

Соискателями ученой степени доктора наук могут лица, имеющие степень кандидата наук и данным требованиям советует один человек, который и проходит обучение.

Таким образом, созданная система подготовки соискателей позволит через пять лет обеспечить укомплектованность кадрами высшей квалификации до 50%, при условии успешных защит соискателей и отсутствия утечки кадров.

УДК 378.147:004

ИГРОВОЙ ПОДХОД К ОБУЧЕНИЮ ПРОГРАММИРОВАНИЮ

Бессмертный Н.А., Денисенко А.С., Гридюшко А.В., Нестеренков С.Н.

*ЦИИР, Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Аннотация: В данной статье рассматривается применение игрового подхода к обучению программирования в высших учебных учреждениях. Для успешной реализации обучения, важным является знание всех современных инструментов. Они становятся базой современного образования, гарантирующей необходимый уровень качества, вариативности, дифференциации и индивидуализации обучения и воспитания

Ключевые слова: *программирование, scratch, app inventor, информационные технологии, образовательный процесс, образовательные сайты, project block.*

Как правило, традиционная методика обучения программированию, сложившаяся к настоящему времени, заключается прежде всего в том, что учащиеся знакомятся сначала с теоретическими основами программирования, а затем им предлагается написать программу, используя полученные теоретические знания по конкретному языку программирования. Конечно, существует мнение, что хорошему программисту не нужен компьютер вообще, а только ручка и бумага, но студенты не готовы к таким «подвигам» - им важен не столько процесс, сколько результат. Хотя, как правило, в учебниках описаны задачи вычислительного типа, которые также не представляют никакого интереса для учащихся и воспринимаются ими как повинность.

Общепринятая методика достаточно эффективна при обучении людей с достаточной математической подготовкой либо уже ориентированных на то, чтобы стать профессиональными программистами. При обучении программированию первокурсников, не имевших начальных знаний по предмету, необходимо ставить перед собой другую цель - объяснить суть программирования, которая заключается в общении с машиной на языке, понятном ей. Таким образом, студенты должны выучить новый язык общения. Оптимальная методика включает в себя два основных момента.

Первый момент состоит в том, что язык программирования не дается сначала в полном теоретическом объеме, а берется конкретная задача и объясняются лишь те элементы языка, которые необходимы для ее решения. Постепенно работа усложняется, и при написании очередной программы даются новые знания о языке программирования, которые необходимы для решения более сложной задачи. Таким образом, обучение идет от простого к сложному на конкретных задачах по программированию. При усложнении задач возникает потребность в использовании переменных, различных операторов, функций - все это объясняется студентам после обсуждения алгоритма решения задачи. Так у учеников постепенно накапливаются знания о программировании, которые тут же закрепляются на конкретной задаче. Усложняется задача - расширяются знания.

Второй момент при обучении программированию заключается в том, что ученикам предлагаются не вычислительные задачи, а именно игровые.

Наиболее эффективно проявляют себя такие специализированные программные средства, которые позволяют создавать относительно простые игры без явного использования программирования.

К таким средствам можно отнести Scratch.

Scratch был разработан маленькой командой ученых из MIT Media Lab в 2007 году. Основными компонентами программы являются объекты-спрайты. Спрайт состоит из графического представления — набора кадров-костюмов и сценария-скрипта. Для редактирования костюмов спрайтов в scratch встроен графический редактор. Действие scratch-программы происходит на сцене размером 480×360 пикселей с центром координат в середине сцены.

Для программирования сценариев в scratch используется drag-and-drop-подход: блоки из палитры блоков перетаскиваются в область скриптов. Поддерживается работа с числами, строками, звуками, логическими значениями, устройствами ввода-вывода и связанными списками.

В 2011 году в MIT был представлен программный продукт App Inventor.

App Inventor — среда визуальной разработки android-приложений, требующая от пользователя минимальных знаний программирования. Первоначально разработана в Google Labs, после закрытия этой лаборатории была передана Массачусетскому технологическому институту.

Компилятор, переводящий визуальный блочный язык App Inventor в байт-код Android, основан на фреймворке GNU для реализации динамических языков Kawa, реализующего Scheme для java платформы и Android.

Для программирования в App Inventor используется графический интерфейс, визуальный язык программирования, очень похожий на язык Scratch и StarLogo TNG. В начале марта 2011 года Массачусетский институт запустил публичную бета-версию проекта, доступную на сайте appinventor.mit.edu.

Компания Google в 2016 году, после закрытия проекта App Inventor, запустила программу Project Block, которая является по своей сути аппаратной платформой с открытым исходным кодом, которая позволяет ученикам создавать небольшие блоки, похожие на конструктор, создавая в итоге законченный алгоритм.

Для создания программ пользователь должен перемещать графические блоки, не прибегая к набору текстов, за исключением ввода значений констант. Визуальное программирование на Project Block освобождает пользователя от контроля за правильностью синтаксиса программы, что является большим подспорьем на стадии начального обучения пользователя программированию.

В результате игрового подхода к обучению ученики быстрее осваивают практическое программирование, получая теоретические знания в ходе решения поставленной задачи, а также постоянно поддерживается мотивация учиться новым подходам к решению и самостоятельной постановке все более сложных задач.

Литература

1. Scratch – Imagine, Program, Share // MIT [Электронный ресурс]. – Режим доступа: <https://scratch.mit.edu/about>

2. Project Bloks: Making code physical for kids // Googleblog [Электронный ресурс]. – Режим доступа: <https://research.googleblog.com/2016/06/project-bloks-making-code-physical-for.html>

3. Нестеренков, С.Н. Модель построения расписания на основе прецедентов / С.Н. Нестеренков – Минск: Информатизация образования. - 2015. - N 1. - С. 61-73.

4. Игровые методы при обучении программированию. [Электронный ресурс]. – Режим доступа: <http://festival.1september.ru/articles/601183/>.

5. Нестеренков, С. Н. Интегрированная информационная система как средство автоматизации управления образовательным процессом в учреждениях высшего образования / С. Н. Нестеренков, Т.А. Рак, О.О. Шатилова // Информационные технологии и системы 2017 (ИТС 2017) : материалы междунар. науч. конф., Минск, 25 окт. 2017 г. / Белорус. гос. ун-т информатики и радиоэлектроники ; редкол.: Л. Ю. Шилин [и др.]. - Минск, 2017. - С. 212.

6. Житников, С. В. Игровой подход в обучении программированию детей и подростков / С. В. Житников // Информационные системы и технологии: 53-я научная конференция аспирантов, магистрантов и студентов. (Минск, 6 мая 2017 г.). – Минск: БГУИР, 2017. – С. 49-50.

УДК 378.147:004

ДУХОВНЫЙ ОПЫТ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА НА БАЗЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Вильчук Ю.В.

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Республика Беларусь

Существующие информационные технологии не совершенны, существует ряд проблем касательно обучающихся и педагогов по использованию соответствующих информационных технологий. В процессе взаимодействия человека с компьютерными информационными технологиями наблюдается также ряд проблем. Эти проблемы вызваны традиционно установленными психологическими воззрениями и инертностью мышления. Актуальность данного исследования обусловлена созданием духовной синергетической информационно-технологической системы обучения, способной решить многие проблемы эффективности образовательного процесса на любой стадии, в любой сфере деятельности.

Чтобы повысить эффективность образовательного процесса на базе духовных информационных технологий, нужно исследовать систему причинно-следственных взаимосвязей педагога и обучающихся на разных уровнях восприятия по отношению к изучаемым дисциплинам. Чтобы это сделать, нужно обладать гибким сферическим мышлением, которое представляет собой динамический собирательный образ логически последовательных изменяющихся в пространстве нейронных связей с целью объёмного (вовлечения) охвата максимального количества обучающихся для совместного сознательного погружения в образовательный процесс. И здесь главную роль играют не информационные технологии, а мастерство педагога, умеющего виртуозно использовать каждый сантиметр обучения и к месту применять нужные знания.

По сути, информационные технологии являются лишь вспомогательным средством, которое необходимо использовать в разумных пределах. Суть любого образовательного процесса: научить человека правильно мыслить и воспринимать без искажений поступающую информацию.

Каким образом учить? Какие технологии использовать и в каком объёме представлять наглядно материал, определяет сам преподаватель в зависимости от поставленных целей и методической программы обучения утверждённой Министерством Образования.

Информационные технологии в образовательном процессе дают нам возможность комплексно воспринимать информацию, поступающую от преподавателя. Однако не всю информацию нужно визуализировать на дисплее проектора, чтобы не перегружать зрительные каналы восприятия иначе этим можно только ухудшить образовательный процесс. Довольно часто на практике приходилось сталкиваться с подобными ситуациями.