Государственное бюджетное профессиональное образовательное учреждение

«Арзамасский коммерческо – технический техникум»

Исследовательская работа

«Разработка обучающего электронного курса «Юный программист»

Выполнили: Мишин Артем,

Черепенков Андрей, Мальцев Артем

г. Арзамас, 2022

Содержание

|  |  |
| --- | --- |
| Введение | 4 |
| 1 Теоретические основы разработки обучающего курса «Юный программист» | 6 |
| 1.1 Понятие и классификация электронных обучающих курсов | 6 |
| 1.2 Роль электронных обучающих курсов в повышении качества обучения | 9 |
| 1.3 Описание предметной области | 11 |
| 2 Проектирование и разработка электронного обучающего курса | 13 |
| 2.1 Анализ технического задания и определение требований к ё электронному обучающему курсу «Юный программист» | 13 |
| 2.2 Построение логической модели предметной области | 16 |
| * 1. Обоснование выбора програмного обеспечения | 17 |
| 2.4 Разработка интерфейса электронного обучающего курса «Юный программист» | 22 |
| Заключение | 42 |
| Список использованных источников | 43 |
| Приложение | 44 |

Введение

Профессия программиста на сегодняшний день является одной из наиболее популярных и востребованных на рынке труда. В настоящее время появился достаточно большой спрос на отечественное программное обеспечение, способное конкурировать с зарубежными аналогами.

Профессия программист является в то же время одной из самых сложных областей для обучения. Для совершенствования знаний и практических навыков в профессии существует много образовательных курсов по обучению, которые требует от обучаемого максимум самостоятельности и склонности к самообучению и саморазвитию. Многие программисты стремятся повысить свой уровень квалификации в данной предметной области, но возникает вопрос «Какой сервис для обучения выбрать?»

При анализе различных электронных обучающих курсов, выявили, что нет современного электронного ресурса для самообразования по профессии программист, который включал бы в себя материалы для обучения, практикум с разобранными примерами, ссылками на онлайн-редакторы кода, использование возможностей GITHUB, тестовых заданий для самоконтроля в бесплатном доступе для любого желающего пользователя.

Тема данной исследовательской работы: «Разработка обучающего электронного курса «Юный программист»».

Целью данной исследовательской работы является разработать обучающий электронный курс для начинающих программистов.

Для достижения цели необходимо решить следующие задачи:

* провести анализ электронных обучающих веб-сайтов для начинающих программистов;
* выполнить проектирование веб-сайта (электронного обучающего курса);
* осуществить программную реализацию веб-сайта;
* выполнить тестирование и отладку веб-сайта;
* апробировать веб-сайт.

Объектом исследования - является веб-сайт для начинающих программистов.

Предметом исследования является использование веб-сайта, как дополнительного обучающего средства профессии программиста.

Целевая аудитория –студенты профессии программист.

Практическая значимость работы выражается в реализации обучающего веб-сайта, предназначенного для приобретения студентами практических знаний и навыков по профессиональным учебным дисциплинам специальности «Информационные системы и программирование». А также обучающий веб-сайт значительно улучшит процесс обучения.

Исследовательская работа состоит из введения, теоретической части, практической части, а также заключения, списка литературы и приложения.

1 Теоретические основы разработки обучающего курса «Юный программист»

* 1. Понятие и классификация электронного обучающего курса

Одним из приоритетных направлений процесса информатизации современного общества является информатизация образования — процесс обеспечения сферы образования методологией и практикой разработки, а также оптимального использования современных или, как их принято

Информатизация образования активизирует наращивание и использование интеллектуального потенциала человека и общества в целом, обеспечивает каждому индивиду доступ к информационному ресурсу с возможностью его визуализации и осуществления интерактивного взаимодействия с информационным источником. При этом происходит интеграция образовательных ИКТ с другими развивающими педагогическими технологиями, что обуславливает повышение качества обучения школьников, готовит их к самостоятельной профессиональной деятельности и успешной социализации в обществе.

Одним из средств внедрения современных образовательных технологий являются электронные обучающие курсы по изучению предметного содержания профессиональных дисциплин.

Электронный учебный курс (ЭУК) — это образовательное [электронное издание](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B8%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5), или информационная система комплексного назначения для реализации дидактических возможностей средств информационно-коммуникационных технологий и поддержки учебного процесса в учреждениях общего, специального, профессионального образования, а также для [самообразования](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%BC%D0%BE%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) в рамках учебных программ, в том числе нацеленных на непрерывное образование. ЭУК является частью электронного образования.

ЭУК позволяет выполнять все основные методические функции электронных изданий:

* справочно-информационные;
* контролирующие;
* функции тренажера;
* имитационные;
* моделирующие;
* демонстрационные.

С точки зрения [информационно-коммуникационных технологий](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D0%B8) ЭУК — это информационная система (программная реализация) комплексного назначения, обеспечивающая посредством единой компьютерной программы, без обращения к бумажным носителям информации, реализацию дидактических возможностей средств ИКТ во всех звеньях дидактического цикла процесса обучения:

* постановку познавательной задачи;
* предъявление содержания учебного материала;
* организацию применения первично полученных знаний (организацию деятельности по выполнению отдельных заданий, в результате которой происходит формирование научных знаний);
* обратную связь, контроль деятельности учащихся;
* организацию подготовки к дальнейшей учебной деятельности (задание ориентиров для самообразования, для чтения дополнительной литературы).

При этом ЭУК, обеспечивая непрерывность и полноту дидактического цикла процесса обучения, предоставляет теоретический материал, организует тренировочную учебную деятельность и контроль уровня знаний, информационно-поисковую деятельность, математическое и имитационное моделирование с компьютерной визуализацией и сервисные функции.

Электронный учебный курс может иметь встроенные механизмы адаптации под нужды конкретного учащегося, делающие процесс обучения более индивидуальным, а значит и более эффективным. Эти механизмы должны учитывать возможные различия между обучаемыми, определяющие разницу в восприятии учебного материала.

Основное назначение компьютерного учебного курса - формирование и закрепление новых знаний, умений и навыков в определенной предметной области и в определенном объеме в индивидуальном режиме, либо при ограниченной по объему методической помощи преподавателя (в частности, сетевых консультациях).

Современные компьютерные дидактические программы: электронные учебники, компьютерные задачники, учебные пособия, гипертекстовые информационно-справочные системы - архивы, каталоги, справочники, энциклопедии, тестирующие и моделирующие программы-тренажеры разрабатываются на основе мультимедиа-технологий, которые возникли на стыке многих отраслей знания.

Использование цветной компьютерной анимации, высококачественной графики, видеоряда, схемных, формульных, справочных презентаций позволяет представить изучаемый курс в виде последовательной или разветвляющейся цепочки динамических картинок с возможностью перехода (возврата) в информационные блоки, реализующие те или иные конструкции или процессы. Мультимедиа-системы позволяют сделать подачу дидактического материала максимально удобной и наглядной, что стимулирует интерес к обучению и позволяет устранить пробелы в знаниях. Кроме того, подобные системы могут и должны снабжаться эффективными средствами оценки и контроля процесса усвоения знаний и приобретения навыков.

1.2 Роль электронных обучающих курсов в повышении качества обучения

Повышение качества образования – приоритетное направление в деятельности любого образовательного учреждения, поскольку именно этот критерий позволит системе образования трансформироваться в новое состояние, обеспечивающее достойный уровень обучения и воспитания, адекватный потребностям развивающейся личности и рынка труда.

Требования к качеству профессиональной подготовки растут, диапазон индивидуальных различий широк. В этих условиях ведется активный поиск таких инструментов построения учебного процесса, которые были бы эффективны для обучения, безотказно обеспечивали успех в руках обычного педагога.

В связи с переходом на новые стандарты образования, особое значение приобретает личностно-ориентированный подход в обучении. Очень важно создать каждому обучающемуся комфортные условия и темп деятельности, предоставляя ему больше самостоятельности, не снижая при этом ответственности за конечный результат.

Средством повышения качества обучения являются электронные образовательные курсы, так как они позволяют:

* выбрать индивидуальный режим работы обучающихся;
* повысить качество усвоения изученного материала с помощью мультимедийных средств;
* повысить уровень самостоятельности учебной деятельности;
* организовать разные формы использования ЭУ;
* повысить самоорганизацию, дисциплинированность и ответственность обучающихся.
* выбрать на свое усмотрение интересный электронный учебный курс.
  1. Описание предметной области

Электронный образовательный курс, как правило, представляет собою самостоятельное мультимедийное средство обучения, поэтому его структура должна быть представлена на качественно новом уровне.

Электронный курс включает в себя следующие обязательные компоненты (блоки):

* средства изучения теоретических основ языка программирования;
* средства изучения практических основ языка программирования;
* справочники, редакторы кода, сервис хранения и совместной разработки;
* контроль знаний при изучении курсов;
* дополнительные средства мультимедиа.

Вышеуказанные компоненты взаимосвязаны между собой следующим образом: электронный ресурс разбит на разделы, каждый раздел содержит теоретические сведения, практические сведения (представлены в виде видеороликов, с пояснениями по выполнению работы), практические задания для самостоятельного выполнения и блок контроля (тестовые задания), форма регистрации пользователей, меню по веб-сайту, в которую включены дополнительные функциональные возможности.

Электронный ресурс должен отвечать следующим требованиям:

* четкая структуризация предметного материала (разделение теоретической и практической части);
* сложность и глубина структуризации учебного материала;
* наличие рекомендаций по изучению учебного материала;
* компактность представленного информационного материала;
* краткость и ясность в изложении основных моментов;
* графическое оформление и наличие иллюстративного материала (поясняющие схемы, рисунки и др., видео файлы);
* включение текущего контроля знаний.

Основная особенность электронного учебного пособия состоит в сочетании необходимого теоретического материала, практического и контрольного теста.

Использование электронных ресурсов как дополнительно средства обучения позволяет более глубоко изучить материал, ознакомиться более подробно с интересующими или трудными темами. Богатый и красочный иллюстративный материал позволяет наглядно продемонстрировать теоретическую информацию во всем ее многообразии и комплексности. А разобранные практические задания в виде видеоролика способствуют лучшему пониманию учебного материала.

2. Проектирование и разработка электронного обучающего курса

2.1 Анализ технического задания и определение требований к электронному обучающему курсу «Юный программист»

Средства разработки электронных ресурсов можно разделить на группы, используя комплексный критерий, включающий такие показатели, как назначение и выполняемые функции, требования к техническому обеспечению, особенности применения. В соответствии с указанным критерием возможна следующая классификация:

* традиционные алгоритмические языки;
* инструментальные средства общего назначения;
* средства мультимедиа;
* гипертекстовые и гипермедиа средства.

Традиционные алгоритмические языки.

Характерные черты электронных ресурсов, созданных средствами прямого программирования:

* разнообразие стилей реализации (цветовая палитра, интерфейс, структура электронного ресурса, способ подачи материала и т.д.);
* сложность модификации и сопровождения;
* большие затраты времени и трудоёмкость;
* отсутствие аппаратных ограничений, т.е. возможность создания электронного учебного пособия, ориентированного на имеющуюся в наличии техническую базу.

Для создания электронного ресурса были выбраны следующие инструментальные средства: язык гипертекстовой разметки HTML, CSS, язык программирования Java Script,PHP графический редактор CS Photoshop,FIGMA.

К электронному ресурсу предъявляются следующие дидактические требования:

1. Требование научности означает достаточную глубину, корректность и научную достоверность изложения содержания учебного материала, с учетом последних научных достижений.

2. Требование доступности означает необходимость определения степени теоретической сложности и глубины изучения учебного материала сообразно возрастным и индивидуальным особенностям учащихся.

3. Требование проблемности обучения обусловлено самой сущностью и характером учебно-познавательной деятельности. Когда учащийся сталкивается с учебной проблемной ситуацией, требующей разрешения, его мыслительная активность возрастает.

4. Требования наглядности обучения означает необходимость учета чувственного восприятия изучаемых объектов, их макетов или моделей и их личное наблюдение учащимся.

5. Требования самостоятельности и активизации обучаемого предполагает самостоятельных действий, учащихся при работе с электронным ресурсом, четкое понимание конечных целей и задач учебной деятельности.

6. Требования системности предполагает последовательность усвоения учащимся определенной системы знаний в изучаемой предметной области. Необходимо, чтобы знания, умения и навыки формировались в определенной системе, в строго логическом порядке и находили применение в жизни.

Кроме этого, к ЭР (электронный ресурс) предъявляются структурные требования, которые включают следующие параметры:

* средства изучения теоретических основ дисциплины (информационная составляющая);
* средства поддержки практических знаний;
* средства контроля знаний при изучении дисциплины;
* дополнительные функциональные возможности.

Содержательные требования должны отражать:

* теоретическую составляющую, которая будет реализована в виде лекционного материала.
* практическую составляющую, представлена в виде видеороликов, задания для самостоятельного выполнения;
* контрольный блок, который будет содержать материал для проверки знаний (тестирование).

Требования к учебной деятельности:

* формирование новых знаний будет обеспечиваться через лекционный материал;
* практическое закрепление материала будет обеспечиваться через выполнение практических заданий;
* проверка освоенности изученного материала – осуществляется через тестирование.

Формы организации:

* сетевое использование позволит обучающимся работать с ЭР при самостоятельном изучении теоретического материала и при выполнении практических занятий.

Также к электронному пособию предъявляются эргономические требования. Для электронного учебного пособия выбран шрифт текста –TimesNewRoman, гарнитура шрифта - 14. Данный шрифт оптимально подходит для создания благоприятной визуальной среды пособия. На странице использованы четкие изображения в высшем качестве текстур. Размер используемых объектов оптимален для восприятия.

2.2 Построение логической модели предметной области

В основном, электронные ресурсы используются для организации самостоятельной работы обучающих и должны четко определять, какие именно разделы и в какой последовательности должны быть изучены и взаимосвязаны между собой. Должны быть учтены последовательности изучаемого материала: теоретическая часть, практическая, контрольные задания, дополнительные материалы, ссылки, ресурсы.

Для четкости представления структуры пособия ее можно обозначить графически. Структура может выглядеть следующим образом (рис. 1).

Обучающий электронный курс «Юный программист»

Контрольный блок

Практическая часть

Об авторе

Теоретический материал

Практическое задание №1

Тестирование

Практическое задание №2

Тема №1

Тема №2

Практическое задание №3

Тема №3

Тема №4

…………….

Рис.1 Структурная схема электронного учебника

2.3 Обоснование выбора програмного обеспечения

Для разработки веб-сайта можно использовать различные средства и среды программирования. Каждая из таких сред программирования имеет свои преимущества и недостатки. Многие из них облегчают работу. Для разработки данной программы использовалась язык гипертекстовой разметки HTML, CSS, язык программирования Java Script, графический редактор CS Photoshop.

Для верстки сайта использован язык гипертекстовой разметки HTML5. Язык HTML был разработан британским учёным Тимом Бернерсом-Ли в 1989-1991 годах и создавался как язык обмена научной и технической документацией.

Гипертекстовые документы (HTML-страницы) интерпретируются браузером клиента в удобное для них представление, которое позволяет просматривать различного вида информацию (текст, графику). С помощью средств форматирования HTML можно составить сложную иерархическую структуру из HTML-страниц.

HTML – теговый язык разметки. Любой документ на этом языке представляет собой набор элементов. Элементы могут быть пустыми, вложенными и иметь атрибуты, определяющие какие-либо их свойства.

HTML5 – это последняя версия стандарта HTML, которая предоставляет для разработчиков больше гибкости при разработке дизайна сайтов. В HTML5 есть значительные улучшения по сравнению с более ранними версиями:

* постоянная обработка ошибок;
* улучшенная семантика для элементов;
* расширенная поддержка функций веб-приложений;
* упрощение создания мобильных сайтов;
* элементы canvas и menu;
* настраиваемые атрибуты данных.

Для оформления сайта использован внешний CSS-файл. CSS – это формальный язык описания и оформления внешнего вида документа. Основной целью разработки CSS является разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое производится с помощью формального языка CSS).

Правила CSS пишутся на [формальном языке](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) CSS. Правила могут располагаться как в самом [веб-документе](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82), внешний вид которого они описывают, так и во внешних [файлах](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB), имеющих [формат](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D1%82_%D1%84%D0%B0%D0%B9%D0%BB%D0%B0) CSS. Формат CSS — это [текстовый файл](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB), в котором содержится перечень правил CSS и комментариев к ним.

Стили CSS могут быть подключены или внедрены в описываемый ими [веб-документ](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82) четырьмя способами:

* когда описание стилей находится в отдельном файле, оно может быть подключено к документу посредством элемента <link>, включённого в элемент<head>;
* когда файл стилей размещается отдельно от родительского документа, он может быть подключён к документу инструкцией @import в элементе <style>;
* когда стили описаны внутри документа, они могут быть включены в элемент <style>, который, включается в элемент<head>;
* когда стили описаны в теле документа, они могут располагаться в атрибутах отдельного элемента.

В первых двух случаях к документу применены внешние стили, а во вторых — внутренние стили.

До появления CSS оформление веб-страниц осуществлялось исключительно средствами [HTML](https://ru.wikipedia.org/wiki/HTML), непосредственно внутри содержимого документа. Однако с появлением CSS стало возможным принципиальное разделение содержания и представления документа. За счёт этого нововведения стало возможным лёгкое применение единого стиля оформления для массы схожих документов, а также быстрое изменение этого оформления.

Преимущества:

* Несколько дизайнов страницы для разных устройств просмотра. Например, на экране дизайн будет рассчитан на большую ширину, во время печати меню не будет выводиться, а на сотовом телефоне меню будет следовать за содержимым.
* Уменьшение времени загрузки страниц сайта за счёт переноса правил представления данных в отдельный CSS-файл. В этом случае браузер загружает только структуру документа и данные, хранимые на странице, а представление этих данных загружается браузером только один раз и может быть [закэшировано](https://ru.wikipedia.org/wiki/%D0%9A%D1%8D%D1%88" \o "Кэш).
* Простота последующего изменения дизайна. Не нужно править каждую страницу, а достаточно лишь изменить CSS-файл.
* Дополнительные возможности оформления. Например, с помощью CSS-вёрстки можно сделать блок текста, который остальной текст будет обтекать (например для меню) или сделать так, чтобы меню было всегда видно при прокрутке страницы.

Внешнюю таблицу стилей нужно написать всего один раз при создании сайта для каждого из устройств, на котором планируется вывод ин­формации. К тому же внешняя таблица стилей может быть единой для целого сайта.

Недостатки:

* Различное отображение вёрстки в различных браузерах (особенно устаревших), которые по-разному интерпретируют одни и те же данные CSS.
* Часто встречающаяся необходимость на практике исправлять не только один CSS-файл, но и теги HTML, которые сложным и ненаглядным способом связаны с селекторами CSS, что иногда сводит на нет простоту применения единых файлов стилей и значительно увеличивает время редактирования и тестирования.

JavaScript (аббр. JS)  -  [мультипарадигменный](https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BF%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), поддерживает [объектно-ориентированный](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [императивный](https://ru.wikipedia.org/wiki/%D0%98%D0%BC%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [функциональный](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) стили. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам [приложений](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0). Наиболее широкое применение находит в [браузерах](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) как язык сценариев для придания [интерактивности](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C) [веб-страницам](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0).

Основные архитектурные черты: [динамическая типизация](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), [слабая типизация](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%B0%D0%B1%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), [автоматическое управление памятью](https://ru.wikipedia.org/wiki/%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0_%D0%BC%D1%83%D1%81%D0%BE%D1%80%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [прототипное программирование](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5" \o "Прототипное программирование), функции как [объекты первого класса](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_%D0%BF%D0%B5%D1%80%D0%B2%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0).

Кроме того, JavaScript имеет ряд свойств, присущих [функциональным языкам](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) — функции как [объекты первого класса](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_%D0%BF%D0%B5%D1%80%D0%B2%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0), объекты как списки, [карринг](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%80%D1%80%D0%B8%D0%BD%D0%B3" \o "Карринг), [анонимные функции](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC%D0%BD%D1%8B%D0%B5_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8), [замыкания](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BC%D1%8B%D0%BA%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) — что придаёт языку дополнительную гибкость.

Несмотря на схожий с Си синтаксис, JavaScript по сравнению с языком [Си](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) имеет коренные отличия:

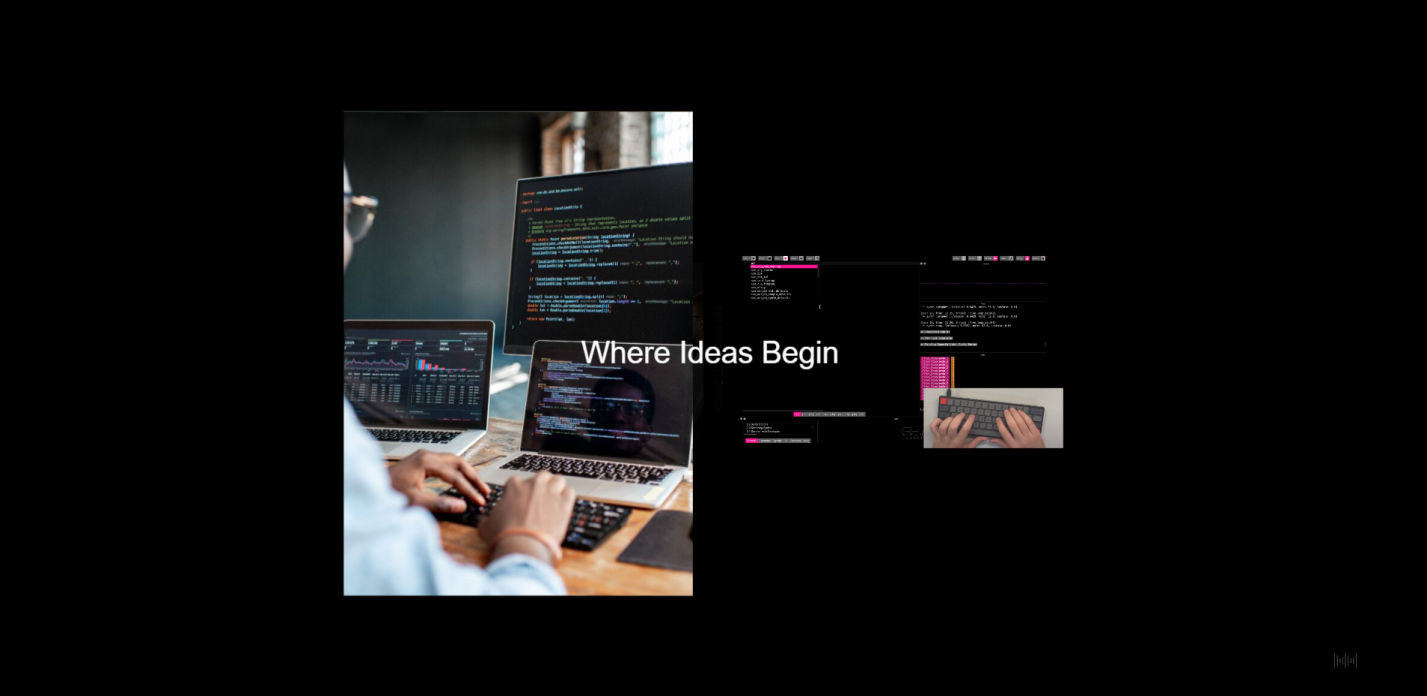
* [объекты](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) с возможностью [интроспекции](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D1%80%D0%BE%D1%81%D0%BF%D0%B5%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5));
* функции как [объекты первого класса](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_%D0%BF%D0%B5%D1%80%D0%B2%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0);
* автоматическое [приведение типов](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5_%D1%82%D0%B8%D0%BF%D0%BE%D0%B2);
* автоматическая [сборка мусора](https://ru.wikipedia.org/wiki/%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0_%D0%BC%D1%83%D1%81%D0%BE%D1%80%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5));
* [анонимные функции](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC%D0%BD%D1%8B%D0%B5_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8).

В языке отсутствуют такие:

* [стандартная библиотека](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0): в частности, отсутствует [интерфейс программирования приложений](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) по работе с файловой системой, управлению потоками ввода-вывода, базовых типов для [бинарных данных](https://ru.wikipedia.org/wiki/%D0%94%D0%B2%D0%BE%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB);
* стандартные [интерфейсы](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) к [веб-серверам](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) и [базам данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85);
* [система управления пакетами](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BF%D0%B0%D0%BA%D0%B5%D1%82%D0%B0%D0%BC%D0%B8), которая бы отслеживала зависимости и автоматически устанавливала их.
  1. Разработка интерфейса электронного обучающего курса «Юный программист»

Веб-сайт предназначен для получения теоретических знаний и практического опыта по изучению выбранного обучающего курса.

Структурный вид программы выглядит следующим образом (рис. 2).



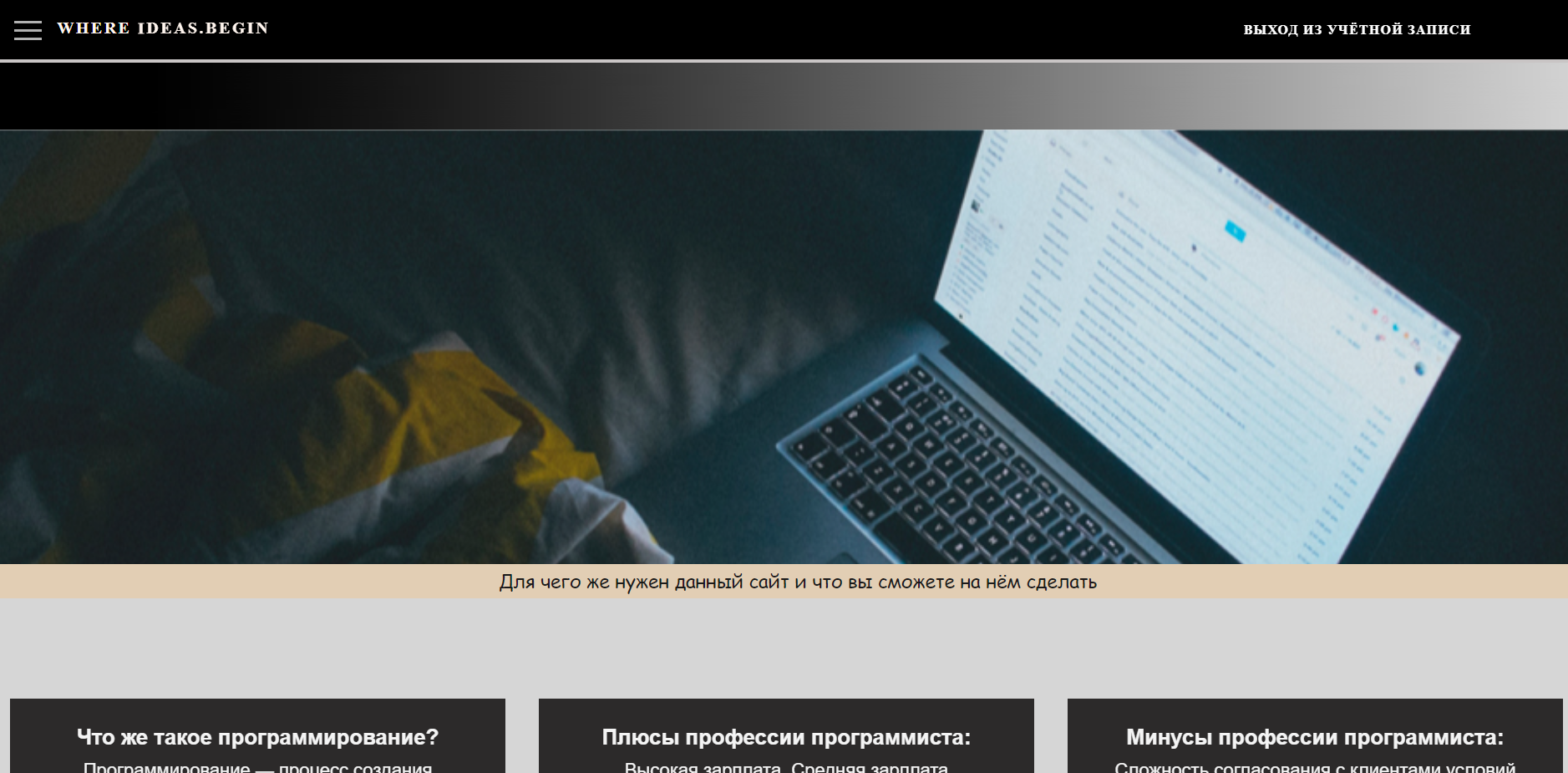


Рис. 2 Гл. страница веб-сайта

Меню веб-сайта выглядит следующим образом.

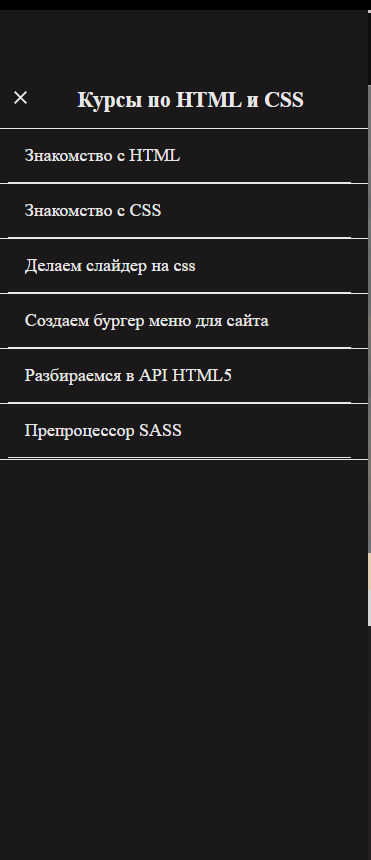


Рис. 3 Меню веб-сайта

Форма регистрации выглядит следующим образом. После регистрации пользователь получает доступ к электронному ресурсу.

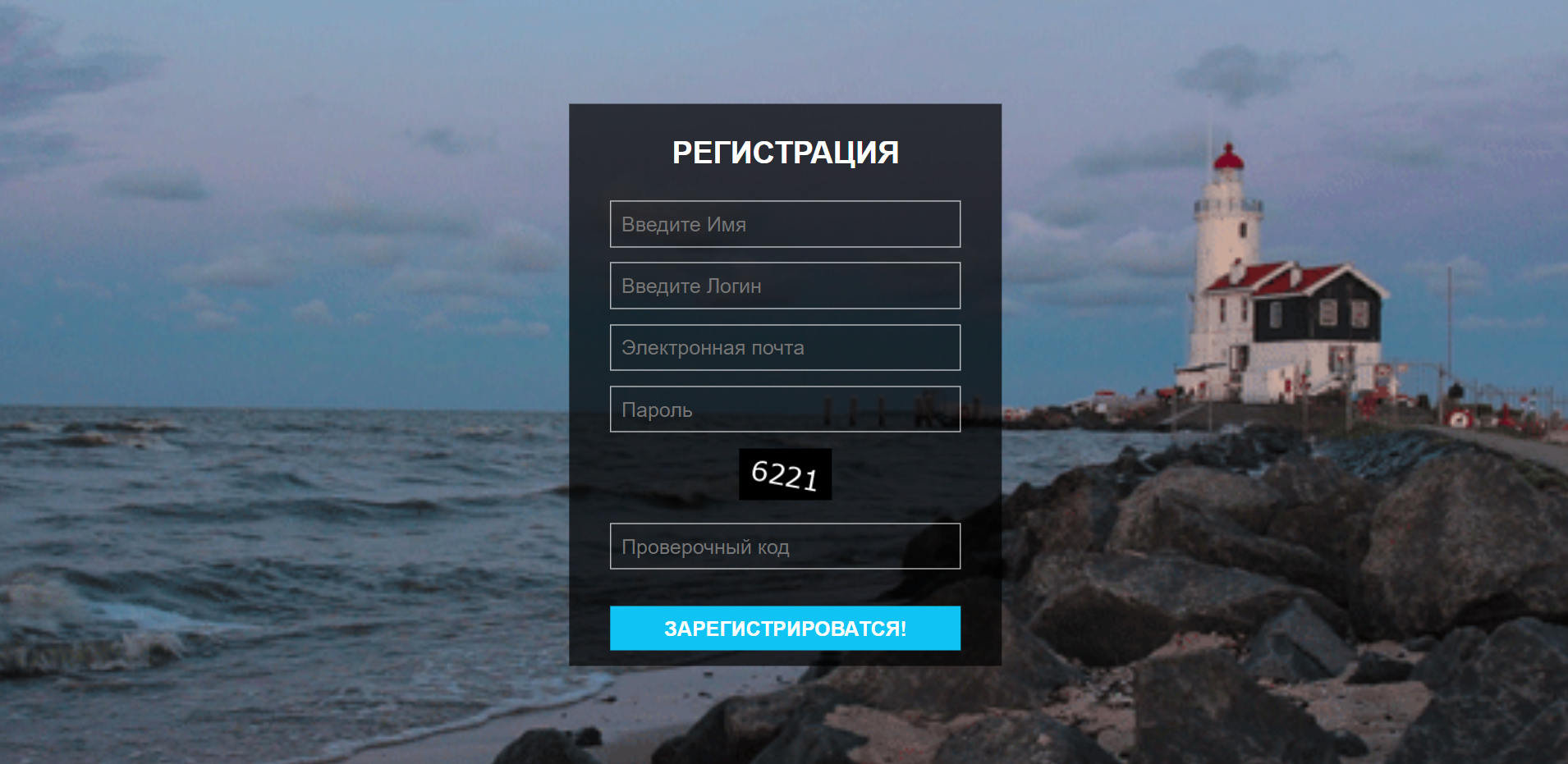


Рис. 4 Форма регистрации

После входа в систему необходимо выбрать обучающий курс на выбор. Далее появится доступ к теоретической части, практической части, контрольных тестов по темам (рис. 5,6).



Рис. 5 Теоретическая часть

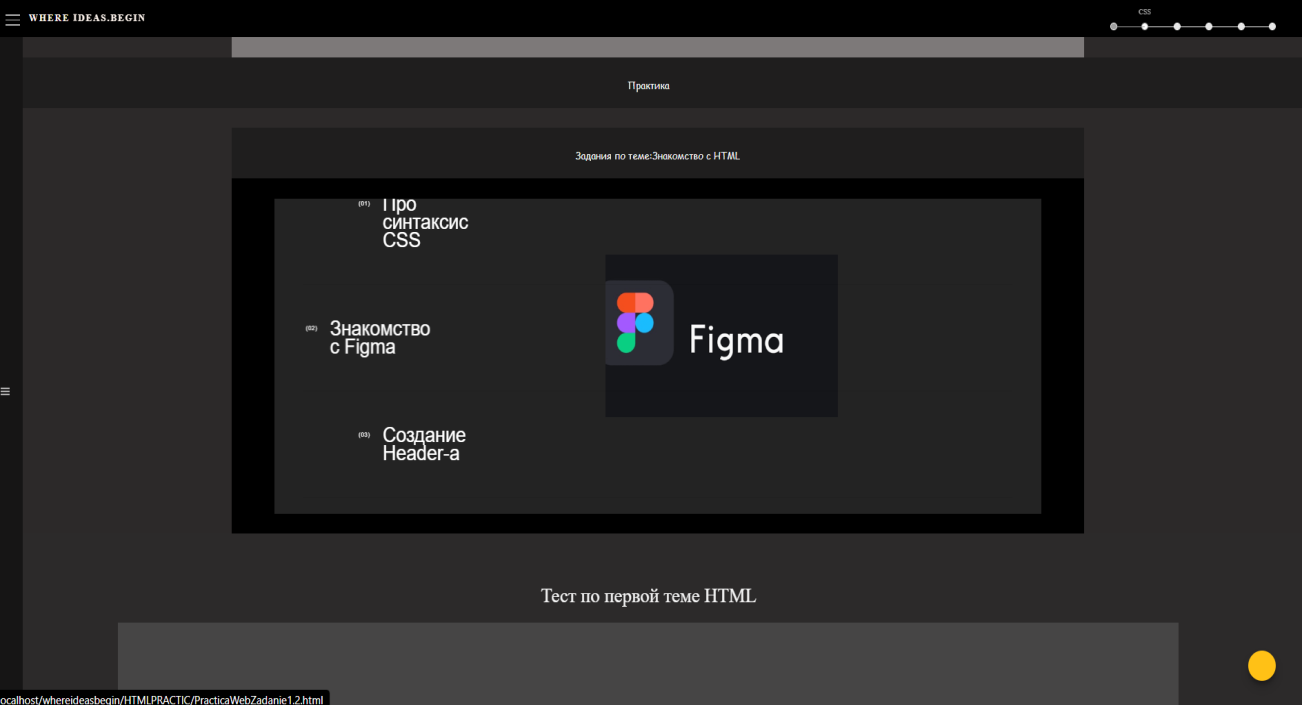


Рис. 6 Практическая часть

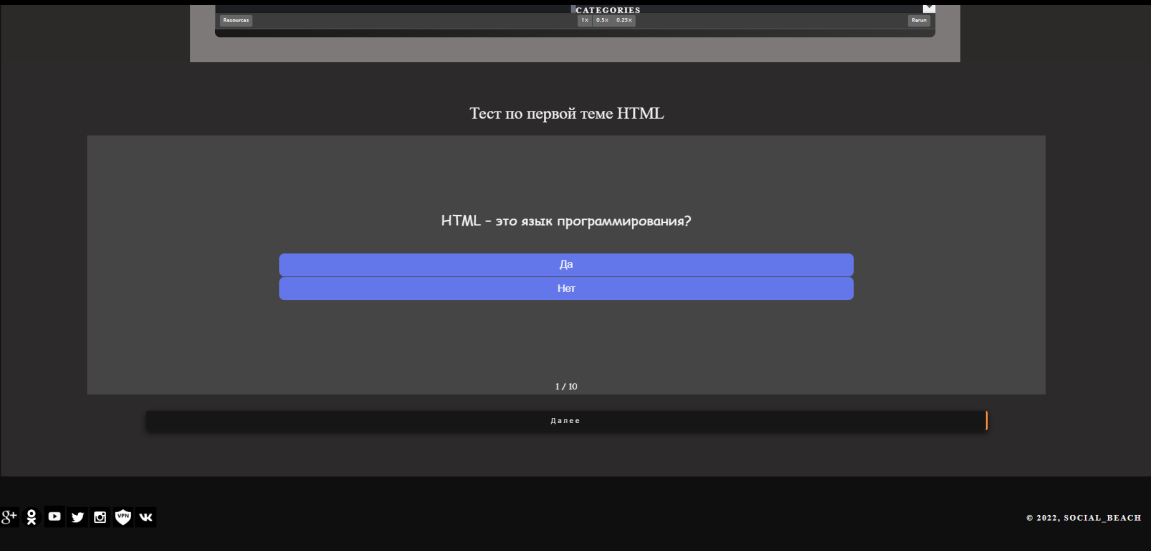


Рис. 7 Контрольные тесты

Окно «О программе» содержит краткие сведения о разработчике.

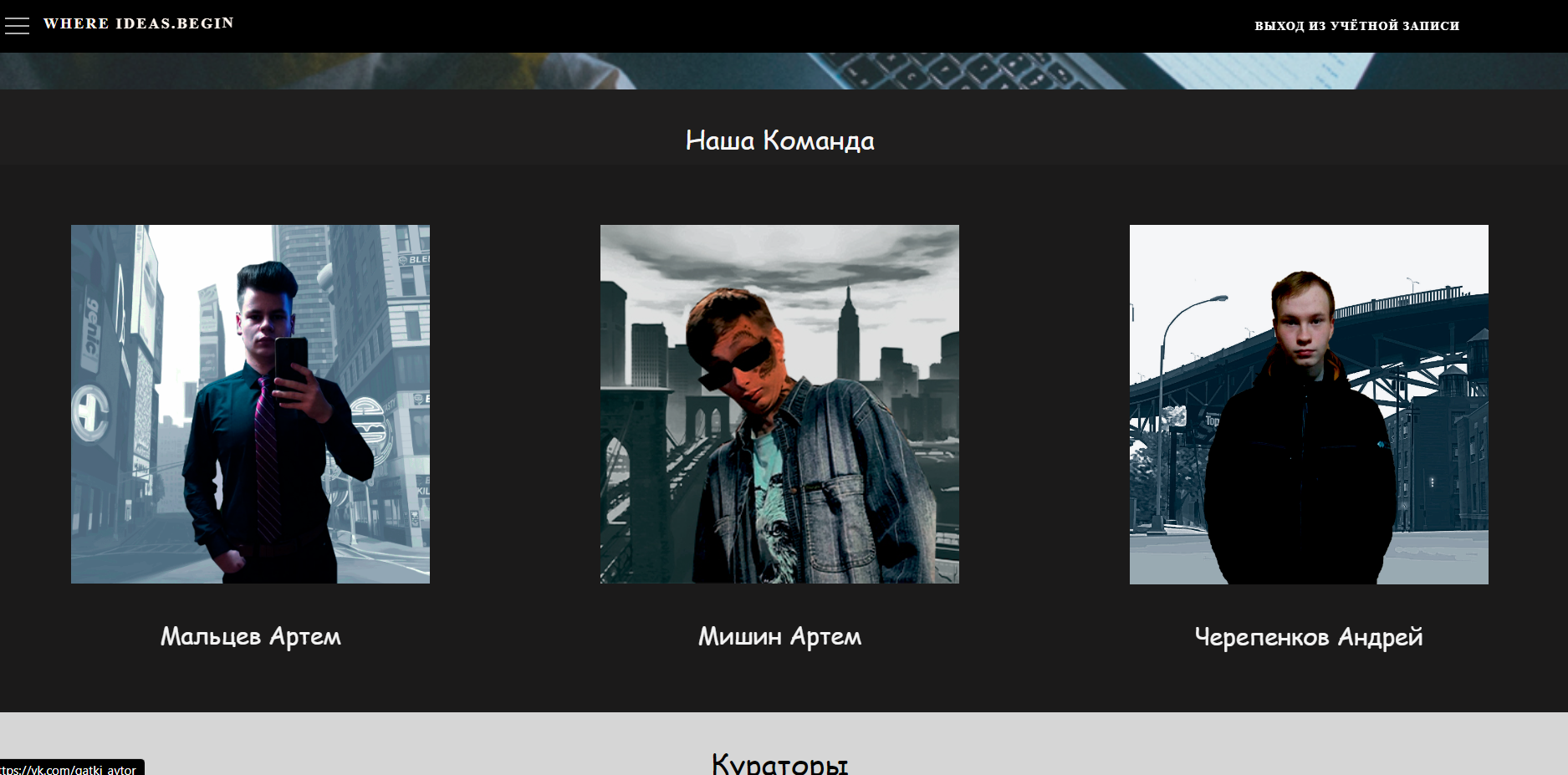


Рис. 8 Сведения о разработчике

Заключение

Целью исследовательской работы была разработка обучающего веб-сайта для самостоятельного обучения.

Были пройдены следующие этапы:

* постановка целей и задач исследовательской работы;
* исследование предметной области;
* анализ требований;
* проектирование;
* разработка веб-сайта;
* тестирование и отладка;
* документирование.

Средой разработки было: язык гипертекстовой разметки HTML, стили CSS, язык программирования JavaScript. Данные средства разработки помогли реализовать все необходимые возможности электронного ресурса.

Электронный обучающий ресурс включает в себя теоретический материал, практические задания, контроль знаний.

После обучения каждая тема отмечается другим цветом. Веб-сайт содержит информацию о разработчике данного проекта.

Программа создана для использования во внеурочное время, для самостоятельного обучения.

Цели исследовательской работы были достигнуты, а также были приобретены новые навыки в работе.

Список использованных источников

1. Аллатова И.В. Новые информационные технологии в обучении. – М.: Изд. МГПУ, 2006. – 318 с
2. Архангельский, А. Я. Программирование в Borland Delphi 7.0.: рук. разработчика – М. : ООО «Бином-Пресс», 2009 – 1152 с.
3. Денни, Д. Borland Delphi7. Руководство разработчика. [пер. с англ.] / С.М. Пишков, Б.А. Путятин Ю.С.; СПб.: «Символ-Плюс», 2000 - 684 с.; 28 см. – Библиогр.: с. 662-668.- Предм. указ.: с.669-672.
4. Захарова И.Г. Информационные технологии в образовании: учебное пособие для студентов высших учебных заведений. – 3-е изд., – М.: «Академия», 2007. – 192 с.
5. Зимина О.В. Печатные и электронные учебные издания в современном высшем образовании: Теория, методика, практика. М.: Изд-во МЭИ, 2003. – 167 с.
6. Климова, Л.М. Delphi 7. Основы программирования. Решение типовых задач: − Издательство: Кудиц-образ, 2006. – 480 с.
7. Клюшин, Д.А. Полный курс Delphi7.: уч. пособие для вузов / Т.Р. Бешелев, Г.А. Гурвич; под общ. ред. Д. А. Архангельский. − М.: «Горячая Линия – Телеком», 2001 - 382 с.; 25 см. – Библиогр.: с. 325-363.

Приложение:

