Муниципальное бюджетное общеобразовательное учреждение

«Средняя общеобразовательная школа с углубленным изучением отдельных предметов № 42 имени Б.Г. Шуклина»

Итоговый индивидуальный проект

по информатике

на тему

«Языки программирования и их значение для современного мира»

Выполнил

Хороших Игорь Алексеевич

ученик 9В класса

МБОУ «СОШ № 42»

Руководитель проекта

Абакумова Ольга Николаевна

учитель информатики

Курск, 2020

**Содержание**

[Введение 3](#_Toc35468747)

[Глава 1. История развития языков программирования 4](#_Toc35468748)

[1.1 Java 5](#_Toc35468749)

[1.2 С++ 8](#_Toc35468750)

[1.3 С# 10](#_Toc35468751)

[Глава 2. Программирование в современном мире 12](#_Toc35468752)

[2.1. Описание профессии «программист» 12](#_Toc35468753)

[2.2 Создание игры с помощью Game Maker 14](#_Toc35468754)

[Заключение 17](#_Toc35468755)

[Список литературы 18](#_Toc35468756)

**Введение**

Со времени появления первых программируемых машин было создано более восьми тысяч языков программирования (включая эзотерические, визуальные и игрушечные). Каждый год их число увеличивается. Некоторыми языками умеет пользоваться только небольшое число их собственных разработчиков, другие становятся известны миллионам людей. Профессиональные программисты могут владеть десятком и более разных языков программирования.

*Цели данного исследования*:

1. Обозначить роль программирования в современном мире.
2. Показать важность профессии «программист».
3. Создать компьютерную игру с помощью языка программирования.

*Задачи исследования*:

1. Исследовать историю создания языков программирования.
2. Определить актуальность профессии «программист».
3. Определить, какое место в жизни современного человека занимает программирование.

*Объект исследования* – языки программирования.

*Гипотез*а: возможно ли создать игру при помощи языков программирования.

Актуальность исследования обусловлена тем, что сейчас широко распространены языки программирования, с их помощью создано всё программное обеспечение компьютера, а также является востребованной такая профессия, как программист, практически во всех сферах деятельности.

Практическая значимость: исследование предоставляет дополнительный материал об истории развития языков программирования, способствует повышению интереса к программированию, показывая, что даже школьник может создавать свои программные продукты, которые могут быть созданы для разнообразных целей.

**Глава 1. История развития языков программирования**

Каким бы совершенным ни был компьютер, без программного обеспечения он представляет собой просто груду металла и пластика. Вычислительные устройства сейчас находятся повсюду: в сотовых телефонах и банкоматах, в станках с числовым программным управлением и в телевизорах. Трудно найти сферу жизни, в которой они не были бы тем или иным образом задействованы. И все эти устройства работают благодаря программам, написанным с помощью тех или иных языков программирования. Именно программы определяют, что и как делает компьютер, в какой последовательности он выполняет те или иные операции. Первые языки программирования начали появляться в начале пятидесятых годов и использовались для преобразования простых арифметических выражений в машинный код. Машинный код – это система команд вычислительной машины, интерпретируемых непосредственно микропроцессором. Но человеку писать программу в машинных кодах очень неудобно. Для того чтобы облегчить труд программиста, и начали создаваться языки программирования.

**Язык программирования** – формальный язык, предназначенный для записи [компьютерных программ](https://ru.wikipedia.org/wiki/Компьютерная_программа). Он определяет набор [лексических](https://ru.wikipedia.org/wiki/Лексика), синтаксических и [семантических](https://ru.wikipedia.org/wiki/Семантика_(программирование)) правил, задающий внешний вид программы и действия, которые выполнит исполнитель под её управлением.

Языки программирования делятся на языки высокого уровня и низкого. Чем выше уровень языка, тем легче на нем писать программисту. Такой язык более понятен человеку, так как позволяет с помощью простых смысловых конструкций задавать необходимую последовательность действий. После создания программы происходит ее компиляция – то есть автоматический перевод в понятный процессору язык машинных кодов. Языки низкого уровня находятся гораздо ближе к языку машинных кодов, поэтому писать на них труднее. Но у них есть свое преимущество – написанные на таком языке программы получаются очень быстрыми и компактными. Наиболее популярным низкоуровневым языком является Assembler. Некоторые его преимущества настолько очевидны, что даже в сложных программах, написанных на высокоуровневых языках, часто применяют вставки на Ассемблере. Несмотря на существование большого количества языков программирования, получившие широкое распространение можно пересчитать по пальцам. Одним из самых распространенных является язык C++. Это очень удобный и достаточно простой для программиста язык, позволяющий создавать программы любого уровня сложности. Не так давно компания Microsoft разработала язык C# (читается, как «си шарп»), обладающий рядом новых возможностей и предназначенный для написания программ под операционную систему Windows. Компания Microsoft выпустила и очень популярную среду программирования Microsoft Visual Studio, позволяющую программировать на С++, С# и некоторых других языках. Очень известным является язык программирования Delphi. Свое происхождение он ведет от некогда знаменитого Паскаля, однако благодаря усилиям компании Borland приобрел ряд новых качеств, став, по сути, новым языком. Писать на этом языке достаточно просто и удобно, а благодаря среде программирования Borland Delphi он получил очень широкое распространение. Без языков программирования было бы невозможным и существование интернета. Такие языки, как Perl и PHP позволяют создавать скрипты, определяющие выполнение на страницах сайта необходимых действий. Даже создание самой простой интернет-страницы невозможно без знания HTML – стандартного языка разметки документов.

### **1.1 Java**

История Java восходит к 1991 году, когда группа инженеров из компании Sun под руководством Патрика Нотона (Patrick Naughton) и члена Совета директоров (и разностороннего компьютерного волшебника) Джеймса Гослинга (James Gosling) занялась разработкой небольшого языка, который можно было бы использовать для программирования бытовых устройств, например, контроллеров для переключения каналов кабельного телевидения (cable TV switchboxes).

Поскольку такие устройства не потребляют много энергии и не имеют больших микросхем памяти, язык маленьким и генерировать очень компактные программы. Кроме того, поскольку разные производители могут выбирать разные центральные процессоры, было важно не привязываться к какой-то одной архитектуре компьютеров. Проект получил кодовое название "Green".

Стремясь изобрести небольшой, компактный и машинно-независимый код, разработчики возродили модель, использованную при реализации первых версий языка Pascal заре эры персональных компьютеров. Никлаус Вирт, создатель языка Pascal, в свое время разработал машинонезависимый язык, генерирующий промежуточный код для некоей гипотетической машины. Этот язык стал коммерческим продуктом под названием UCSD Pascal. (Такие гипотетические машины часто называются виртуальными –

например, виртуальная машина языка Java, или JVM.)

Этот промежуточный код можно выполнять на любой машине, имеющей соответствующий интерпретатор. Инженеры, работавшие над проектом "Green", также использовали виртуальную машину, что решило их основную проблему.

Однако большинство сотрудников компании Sun имели опыт работы с операционной системой UNIX, поэтому в основу разрабатываемого ими языка был положен язык C++, а не Pascal. В частности, они сделали язык объектно-, а не процедурно-ориентированным.

Как сказал Гослинг в своем интервью: "*Язык — это всегда средство, а не цель*". Сначала Гослинг решил назвать его "Oak" ("Дуб"). Однако слово Oak уже используется в качестве имени ранее созданного языка программирования, и изменили название на Java.

Тем временем в рамках Интернет разрасталась сеть World Wide Web. Ключом к этой сети является браузер, превращающий гипертекст в изображение на экране.

В 1994 году большинство людей пользовалось браузером Mosaic, некоммерческим Web-браузером, разработанным в суперкомпьютерном центре Университета штата Иллинойс в 1993 году.

Однако, была цель создания браузера должен представлять собой одно из немногих приложений клиент-серверной технологии, в которой жизненно важным было сделать архитектурную независимость, выполнение в реальном времени, надежность, безопасность, не являлось чрезвычайно важными для рабочих станций.

На самом деле браузер был разработан Патриком Нотоном и Джонатаном Пэйном. Позднее он превратился в современный браузер HotJava. Этот браузер был написан на языке Java, чтобы продемонстрировать всю его мощь. Однако разработчики не забывали о мощных средствах, которые теперь называются апплетами, наделив свой браузер способностью выполнять код внутри Web-страниц. Демонстрация технологии была представлена на выставке Sun World '95 23 мая 1995 года.

Компания Sun выпустила первую версию языка Java в начале 1996 года. Через несколько месяцев после нее появилась версия Java 1.02. Она не подходит для разработки серьезных приложений, хотя её можно применять для разработки Web-страниц с пляшущими человечками, однако в версии Java 1.02 ничего нельзя даже напечатать.

Версия Java 1.1, заполнила большинство зияющих провалов, намного улучшив возможность отражения и добавив новую модель событий для программирования графического пользовательского интерфейса. Несмотря на это, она все еще была довольно ограниченной.

Java 2 Standart Edition Software Development Kit Version 1.2 (Стандартное издание пакета инструментальных средств для разработки программного обеспечения на языке Java 2, версия 1.2).

Кроме стандартного издания пакета ("Standart Edition") были предложены еще два варианта: "микроиздание" ("Micro Edition") для портативных устройств, например, для мобильных телефонов, и. "промышленное издание" ("Enterprise Edition") для создания сер верных приложений.

Версии 1.3 и 1.4 стандартного издания пакета инструментальных средств намного совершеннее первоначального выпуска языка Java 2. Они обладают новыми возможностями и, разумеется, содержат намного меньше ошибок.

**Программы, созданные на языке Java**

Главный плюс Java – принцип «написано однажды – работает везде». Это значит, что ПО, написанное на одной платформе, будет запускаться и на других устройствах. Java используется для создания приложений для мобильных устройств, удаленных процессоров, беспроводных модулей, датчиков, да и в целом – практически любых электроустройств. Адаптивность – одна из причин, по которой Twitter перешел на JVM. Java – в числе самых популярных языков на GitHub по количеству коммитов. Язык востребован благодаря огромному выбору библиотек под любые задачи постоянно их количество наращивается. Таким образом, Java постоянно развивается и создателями языка, и его «пользователями». Благодаря разнообразию библиотек, этот ЯП – гибкий, поэтому он прекрасно подходит для реализации новых функций. И для больших корпораций, и для компаний меньшего масштаба это ценно. Так, гибкость языка помогла компании Spotify построить набор модулей для разработки микросервисов Apollo. Основные бренды, которые используют Java:

* ***Facebook*** – крупнейшая социальная сеть в мире.
* ***Twitter*** – социальная сеть для публичного обмена сообщениями при помощи веб-интерфейса.
* ***Netflix*** – американская развлекательная компания, поставщик фильмов и сериалов на основе потокового мультимедиа.
* ***YouTube*** – видеохостинг, предоставляющий пользователям услуги хранения, доставки и показа видео.
* ***PayPal*** – крупнейшая дебетовая электронная платёжная система.
* ***Amazon***– американская компания, крупнейшая в мире на рынках платформ электронной коммерции и публично-облачных вычислений по выручке и рыночной капитализации.

**1.2 С++**

С++ – один из наиболее популярных языков программирования, используемый для разработки программного обеспечения, поддерживающий процедурное (при котором последовательно выполняемые команды можно собрать в подпрограммы), объектно-ориентированное (стиль кодирования, который позволяет группировать задачи в классы), обобщенное программирование (написание кода, который может быть многократно использован с объектами различных типов), сочетающий свойства высокоуровневых и низкоуровневых языков. С++ используется для создания программ, рассчитанных на выполнение определенных задач в разных сферах жизни человека, игр, драйверов, операционных систем и прочего. Он образован на основе языка С. Многие программы на С исправно работают и с компилятором С++ из-за тесной связи.

Название C++ происходит от языка программирования C, в котором унарный оператор ++ обозначает увеличение значения переменной на 1. Сначала этот язык носил имя С с классами, затем стал называться C++. После буквы С добавлено именно два плюса, потому что один образует синтаксическую ошибку в языке С, а также это имя уже было занято.

С++ возник в начале 1980-х годов. Бьёрн Страуструп, сотрудник фирмы Bell Laboratories, усовершенствовал язык С под свои нужды. То есть, изначально создание нового языка он не планировал. Но в конце 70-х годов программист начал работать над задачами теории очередей (в моделировании телефонных вызовов), он обнаружил, что попытки применения действующих тогда языков моделирования, таких, как Симула и BCPL, оказываются неэффективными, а применение высокоэффективных машинных языков слишком сложно. Так, язык Симула имеет полезные для разработки большого программного обеспечения возможности, но все работает чересчур медленно, а язык BCPL достаточно быстр, но сильно близок к низкоуровневым языкам, он не подходит для разработки больших программ.

Для создания нового языка программирования самой лучшей основой послужил язык С, так как он универсален, прост, переносим, быстр и функционален в исполнении многих задач, стыкуется со средой программирования UNIX. Страуструп добавил к С возможность работы с типами данных (классами) и объектами. Благодаря быстродействию С практические задачи моделирования оказались доступными для решения по времени разработки и вычислений. В первую очередь в C были добавлены типы данных, их наследование, проверка типов и аргументы по умолчанию. Первые версии языка, изначально называвшегося С с классами, стали доступны с 1980 года. Затем Страуструп создал компилятор cfront, который перерабатывал код С++ в код простого С, что позволило сразу применять его на практике.

Язык стал невероятно популярен. К 1983 году в него были добавлены различные константы, виртуальные функции, ссылки, перегрузка функций и операторов, контроль пользователя над управлением свободной памятью, улучшенная проверка типов и новый стиль комментариев //. В 1985 году вышло издание языка программирования C++, это было первое описание языка, что из-за отсутствия официального стандарта было очень важно. В 1989 году состоялся выход второй версии C с новыми возможностями наследования, функциями и классами. В 1998 году был утвержден международный стандарт языка C++. Он является свободным, никто не обладает правом на владение им.

Стандарт C++ состоит из двух основных частей: описание стандартного сборника подпрограмм или объектов, используемых для разработки программ, и описание ядра языка. Кроме того, существует огромное количество нестандартных библиотек C++. В программах на C++ можно использовать многие библиотеки C.

Нововведениями C++ в сравнении с C являются:

* поддержка объектно-ориентированного программирования;
* поддержка обобщённого программирования;
* новые типы данных;
* исключения - реакции программы на ошибки;
* пространства имён (хранилища для группировки уникальных идентификаторов);
* встраиваемые функции;
* перегрузка операторов (существования в одной области видимости нескольких различных вариантов применения оператора);
* перегрузка имён функций;
* ссылки и операторы управления свободно распределяемой памятью;
* дополнения к стандартной библиотеке.

Также часто сравниваются Java и C++ как языки, которые унаследовали синтаксис С, несмотря на большие различия на всех уровнях. Java используется в конкретном секторе промышленности: безопасный язык с низким порогом вхождения для разработки прикладных приложений широкого рынка с высокими показателями адаптации к другой среде. С++ претендует на универсальное применение во всех задачах для всех категорий программистов, но не удовлетворяет в полной мере требованиям ни одной из заявленных сфер применимости. Базовая библиотека С++ в сравнении с Java имеет меньше ресурсов, но предоставляет свободу выбора сторонних библиотек.

**Программы, созданные на языке С++:**

* операционные системы;
* системы управления базами данных;
* компиляторы и интерпретаторы;
* драйверы;
* микроконтроллеры.

**1.3 С#**

Начало нового тысячелетия Microsoft решила отметить выпуском новоявленных программных продуктов. К 2000-му году компанией были подготовлены промышленные модификации новых решений и компонентных технологий в сфере обмена сообщениями и информацией, а также изготовления Internet-приложений (ASP+, COM+, SOAP, ADO+, Biztalk Framework). С целью поддержки этих новшеств компания Microsoft выпустила платформу .NET - инструментарий для разработки приложений. Эта платформа совместила «под общей крышей» несколько языков программирования, что для того времени было в новинку.

Технология активных серверных страниц (Active Server Page) ASP.NET также стала ещё одним нововведением платформы. Посредством неё можно было довольно быстро разработать взаимодействующие с базами данных веб-приложения.

C# – язык программирования, созданный специально для ASP.NET. Следует отметить, что и сама ASP.NET также была полностью написана на этом языке.

Название «Си шарп» (в пер. с англ. sharp – диез). В музыкальной нотации знак «#» читается как «диез» и обозначает повышение на полтона высоты звука. Можно рассмотреть образование названия «C#» и с другой стороны, как производное следующей череды трансформаций: C → C++ → C++++(C#), поскольку символ «#» является совокупностью 4-х знаков «+».

Поскольку есть технические ограничения на отображение (браузеры, стандартные шрифты и т. д.) а также по причине того, что знак диез ♯ на стандартной клавиатуре не представлен, был выбран знак # для представления знака диез при написании имени языка программирования. Зафиксировали это соглашение в Спецификации Языка C# ECMA-334. Поскольку названия языков программирования переводить не принято, язык следует по-английски называть «Си шарп».

Авторы этого языка программирования – Вилтамут Скотт и Хейльсберг Андерс – создатель Дельфи и Турбо Паскаля.

Языком C# поддерживаются все три основных составляющих объектно-ориентированного программирования: полиморфизм, наследование и инкапсуляция. Кроме того, в нем реализован замысел автоматической «сборки мусора», динамического связывания и обработки исключений.

C#, как и Java, предназначался изначально для веб-разработки, и около 75% его синтаксических возможностей подобны языку программирования Java. Кроме того, его называют «очищенной разновидностью Java». 5% позаимствовано из Visual Basic, а 10% из C++. Примерно 10% C# – это воплощение собственных задумок разработчиков. Несмотря на весьма значительные различия между компонентной объектной формой COM (основной стандарт Microsoft для реализации программного обеспечения и компонентного проектирования) и моделью Java, рассматриваемые языки программирования во многом сходны между собой.

Единая среда исполнения программ базируется на применении промежуточного языка IL (с англ. Intermediate Language – промежуточный язык), выполняющего практически ту же функцию, что и байт-код виртуальной машины Java. Применяемые в пределах технологии .NET компиляторы *т*ранслируют с различных языков программы в IL-код. IL-код по такому же принципу, как и байт-код Java, являет собой команды гипотетической стековой вычислительной машины. Однако в устройстве и применении IL имеются и различия.

В первую очередь, IL в отличие от JVM, к одному языку программирования не привязан. Предварительные версии Microsoft.NET включают в себя компиляторы языков Visual Basic, Си#, Си++. Независимыми разработчиками могут добавляться другие языки, при этом, они могут создавать свои компиляторы в IL-код.

Кроме того, IL предназначается не для программной интерпретации, а для дальнейшей компиляции в машинный код. Такое применение промежуточного языка позволяет достичь гораздо большего быстродействия программ. Файлы, содержащие IL-код, несут достаточно данных для функционирования оптимизирующего компилятора.

## 

**Глава 2. Программирование в современном мире**

**2.1. Описание профессии «программист»**

Программист – это специалист, занимающийся разработкой программного обеспечения (ПО) для персональных, встраиваемых, промышленных и других разновидностей компьютеров, то есть программированием. Это может быть, как руководитель крупных проектов по разработке ПО, так и «одиночка», пишущий код конкретной программой.

Перед большинством предприятий и организаций вопрос автоматизации, и многие управленцы стремятся использовать достижения современных информационных технологий, поэтому программист является своего рода консультантом, выполняющим посредническую функцию между тем, чего желает получить руководитель, и тем, что предлагает на данный момент мир высоких технологий.

Различают системных и прикладных программистов. Прикладной – это специалист, который осуществляет разработку и отладку программ для решения различных задач. Системный – специалист, занимающийся разработкой, эксплуатацией и сопровождением системного программного обеспечения.

**Личные качества**

Практически ежемесячно выходят новые версии программ, обновляются характеристики оборудования, и специалисты в области информационных технологий должны быть всегда в курсе этих изменений. Поэтому способность к самообучению – один из главных навыков, которым должен обладать программист.

Владение английским языком на уровне чтения технической документации является еще одним обязательным требованием, предъявляемым к представителям этой профессии. Для таких специалистов очень важно умение работать: в команде, над большими проектами, со средствами коллективной разработки, с крупными финансовыми системами (бюджетными, банковскими, управленческого учета). Для претендентов на позицию ведущего программиста желательны навыки управления проектами и коллективом, самостоятельность, инициативность, а также способность нести личную ответственность за поставленную задачу.

**Образование**

Программистов готовят во многих высших учебных заведениях. Хотя в этой сфере успешно работает и множество специалистов с другим образованием, а иногда даже вообще без высшего образования.

Много хороших программистов – самоучки.

Самым важным в процессе обучения является не столько прослушивание лекций, сколько решение задач и участие в различных учебных проектах. Будущему программисту важно научиться думать, находить верные решения. Информационные технологии развиваются очень быстро, поэтому для таких специалистов необходимы навыки самообразования, чтобы после получения основного диплома идти в ногу со временем и не терять квалификации. Специалисту в этой области нужно знать языки программирования, уметь создавать веб-страницы, владеть иностранным языком (английским как минимум) и постоянно совершенствовать свои знания, так как через несколько лет он может оказаться не удел. Не все программисты создают сайты. Есть те, которые разрабатывают программы специально для работы конкретного предприятия (прикладные программисты).

**Место работы и карьера**

В основном в России требуются прикладные программисты практически любая компания сегодня нуждается в автоматизации своего бизнеса, в возможности работать по внутренней компьютерной сети и быстро обмениваться информацией в электронном виде между подразделениями и отделами, вести учет продукции, закупок, реализованного товара.

Они адаптируют, а если нужно, и разрабатывают пакет специальных программ, учитывая особенности бизнеса организации: одно дело автоматизировать деятельность издательского дома, другое – торговой фирмы или ресторана.

Однако многим предприятиям требуется не только автоматизация бизнеса, но и постоянный мониторинг налаженной системы, ее изменение в случае расширения компании, поэтому работодатели готовы к зачислению программистов к себе в штат.

Все же некоторые организации предпочитают заказывать автоматизацию своего бизнеса специализированным ИТ-компаниям, предоставляющим такие услуги.

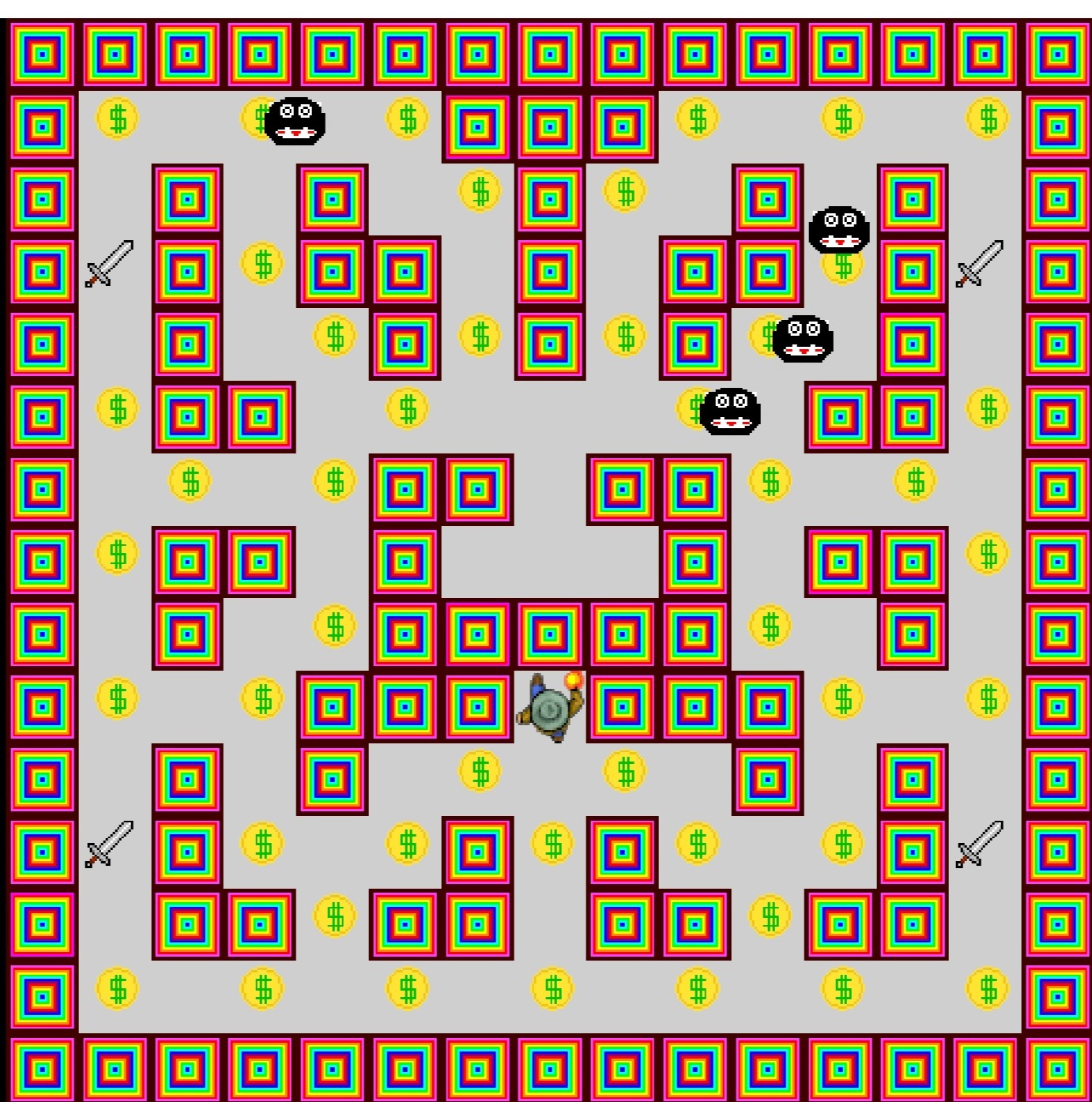
Еще одна сфера деятельности программиста – web-программирование: разработка, модернизация и поддержка сайтов. Эта область требует многих специфических знаний – PHP, MYSQL, XHTML, CSS, JavaScript, XML. Причем очень часто работодатели желают получить профессионала «два в одном» – web-дизайнера и web-программиста, чтобы он не только решал технические вопросы (быстрая загрузка сайта, работа ссылок, поисковика), но также занимался и оформлением сайта, а иногда даже и его информационным наполнением.

Можно основать собственный бизнес, который потенциально может оказаться очень доходным (достаточно вспомнить, что Билла Гейтса – основателя корпорации Microsoft). Однако для того, чтобы успешно управлять таким бизнесом, недостаточно уметь создавать хорошие компьютерные программы; для этого требуются основательные знания в области экономики, юриспруденции, менеджмента.

**2.2 Создание игры с помощью Game Maker**

Для начала я хотел узнать сколько людей проводят своё свободное время за компьютерными играми. Для этого мне понадобилось провести опрос. По результатам которого я **выяснил, что** **из всех людей,** прошедших опрос, (в опросе участвовало 151 человек) 37% предпочитают провести своё свободное время, за компьютерными играми. Выше находиться только люди, которые предпочитают в свободное время гулять с друзьями (47%) или общаться в социальных **сетях (**43%).

Для создания игры мне потребовалась программа «Game Maker». С помощью функционала программы мне удалось сделать игру «Qzavrik». Для начала я сделал модели главного персонажа, модели бонусов и врагов, а также оружия. После создания моделей я написал скрипты персонажей.  
Цель игры – собрать все монеты на уровне, после того как это будет сделано, вас перемешает на следующей уровень.



Игра имеет очень низкие системные требования, поэтому сможет запуститься на любом ПК.

Данная разработка относится к классу арканных игр.

Классические аркады характеризуются следующими свойствами:

Игра на одном экране. В классических аркадах весь игровой процесс сосредоточен на одном экране. Прежде всего это обусловлено исторически, что произошло из-за технических ограничений, но в то же время это значительно влияло на геймдизайн. Так, игроки в любой момент времени могли видеть весь игровой мир и принимать решения, исходя из полной информации о его состоянии. Множество игр жанра имело более одного экрана, и они сменяли друг друга как уровни. Характерными примерами здесь являются Joust, Pac-Man, Mario Bros.[8]

Бесконечная игра. Потенциально игроки могут играть в аркаду бесконечное время, и соответственно, не могут выиграть. Это влияло на то, что игроки делали вызов сами себе – насколько долго они смогут продержаться. Относительно геймдизайна в аркадах игрок никогда не выигрывал, и каждая игра заканчивалась поражением. В то же время, игры проектировались таким образом, чтобы со временем игроку становилось всё сложнее, и таким образом бесконечная игра предлагала бесконечную сложность. Данная ситуация изменилась с появлением рынка домашних компьютеров, когда издатели изменили своё желание, чтобы игроки проходили игру и после этого хотели купить новую. [9]

Игровой счёт. Практически все классические аркады включают в себя игровой счёт, когда игрок получает очки за выполнение различных целей или задач. Здесь имеется другой важный концепт аркад, унаследованный от игр пинбола: очки позволяют игроку понять, насколько хорошо он играл, и несмотря на то, что выиграть невозможно. При этом типичное время игры среднего игрока составляет около двух минут, а у опытного до десятков минут. На основании данной особенности у аркад как правило есть таблица рекордов, где игрок может рядом со своих результатов ввести свои инициалы, и тем самым сравнивать себя с другими игроками, и соревноваться. [10]

Быстрое обучение, простой игровой процесс. Для классических аркад характерно то, что игрокам легко научиться геймплею, но становится практически невозможным стать мастером в игре из-за её сложности. Вместе с тем, если игрок погибает в аркаде, то это практически всегда происходит по его вине. В таких играх нет «специальных комбинаций клавиш», которые игрок должен выучить из документации для того, чтобы сделать что-то особенное. Очень мало игр расширяют концепт посредством очков здоровья, щитов или таблеток силы. Это связано с тем, что с коммерческой точки зрения аркадам было необходимо охватить как можно больший спектр игроков, то есть, фактически каждый человек в магазине должен быть способным подойти и попробовать сыграть. В то же время, простой игровой процесс не подразумевает что он «плохой» или «ограниченный», – он может быть «элегантным» и «отполированным».

Нет сюжета или истории. Классические аркады практически всегда избегали попыток рассказать какую-либо историю, и данная тенденция продолжается для современных аркад. Играм жанра всегда требовалось, чтобы игроки быстро поняли, что происходит – это научная фантастика, война, спорт или что-то ещё. Множество аркад создавало свои уникальные сеттинги, например, Pac-Man и Q\*Bert. Геймдизайнеры классических аркадных игр не чувствовали, что им нужно наполнять свои миры чем-то. [11]

**Заключение**

Сейчас нам необходимы языки программирования для повседневной жизни, к которой мы привыкли. Так как без языков программирования не было ни ПК, ни программ ничего что связано с компьютерами. Мы бы не общались в соц. сетях, не покупали вещи в интернет магазинах. Не зарабатывали с помощью интернета и компьютера. Я считаю, что языки программирования имеют очень большую роль в жизни каждого человека.

Изучение истории языков программирования, их разнообразия и особенностей позволяет программисту сделать правильный выбор при выборе языка для решения определенной задачи.

В ходе выполнения работы была обозначена роль программирования в современном мире, показана важность профессии программист.

Результатом исследования стало собственное прикладное программное обеспечение – игра, созданная с помощью Game Maker.

Данная разработка носит как развлекательный, так и развивающий характер, поскольку компьютерные игры в разной степени развивают координацию и сосредоточенность, сообразительность, логическое мышление, творческие способности.

Важно, что этот класс игры (аркадная) с простой графикой и звуком наименее опасен и не вызывает длительной привязанности.

Само программирование является очень приятным, удобным, а главное – эффективным способом саморазвития, которое очень важно для достижения новых высот.

Цели и задачи исследования были достигнуты, гипотеза подтверждена.

**Список литературы**

1. URL: <https://otus.ru/nest/post/682/> (дата обращения: 13.01.2020).
2. URL: <https://ru.wikipedia.org/wiki/Java> (дата обращения: 23.11.2020).
3. URL:<https://ru.wikipedia.org/wiki/Си_(язык_программирования)> (дата обращения: 13.01.2020).
4. Зимина К. И. Положительное влияние компьютерных игр на развитие подростков // Современная психология: материалы II Междунар. науч. конф. (г. Пермь, июль 2014 г.). – Пермь: Меркурий, 2014. – С. 43-45. – URL: https://moluch.ru/conf/psy/archive/111/5899/ (дата обращения: 17.12.2020).
5. Страуступ, Б. Язык программирования С++. Специальное издание / Б. Страуступ. – М.: Бином, 2015. – 1136 c.
6. Теренс Пратт*.* Языки программирования: разработка и реализация/ Programming Language Design and Implementation (PLDI). – 1-е издание. – МИР, 1979.
7. Троелсен, Э. Язык программирования С# 5.0 и платформа .NET 4.5 / Э. Троелсен; Пер. с англ. Ю.Н. Артеменко. – М.: Вильямс, 2013. – 1312c.
8. Фридман, А.Л. Основы объектно-ориентированного программирования на языке Си++ / А.Л. Фридман. – М.: Гор. линия-Телеком, 2012. – 234 c.
9. Хейлсберг, А. Язык программирования C#. Классика Computers Science / А. Хейлсберг, М. Торгерсен, С. Вилтамут. – СПб.: Питер, 2012. – 784 c. Бьянкуцци, Ф. Пионеры программирования. Диалоги с создателями наиболее популярных языков программирования / Ф. Бьянкуцци, Ш. Уорден. – М.: Символ, 2011. – 608 c.