмы для непосредственной работы с физической памятью и прочим аппаратным обеспечением компьютера. Дополнительные ограничения снижают возможность написания эффективно работающих Java-программ.

1. Надежность

В языке Java отсутствуют механизмы, потенциально приводящие к ошибкам: арифметика указателей, неявное преобразование типов с потерей точности и т.п. Присутствует строгий контроль типов, обязательный контроль исключительных ситуаций. Многие логические ошибки обнаруживаются на этапе компиляции. Наличие дополнительных проверок снижает эффективность выполнения Java-программ.

1. Сборщик мусора

Освобождение памяти при работе программы осуществляется автоматически с помощью «сборщика мусора», поэтому программировать с использованием динамически распределяемой памяти проще и надежнее. При интенсивной работе с динамически распределяемой памятью возможны ошибки из-за того, что «сборщик мусора» не успел освободить неиспользуемые области памяти.

1. Стандартные библиотеки

Многие задачи, встречающиеся при разработке программного обеспечения, уже решены в рамках стандартных библиотек. Использование объектно-ориентированного подхода позволяет легко использовать готовые объекты в своих программах.

Для запуска приложения необходима установка JRE, содержащего полный набор библиотек, даже если все они не используются в приложении. Отсутствие библиотеки необходимой версии может воспрепятствовать запуску приложения.

1. Самодокументируемый код

Имеется механизм автоматического генерирования документации на основе комментариев, размещенных в тексте программ.

1. Многообразие типов приложений

На языке Java возможно реализовать абсолютно разные по способу функционированию и сфере использования программы.

* 1. **Объектно – ориентированное программирование**

Объектно-ориентированное программирование (ООП) –  это метод программирования, при использовании которого главными элементами программ являются объекты, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

Необходимо обратить внимание на следующие важные части этого определения:

1. Объектно – ориентированное программирование использует в качестве основных логических конструктивных элементов объекты, а не алгоритмы;
2. каждый объект является экземпляром определенного класса;
3. классы образуют иерархии.

Программа считается объектно – ориентированной, только если выполнены все три указанных требования. В частности, программирование, не использующее наследование, называется не объектно-ориентированным, а программированием с помощью абстрактных типов данных.

Объектно-ориентированная технология программирования наиболее распространена и востребована в настоящее время. При объектно – ориентированном подходе к программированию программа представляет собой совокупность взаимодействующих между собой данных – объектов. Функциональную возможность и структуру объектов задают классы – типы данных, определенные пользователем.

Современными языками объектно-ориентированного программирования являются С++ и Java. С середины 90-х годов многие объектно–ориентированные языки реализуются как системы визуального проектирования, в которых интерфейсная часть программного продукта создается в диалоговом режиме, практически без написания программных операторов.

Центральной идеей ООП является реализация понятия "абстракция". Смысл абстракции заключается в том, что сущность произвольной сложности можно рассматривать, а также производить определенные действия над ней, как над единым целым, не вдаваясь в детали внутреннего построения и функционирования. При создании программного комплекса необходимо разработать определенные абстракции.

Одним из основных способов создания абстракции является использование концепции иерархической классификации. Ее суть заключается в том, что сложные системы разбиваются на более простые фрагменты.

Практически все сложные системы иерархичны, и уровни их иерархии отражают различные уровни абстракции. Для каждой конкретной задачи рассматривается соответствующий уровень. Выбор низшего уровня абстракции достаточно произволен. Выбранный уровень в одном случае в качестве низшего уровня может оказаться уровнем достаточно высокой абстракции в другом проекте. Различают типовую иерархию и структурную иерархию, которые далее мы будем называть соответственно структурой классов и структурой объектов.

Во всех объектно – ориентированных языках программирования реализованы следующие основные механизмы (постулаты) ООП:

* Инкапсуляция;
* Наследование;
* Полиморфизм.

Все эти механизмы важны для разработки и использования абстракций.

1) Инкапсуляция – механизм, связывающий вместе код и данные, которыми он манипулирует, и одновременно защищающий их от произвольного доступа со стороны другого кода, внешнего по отношению к рассматриваемому. Доступ к коду и данным жестко контролируется интерфейсом. Основой инкапсуляции при ООП является класс. Механизма инкапсуляции позволяет оставлять скрытыми от пользователя некоторые детали реализации класса (то есть инкапсулировать их в классе), что упрощает работу с объектами этого класса.

2) Наследование – механизм, с помощью которого один объект (производного класса) приобретает свойства другого объекта (родительского, базового класса). При использовании наследования новый объект не обязательно описывать, начиная с нуля, что существенно упрощает работу программиста. Наследование позволяет какому-либо объекту наследовать от своего родителя общие атрибуты, а для себя определять только те характеристики, которые делают его уникальным внутри класса.

Наследование есть очень важное понятие, поддерживающее концепцию иерархической классификации.

3) Полиморфизм – это свойство, позволяющее использовать один интерфейс для целого класса действий. Конкретное действие определяется характерными признаками ситуации.

В более общем виде концепция полиморфизма выражается фразой "один интерфейс – много методов". Это означает, что для группы связанных действий можно использовать один обобщенный интерфейс. Полиморфизм позволяет понизить уровень сложности за счет возможности применения одного и того же интерфейса для задания целого класса действий. Выбор же конкретного действия (т.е. функции) применительно к той или иной ситуации ложится "на плечи" компилятора.