МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вятский государственный университет»**

**( «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Допущено к защите

Руководитель проекта

\_\_\_\_\_\_\_\_/Клюкин В.Л./

(подпись) (Ф.И.О)

«\_\_»\_\_\_\_\_\_\_\_\_\_2015г.

Разработка компьютерной игры

Пояснительная записка

Курсовая работа по дисциплине

«Компьютерная графика»

ТПЖА 09.03.01.024 ПЗ

Разработал студент группы ИВТ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Щесняк Д.С./

Руководитель доцент кафедры ЭВМ \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Клюкин В. Л./

Проект защищен с оценкой «\_\_\_\_\_\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (дата)

Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись) (Ф.И.О)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Киров 2015

Содержание

[1 Введение 3](#_Toc437194091)

[2 Анализ аналогов 4](#_Toc437194092)

[2.1 CodeCombat 4](#_Toc437194093)

[2.2 CodeMonkey 5](#_Toc437194094)

[2.3 Bit’s Quest 6](#_Toc437194095)

[2.4 Ruby Warrior 7](#_Toc437194096)

[3 Постановка задачи 9](#_Toc437194097)

[3.1 Требования к аппаратному обеспечению. 10](#_Toc437194098)

[3.2 Требования к программному обеспечению. 10](#_Toc437194099)

[3.3 Требования к интерфейсу 11](#_Toc437194100)

[4 Схема взаимодействия функциональных блоков. 11](#_Toc437194101)

[4.1 Блок пользовательских настроек 11](#_Toc437194102)

[4.2 Блок игрового процесса 12](#_Toc437194103)

[4.3 Блок визуализации 12](#_Toc437194104)

[5 Математический аппарат 13](#_Toc437194105)

[6 Разработка схем алгоритмов 14](#_Toc437194106)

[7 Разработка программного обеспечения 14](#_Toc437194107)

[7.1 Разработка модульной структуры 14](#_Toc437194108)

# **Введение**

В наше время мы все чаще сталкиваемся с электронными вычислительными машинами, они буквально окружают нас, они используются во всех отраслях. Персональные компьютеры, ноутбуки, планшеты, телефоны, все это все сильнее входит в нашу жизнь. В 21 – м веке очень важно уметь владеть вычислительном машиной не на уровне пользователя, а на уровне программиста, который может создать любое программное обеспечение.

Все выше и выше поднимается уровень языков программирования. В настоящее время для того, чтобы уметь программировать не обязательно знать о тонкостях работы процессора, оперативной памяти и другого аппаратного обеспечения ЭВМ, достаточно уметь создавать алгоритмы и знать синтаксис языка.

Одним из примеров языка высокого уровня является python. Синтаксис языка очень минималистичен, но это не мешает ему пользоваться популярностью в очень разных областях разработки программного обеспечения.

Существует множество способов изучения языков программирования, по моему мнению, наиболее эффективным методом является метод интерактивного обучения: когда пользователь изучает язык программирования и параллельно с написание кода программы видит результат своей работы. Для обучения базовому синтаксису языка, отлично подходит игровая форма, т. к. она может наглядно продемонстрировать как именно работает программа. Поэтому в качестве программного обеспечения для обучения языку программирования python было выбрано именно игровое приложение.

# **Анализ аналогов**

## CodeCombat

Красочная игра c интересным и захватывающим сюжетом. Вам предстоит стать волшебником, который c помощью js-кода меняет окружающий мир. Для игры требуется обязательное подключение к интернету, по причине того, что она браузерная. Имеется поддержка мультиплеера. Поддерживает такие скриптовые языки как javaScript, Python, Lua и экспериментальные языки как CoffeScript. Скриншот игры приведен на рисунке 1. Данная игра находится по адресу http://codecombat.com/

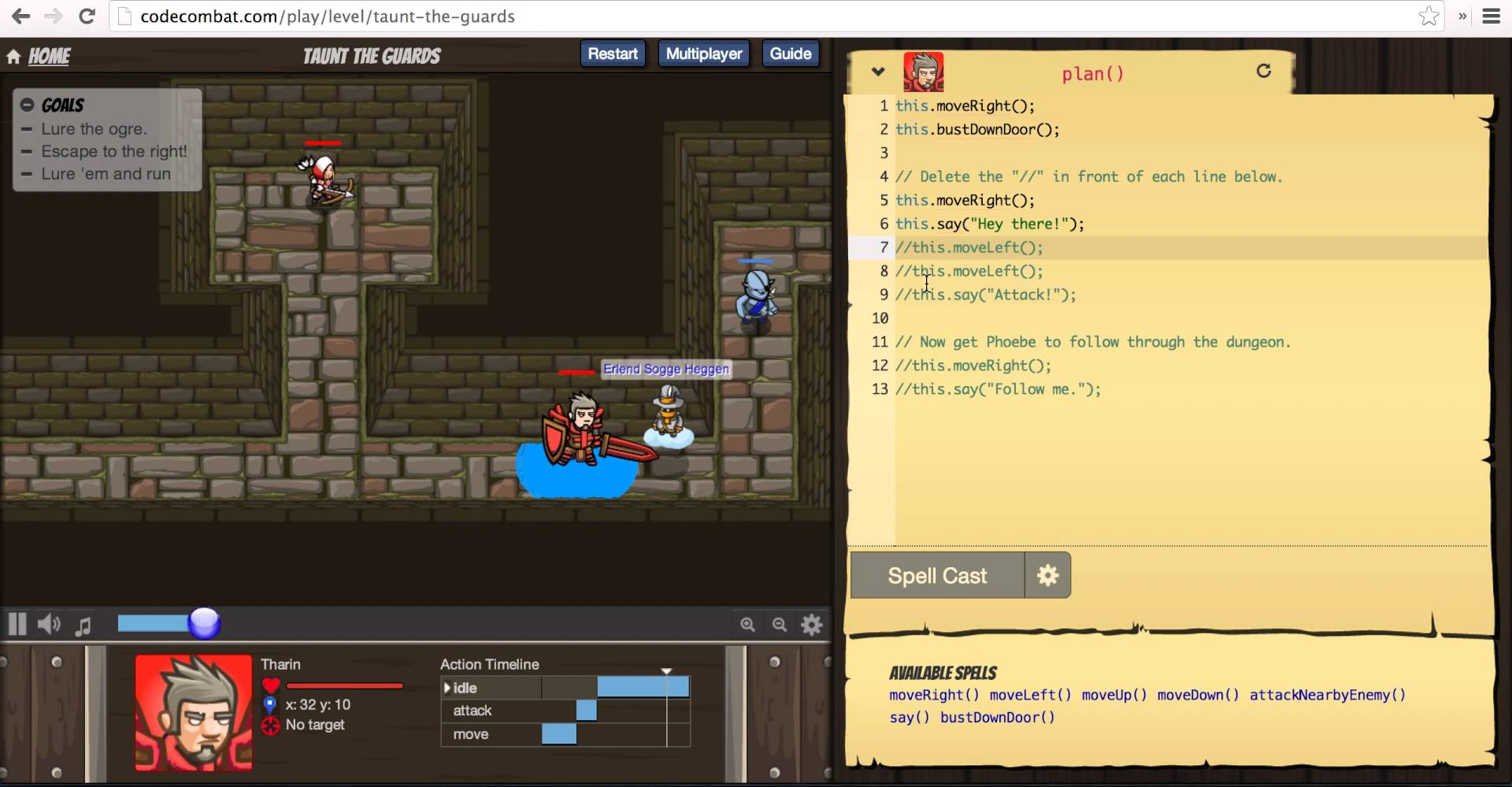


Рисунок 1 – Окно игры CodeCombat

## CodeMonkey

CodeMonkey — это простая онлайновая игра, в процессе прохождения которой можно познакомиться с основами программирования. Весь процесс разделён на отдельные уровни. В любой момент можно прервать прохождение (прогресс сохраняется) или вернуться на несколько шагов назад. Программирование происходит при помощи пседвоязыка. Обязательное интернет подключение, отсутствует мультиплеер. На рисунке 2 представлен скриншот из данной игры. Данная игра находится по адресу https://www.playcodemonkey.com/

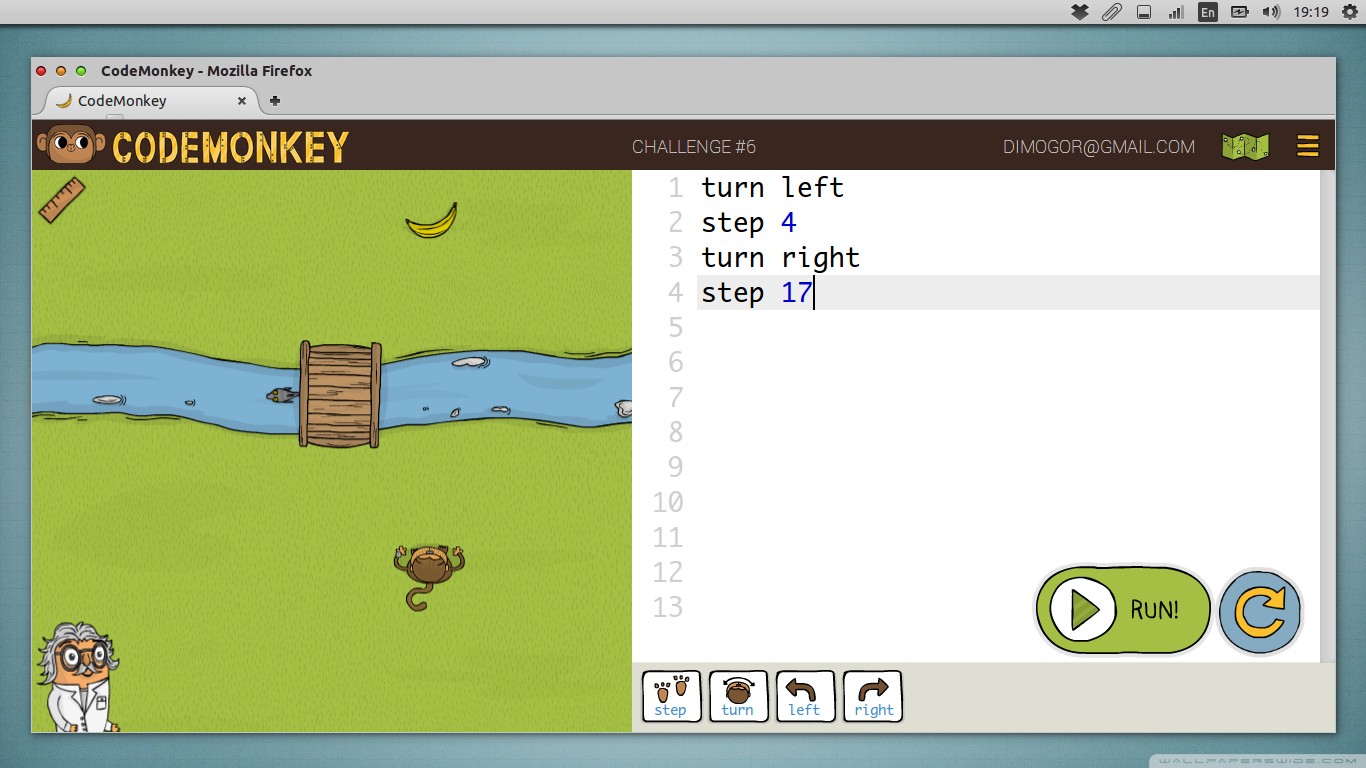


Рисунок 2 – окно игры CodeMonkey

## Bit’s Quest

В игре нужно будет с помощью JavaScript-кода управлять ботом, выполняя задания к каждому уровню. Самое частое задание — добраться до выхода из лабиринта, но иногда приходится сражаться с вражеским кораблем.

Вам предстоит разобраться, что такое события, как использовать функции, и может понадобиться применить [замыкания](http://www.youtube.com/watch?v=r-W_BUG3EbI). Для игры требуется обязательное подключение к интернету. На рисунке 3 представлен скриншот окна игры. Данная игра находится по адресу http://bitsquest.bitbucket.org/



Рисунок 3 – главное окно игры Bit’s Quest

## Ruby Warrior

Сценарий игры: руби-воин должен убить всех врагов. Управлять героем нужно с помощью Ruby-кода.

Для прохождения нужно разбираться в коде, но знаний по Ruby будет достаточно самых базовых. Вас не будут учить программировать, зато заставят решать логические задачи, причем иногда очень сложных. Главное окно программы представлено на рисунке 4. Данная игра находится по адресу https://www.bloc.io/ruby-warrior/

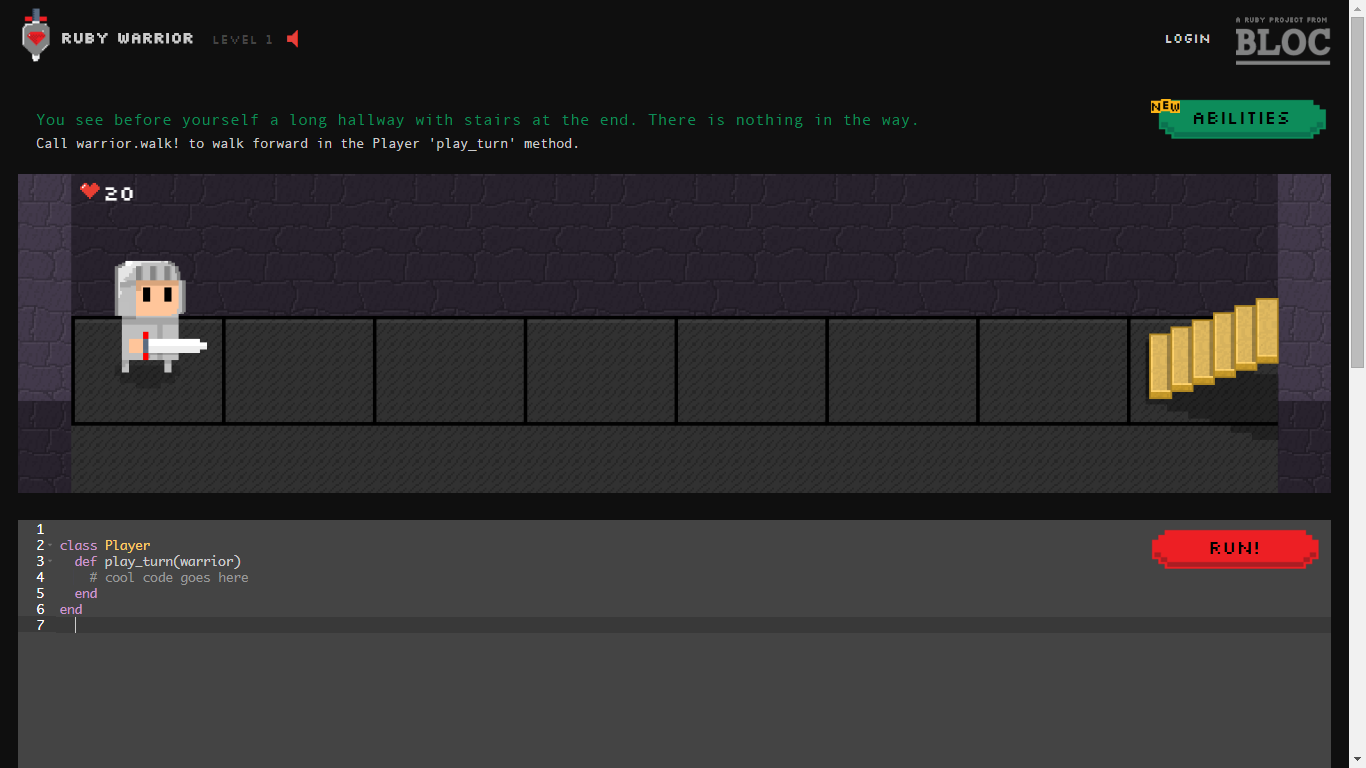


Рисунок 4 – главное окно программы Ruby Warrior

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерий  Аналог | Низкий порог вхождения | Обязательное постоянное интернет соединение | Наличие многопользовательской игры | Обучение осуществляется на реально существующих языках |
| CodeCombat | **+** | **+** | **+** | **+** |
| CodeMonkey | **+** | **+** | **-** | **-** |
| Bit’s Quest | **-** | **+** | **-** | **+** |
| Ruby Warrior | - | + | - | + |

Таблица 1 сравнение аналогов

# **Постановка задачи**

На основе анализов аналогов можно установить критерии, которым должен удовлетворять конечный продукт для возможности конкурирования с представленными аналогами.

Выходное программное обеспечение должно:

* Иметь низкий порог вхождения пользователей.
* Использовать в качестве обучающего языка реальный язык программирования.
* Минимально зависеть от интернет соединения конечного пользователя.
* Иметь возможность многопользовательской игры, а именно сервер логических битв.

В качестве обучающего языка программирования был выбран Python. Python – язык программирования высокого уровня общего назначения, который ориентирован на уменьшение количества кода, сохраняя при этом его читаемость.

Плюсы python:

* Легкий для понимания синтаксис научит пользователя блочной модели программирования
* Огромное количество библиотек для расширения функционала
* Возможность перенести программу на любую платформу на которой есть интерпретатор.
* Благодаря нестрогой типизации программы на python создаются намного быстрее чем на C++/pascal.

Итог: Конечный продукт должен быть доступен как можно большему числу пользователей, другими словами, должен иметь низкий порог вхождения по знаниям в программировании, должен минимально зависеть от интернет – соединения пользователя и иметь возможность многопользовательской игры. В качестве обучающего языка использоваться интерпретатор языка python 3.

## **Требования к аппаратному обеспечению.**

Процессор: Pentium 4/Athlon XP 3 ГГц

ОЗУ: 1 Гб памяти

Видеокарта: Nvidia Geforce 6600 и лучше

Жесткий диск: 10мб

## Требования к программному обеспечению.

**Операционная система: Windows XP и выше**

**Интерпретатор Python версии 3.0 и выше**

## Требования к интерфейсу

**Интерфейс должен быть максимально простым и понятным конечному пользователю. Главное меню не должно содержать излишнее количество пунктов. Главное окно программы должно иметь графического окно, в котором будет визуализироваться весь код написанный пользователем, главное оно должно иметь форму ввода исполняемого кода, и форму вывода, в которой будут выводится ошибки.**

# Схема взаимодействия функциональных блоков.

Схема взаимодействия функциональных блоков представлена на рисунке 5



Рисунок 5 – схема взаимодействия функциональных блоков

## Блок пользовательских настроек

Блок пользовательских настроек включается в себя функции, которые загружают, сохраняют и обрабатывают данные пользователя. А именно: его имя, уровень, который он проходит и статистика по написанному коду.

## Блок игрового процесса

Блок игрового процесса содержит в себе функции обработки нажатий кнопок, загрузки карты, исполнения на интерпретаторе кода : пользователь вводит в специальное поле свои исходный код программы, нажимает на кнопку исполнения, программа передается интерпретатору python, который исполняет ее и заносит в лог всю информацию. Также этот блок включает в себя функцию проверки на прохождение пользователем уровня: если персонаж находится на клетке выхода, то пользователю засчитывается прохождение уровня, иначе нет.

## Блок визуализации

Блок визуализации содержит в себе функции отвечающие за прорисовку на экран: функция рисования карты, анимация движения главного героя, своевременная перерисовка экрана.

# Математический аппарат

Для графического изображения общего принципа работы программы был выбран ориентированный граф, который представлен на рисунке 6.



Рисунок 6 – общая схема работы программы

# Разработка схем алгоритмов

Блок – схемы алгоритмов представлены в приложений А

# Разработка программного обеспечения

## Разработка модульной структуры

Игру было решено разбить на 5 модулей: модуль создания окна, модуль главного меню, модуль игры, модуль ядра, модуль тестовой системы.

Модуль создания окна представляет из себя алгоритмы прорисовки окна, перерасчета размера объектов, при изменении размеров окна, загрузка файла и сохранение файла конфигурации.

Модуль главного меню представляет из себя алгоритм построения главного меню, обработчики событий при взаимодействием с элементами главного меню

Модуль игры включает в себя визуализатор кода пользователя, отслеживание игровых событий.

Модуль ядра включает в себя все необходимые классовые структуры, а именно: класс фонового изображения, необходимы для визуализации фона главного меню, класс пунктов меню, класс текстовых сообщений, класс всплывающих сообщений.

Модуль тестовой системы включает в себя алгоритм исполнения исходного кода пользователя и последующего ведения лога.

Модульная структура представлена на рисунке 7.



Рисунок 7 – модульная структура игры

## Разработка пользовательского интерфейса

Следуя из требований, конечный продукт должен иметь наиболее простой и понятный пользователю интерфейс.

Главное меню будет состоять из:

* Начать игру
* Сменить профиль
* Справка
* Выход

Окно игры будет состоять из:

* Поле ввода исходного кода
* Кнопка исполнения исходного кода пользователя
* Холст на котором будет прорисовываться все действия пользователя

## Исходный код

Исходный код программы представлен в приложении Б

# Заключение

Результатом работы над курсовым проектом является 2-х мерная игра, обучающая языку программирования python. Конечная программа получилась согласно требованиям предъявленных к ней, а именно:

* Имеет низкий порог вхождения пользователей.
* В качестве обучающего языка используется реально существующий язык - python
* Для работы игры не требуется наличие интернет - соединения
* Разработан модуль, который осуществляет функцию битв логических стратегий

В дополнение к этому, вместе с основной игрой добавлен редактор карт для возможности создания собственных уровней, в том числе и для битв логических стратегий.

Главным преимуществом данной игры заключается в возможности игры в нее без наличия интернет – подключения.

Библиографический список

1. Ростовцев В.С. Оформление курсовых и дипломных проектов для студентов специальности 230101 [Текст] / В.С. Ростовцев, С.Д. Блинова. – Киров: Изд-во ВятГТУ, 2006. – 39 с.
2. Девять сервисов для обучения программированию [Электронный ресурс] / VC. RU, 2014- . − Режим доступа : https://vc.ru/p/code, свободный. − Загл. с экрана.