PROGRAMMIEREN FÜR KINDER – GEEIGNETE INSTRUMENTE FÜR DIE ENTWICKLUNG DES INFORMATIKDENKENS

BUČKOVÁ Hana, CZ

Resumé

Die Welt, in der wir jetzt leben, ändert sich sehr dynamisch, es werden immer neue und erhöhte Ansprüche an die Kenntnisse und Fertigkeiten im Bereich der Informations – und Kommunikations technologie. Der gegenwärtige Informatiksunterricht erfährt große Veränderungen, die sich im Unterricht in den Grundschulen widerspiegeln werden. Das Nationalinstitut für Ausbildung bereitet die neuen Ausbildungsprogramme vor, die auf aktuelle technologische Entwicklung reagieren. Zurzeit wird die Implementierung der Veränderungen vorbereitet, wobei die Schüler im größeren Maß ihr Informatikdenken entwickeln werden. Zu diesem Zweck ist eine der passenden Programmierungsumgebung Scratch – der kann man im Unterricht in den Grundschulen benutzen. Es gibt aber auch eine Reihe von den anderen Möglichkeiten - zum Beispiel ScratchJr – der die Kodierung den Kindern im Alter von 5 - 7 Jahren ermöglicht. Weiter Snap! – ermöglicht die Bildung der komplizierten Strukturen.

Der Artikel möchte die visuelle Programmiersprache Scratch vorstellen und seine weitere Möglichkeiten - zum Beispiel ScratchJr, Snap! und Pynkee. Ähnliche Computersplattformen, die Programmierung mit Hilfe des Blocks ausnützt, ist die Programmierungsumgebung Tynker. Wir werden uns auch mit den Programmierungsumgebungen beschäftigen, die sich auf den Unterricht der wirklichen Programmierung mit Hilfe der Programmiersprache wie zum Beispiel Python und JavaScript konzentrieren. Zu den weiteren Programmierungsumgebungen, die für die Kinder passend sind, gehören auch Code Avengers, Code Combat, LightBot und Codemoji. Es gibt also viele Möglichkeiten.

Schlüsselwörter: Scratch, Informatikdenken, Programmierung, Programmiersprache.

PROGRAMMING FOR KIDS – SUITABLE TOOLS FOR THE DEVELOPMENT OF COMPUTATIONAL THINKING

Abstract

The world, we are living now, is changing dynamically. There are still demands on knowledge and skills in the field of informatics and communication technologies. These changes are reflected in the teaching at the elementary schools. The National Educational Institute prepares new educational programmes that respond to the topical technological development. Currently the implementation of the changes are being prepared, where the pupils will develop their informatics thinking in a bigger way. For this use Scratch programming environment is suitable. This environment can be used for teaching at elementary schools. There is a number of options. One of them is ScratchJr which emerged with the intention to enable coding for children aged 5 to 7. Another option for using is Snap! which enable creating more complicated programming structures.

The topic of the article is the introduction of visual programming language Scratch and its options i.e. ScratchJr, Snap! and Pynkee. A similar computer platform which uses programming by means of blocks is the programming environment Tynker. We will also deal with programming environments which focus on teaching of real programming using programming languages e.g. Python and JavaScript. Among other programming environment convenient for children belong Code Avengers, Code Combat, LightBot and Codemoji. There are many other possibilities.

Key words: Scratch, informatics thinking, programming, programming languages.

Einleitung

In der gegenwärtigen Gesellschaft ist es sehr nötig, die Informatiktechnik, Informatikdenken und Computerfähigkeit zu beherrschen. Deshalb sollten die Schüler diese Fähigkeiten schon in der Grundschule entwickeln und diese Kenntnisse werden Devise für ihr zukünftiges Berufs – und Privatleben.

Nach der Strategie der digitalen Ausbildung bis das Jahr 2020 wird das Informatikdenken als „Denkensäufigkeit wie Informatiker bei der Problemlösung“ definiert. Entwicklung dieses Denkens fehlt in den Schulen völlig, obwohl es sehr lohnenswert wäre. Die Schüler in den Schulen lernen zwar Denken aber einige Gesichtspunkte zum Beispiel Informatikdenken unterlassen werden. Informatik verlängert ausdrücklich von den Schülern die Fähigkeit der Lösungsuche, in den anderen Fächern werden sich die fertigen Vorgehensweise der Lösung aneignen. Informatikdenken formuliert das Problem der Weise nach so, daß es seine Lösung ermöglicht, logisch die Daten zu forschen. Diese Daten werden mit Hilfe der Abstraktion präsentieren. Wie zum Beispiel die Modellen und Simulationen, automatisiert die Lösung mit Hilfe des algorithmischen Denkens (wie Schrittfolge) und viele andere Umgebungen. (Strategie digitálního vzdělávání do roku 2020, 2014).

Eine von der Möglichkeit der Entwicklung des Informatikdenkens stellt der Unterricht mit Hilfe der Programmiersprache Scratch vor. Es handelt sich um Programmierumgebung, die frei auf die Internetadresse scratch.mit.edu zur Verfügung ist. Man kann sie kostenlos ausnützen ohne vorherige Installation. Scratch ist für Kinder von 8 Jahren geeignet. Sie müssen keine vorherige Programmierungskenntnisse haben. Scratch ermöglicht das Programmieren mit Hilfe von Blöcken, die sich wie Puzzle stellen und für die Kinder, die Interesse an Programmieren haben, ist es eine ideale Umgebung. Scratch ermöglicht den Schülern ihr algorithmisches Denken, Kreativität, mathematische Fähigkeiten und die Vorstellung, wie das Computer funktioniert zu entwickeln. Dank dem Computer können die Kinder mit den Multimedien leicht arbeiten und viele Projekte schaffen. Wie zum Beispiel Spiele, Animationsgeschichte, Lernprogramme oder Videen. In der Sprache Scratch gibt es anstatt den Textbefehlen, Klammern und Strichpunkte der klassischen Programmiersprache die bunten Teile zur Verfügung. Diese Teile bilden nach der richtigen Schaltung ein funktionierendes Programm. Der Schüler kann sich so mehr auf den Verständnis der Programmierprinzipien und auf die Problemlösung konzentrieren.

Weitere Programmierumgebungen die wir im Artikel beschreiben, fungieren auf Prinzip der Nutzung der Programmiersprachen JavaScript, Python und HTML. Außer der Umgebung Tynker, der auf den Prinzip der Baublocksstelle ähnlich wie Scratch funktioniert. Tynker kann man nicht nur in den Schulen ausnützen aber auch zu Hause online oder auf Handyspattformen. Es ist möglich zu den anderen Spielern einzuschließen und so das Lernen lustiger machen. Code Avengers hilft den Kinder zum Lernen dank dem kurzzeitigen Camp, wo die erfahrenen Lektoren während des Lernens die Fehler der Kinder zur Verbesserungen ausnützen. CodeCombat ist revolutionär wie er die Kinder lernt – das Kind bildet seine eigene Figur und programmiert ihr Verhalten, ihr Ziel. LightBot können wir mit Betriebssystem Android und iOS auf Handys starten, die Kinder können im Team zusammenarbeiten. Codemoji lockt die Kinder mit lustigen „Smiley“ oder emoji Icons.

1 Programmiersprache Scratch

Scratch ist erziehungsorientierte kostenlose, visuelle Programmiersprache für Kinder und Jugendliche im Alter von 8 – 16 Jahren. Wurde von der Gruppe Lifelong Kindergarten am MIT entwickelt und wird von den Sprachen Squeak und Smaltalk abgeleitet. Scratch ist so vorgeschlagen, daß es für Kinder einfach, lustig, edukativ ist. Es handelt sich um eine visuelle Programmiersprache, wobei man keine Befehle und Syntax lernen braucht. In dieser Programmiersprache programmiert man mit Hilfe von Überziehen und Zusammenfalten der Anweisungen. Dieser Typ der Programmierung wird als „drag - and - drop Programmierung“ bezeichnet (Scratch, 2018). Anstatt dem Schreiben von den langen Ketten die für die jüngeren Kinder langweilig und ermüdend ist, benutzt einfache Umgebung, in der man die Bausteine mit dem Mausherüberziehen zusammen legen kann. Die Kinder entwickeln ihr Programmatorsdenken und beherrschen die Grundkonzeption, die sich in allen Programmiersprachen benutzen (Úvod do jazyka Scratch, 2013). Weiter hilft Scratch den Jugendlichen kreativ denken, systematisch überlegen, zusammenarbeiten und das sind die Fähigkeiten für das Leben im 21. Jahrhundert.

Programmiersprache Scratch hat im Vergleich zu den anderen Programmiersprachen viele Vorteile. Der eine ist die Zugänglichkeit der Sprache in Tschechisch, in der vollen Version und ganz kostenlos als online und offline Variante. Zu Scratch gehört auch die Teilung von den geschafften Programmen mit den anderen Usern. Jedes Programm kann man im Internet geben und teilen, was die Kinder zur Bildung der komplizierten Appen motiviert kann. Zu den anderen Vorteilen gehört hardware Anspruchslosigkeit und die Arbeit mit Scratch ist sehr einfach und lustig. Scratch kann man auch ohne wissenschaftliche Kenntnisse der Programmierung begreifen (Programování ve Scratchi, 2018). Scratch hat kleine Möglichkeiten von Anpassen des Aussehens von der Userumgebung. Das Fenster von Scratch ist in 6 Gebiete geteilt.

Der Autor Mitchel Resnick im Artikel Scratch: programming for all sagt, daß Hauptziel des Projekts nicht ist, die Schüler auf Karriere der Programmators vorbereiten aber die neue Generation der Schüler kreativ arbeiten lernen. Die Schüler sollen ihre Ideen mit Hilfe von Programmieren lernen. Das weitere Ziel von Scratch ist für alle Kinder offen machen.

Heute beherrschen die jungen Leute die digitalen Technologien auf Userniveau – zum Beispiel Textnachrichtenschreiben, online Spiele, Informationssuchen auf Web. Sie sind fähig, mit den digitalen Technologien zu kommunizieren aber nur wenige von ihnen können die eigenen Spiele und Animationen bilden. Programmsfähigkeit vorbereitet die Möglichkeit der Äußerung mit Hilfe vom Computer. Das Programmieren unterstützt das Informatikdenken, das uns bei den wichtigen Problemlösungen hilft (Resnick, M., Maloney, J., Eastmond, E., eds., 2009).

Bild 1 – Umgebung von Programm Scratch (Scratch)

2 ScratchJr, Snap! Und Pyonkee

Programmierungsumgebung Scratch kann man nicht nur für die Kinder der Grundschulen aber auch des Vorschulalters in der Variante, die ScratchJR nennt benutzen. Bei den Kindern dieser Alterskategorie unterstützt die Entwicklung des logischen Denkens. Andere Umgebung, mit der wir uns beschäftigen werden, ist Snap! – der für die Kinder in den Grundschulen geeignet ist. Letzte Variante von der wir nur kurz schreiben, wird Pyonkee.

2.1 ScratchJr

ScratchJr entstand dank der Zusammenarbeit der wissenschaftlichen Gruppe DevTech research group an der Universität Tufts. Die wissenschaftliche Gruppe DevTech konzentriert sich auf die Bildung der Technologie und pädagogischen Einstellungen, die in Erwähnung die Bedürfnisse der kleinen Kinder und die Entwicklung der digitalen Fertigkeit und Informatikdenken ziehen (ScratchJr: Coding for Young Kids, 2018).

Programmiersprache ScratchJr ermöglicht den kleinen Kindern im Alter von 5 - 7 Jahren eigene interaktive Geschichte und Spiele zu bilden. Die Kinder überziehen die grafischen Programmierblöcke, damit sich die Figuren bewegen, springen, tanzen und singen. Die Kinder können die Figuren im Farbeditor herrichten und ihre eigene Stimme, Geräusche und eigene Fotos zugeben. Bei der Nutzung der Programmierblöcke ihre Figuren erleben (Aplikace: ScratchJr, 2016).

ScratchJr wurde mit Scratch - Programmierungssprache inspiriert, die Millionen der Jugendlichen auf der ganzen Welt ausnützen. Bei der Programmbildung wurde die Schnittstelle und Programmiersprache so überarbeitet, damit es für die kleineren Kinder passend ist. Es wurden weiter solche Funktionen vorgeschlagen, die den kognitiven, persönlichen, gesellschaftlichen Bedürfnissen der kleinen Kinder entsprechen (About ScratchJr, 2017).

Kodierung ist ein neuer Typ der Bildung. Er hilft, genauso wie das Schreiben, das Denken zu organisieren und eigene Ideen auszudrücken, dasselbe gilt auch für die Kodierung. Wie die kleinen Kinder in ScratchJr kodieren, so lernen sie mit dem Computer bilden und ausdrücken, nicht nur mit

ihm kommunizieren. In diesem Prozess entwickeln die Kinder ihre Fähigkeiten im Bereich des Vorschlags und der Problemlösung, die der Grund für die nächsten Programmierungsfortschritten sind. Sie benutzen Mathematik und Sprache im sinnvollen und motivierten Kontext und unterstützen die Entwicklung der mathematischen und informatischen Fertigkeit. Mit dem Programm ScratchJr lernen die Kinder nicht nur kodieren aber sie kodieren auch sogar (ScratchJr: Coding for Young Kids, 2018).

Bild 2 – Umgebung von Programm ScratchJr (ScratchJr: Coding for Young Kids)

2.2 Snap!

Der Fortsetzer des Projekts Scratch ist der Instrument Snap!, der kann man im Bedarfsfall für den Unterricht der Grundkonzepten ausnützen, die Grundkonzepten werden bei dem Programmieren benutzen. Snap! entstand durch die Syntese Scratch und das Projekt BYOB, der in Webumgebung übertragen wird. Es ist nicht nötig, die Installation der relativ großen Umgebung Scratch. Snap! läuft direkt im Webdursucher und deshalb ist es kein Problem, Snap! gleich im Unterricht zu benutzen. Man kann auch verschiedene intelligente elektronische Einrichtungen benutzen wie zum Beispiel Smartphones, Tablet usw. Bei der Verwendung dieses Projekts im Unterricht verschwinden die Probleme mit der Instalation und dem Projekthalten. Dabei bleiben alle wichtige Eigenschaften von Scratch erhalten und sogar kommt zu manchen Verbreitungen (Snap! – důstojný nástupce programovacího jazyka Scratch, 2015).

Eine der wichtigen Eigenschaften der modernen Version von Snap! ist die Möglichkeit zu den einzelnen Blöcken die Kommentare zu geben. Es handelt sich um eine sehr wichtige Funktion von Snap! Die Kommentare können den Studenten bei der Arbeit mit den demonstrativen Beispielen helfen, erklären die Bedeutung der einzelnen variablen Blöcken. Die nächste, wichtige Eigenschaft Snap! ist die Möglichkeit der Bildung von den neuen Userblöcken, die man benennen kann. So einmal geschaffter Block gehört zum Programm und man teilt ihm mit den anderen Projekten oder mit den Usern. Die Schaffung des neuen Blocks ist intuitiv und einfach. Im Projekt Snap! erscheinen neue Blöcke, die für die Arbeit mit den Listen geeignet sind. Im Projekt Snap! kommt zur Erweiterung der leitenden Strukturen. Es gibt hier die verbesserten Formen des Programmsschlaufe.

Bild 3 – Umgebung von Programm Snap! (Snap! – důstojný nástupce programovacího jazyka Scratch)

2.3 Pyonkee

Pyonkee ist eine App, die auf die Plattform iOS basiert und ist eine weitere Möglichkeit der Nutzung von Umgebung Scratch. Diese App wurde im Jahre 2014 veröffentlicht und von der Firma SoftUmeYa entwickelt. Das Ziel ist, eine bildliche Programmiersprache bereitzustellen. Das Maskottchen der App ist - Affe-Hase - benannt „Pyonkee“ (Scratch wiki, 2018).

Bild 4 – Umgebung von Programm Pyonkee (Scratch wiki)

3 Die nächsten passenden Instrumente für die Entwicklung des Informatikdenkens

Die nächsten Programmierumgebungen, mit den wir uns im Artikel beschäftigen werden, sind Tynker, Code Avengers, CodeCombat, LightBot, Codemoji und sie sind in online Form im Internet zur Verfügung. Diese Programmierumgebungen sind in Englisch und so können die Kinder nicht nur das algorithmische und logische Denken entwickeln aber sogar auch ihre Sprachkenntnisse verbessern.

3.1 Tynker: Coding for kids

Tynker ist ein komplettes Lehrsystem, das jedem von den Anfängern bis zu den fortgeschrittenen Informatoren die intuitive Kodierung ermöglicht. Dank Tynker können die Kinder mit den Sichtblöcken experimentieren vor dem fortgeschrittenen Programmieren (Tynker, 2018). Tynker ist eine Programmierumgebung, die die Sichtblöcke ausnützt und ermöglicht den Kindern ihre eigene Projekte zu bilden, die Figuren zu animieren und Programmierungsrätsel zu schaffen. Diese Programmierumgebung lernt die Kinder auf spielerische Weise die Programmierung und die Code verstehen. Apple in der Zusammenarbeit mit der Gesellschaft Tynker entwickelt eine eigene STEM Bildungsplattformen.

Space Cadet ist ein Anfängerkurs, der für die Kinder vom Kindergarten bis der zweiten Klasse der Grundschule geeignet ist. Er befindet sich in der Weltraumumgebung, wo die Kinder Rätsel lösen und die kompletten Programmierprobleme. Dragon Spells ist auch ein Anfängerkurs für die Kinder des dritten bis fünften Jahrgangs, wo sie Gründe der Programmierung lernen. In diesen Kursen lernen die Schüler kreativ die Rätsel lösen, die Projekte erstellen und die Figuren animieren. (Apple in der Zusammenarbeit mit der Firma Tynker möchten die Programmierung in Swift auch die Kinder im Kindergarten lernen, 2017).

Bild 5 – Umgebung von Programm Tynker (Tynker)

3.2 Code Avengers

Code Avengers ist Programmierumgebung, die für Kinder von 5-16 Jahren geeignet ist. Es wird in drei Alterskategorien geteilt – Grundstufe (5-11 Jahren), mittelfortgeschrittene Stufe (10-15 Jahren), fortgeschrittene Stufe (13-16 Jahren). Die Kinder lernen in den Online-Kursen Kodierung zum Beispiel Webseiten, Apps, Spiele und andere. So werden die Computer und mathematische Fähigkeiten und Problemlösung entwickeln. Man organisiert auch 2-5 täglichen Camp, wo der Unterricht mit dem Lehrer gibt's. Die Anfängerstudenten können die Konzeption der Kodierung auswählen, die sie als erste lernen möchten. Sie wählen aus den Programmierumgebungen Python, HTML und Javascript aus. Code Avengers konzentriert sich auf die Experimenten und Fehler, was sehr wichtig für die Anfängersprogrammierer ist. Die lassen sich oft entmutigen, wenn sie gerade Fehler machen. Wenn der Programmierer Erfolg erreicht, ist er fähig die eigene Spiele zu erstellen (Code Avengers, 2016).

Bild 6 – Umgebung von Programm Code Avengers (Code Avengers)

3.3 CodeCombat

CodeCombat ist ein Spiel des Adventurstyps, das 5 000 000 Spieler aus 200 Ländern der Welt spielen und konzentriert sich auf den Unterricht der wirklichen Programmierung mit Hilfe von den Programmiersprachen Python und JavaScript, einschließlich aller Formalregel von Syntax. Der Spieler wählt sich eine Figur aus, mit der er spielen möchte und geht in Geschichte weiter Niveau für Niveau. In jedem Niveau bekommt der Spieler ein einfaches Ziel zur Erfüllung, zum Beispiel - zum Ausgang kommen, Edelsteine abnehmen, das Dorf von der Attacke der Räuber schützen usw. Im Vergleich zu den anderen Spielen programmiert man die Figur anstatt der direkten Beherrschung. Es wird hier kein Spiel programmiert, sondern die künstliche Intelligenz der Figur. Der Benutzer hat die begrenzten Methoden zur Erfüllung seiner Aufgabe, dank seiner Ausrüstung, die er zur Verfügung im Spiel hat. Die Funktionsfähigkeit der Code kann man in jederzeit beglaubigen und beobachten, wie sich die programmierte Figur verhält. CodeCombat lernt zwar nur Zusammenfassen von den vorbereiteten, einfachen Befehlen, er lernt keine Vorschlagsmuster aber das eigene Spiel ist ziemlich umfangreich und beginnt mit den einfachen Befehlen wie zum Beispiel, gehe oben und kommt zum Zaubersenden. In der Ausgangseinstellung wird in Python programmiert, aber man kann die Programmiersprache ändern, in der Gegenwart ist JavaScript erreichbar (CodeCombat, 2018).

Bild 7 - Umgebung von Programm CodeCombat (CodeCombat)

3.4 LightBot

LightBot ist ein didaktisches Spiel, das zum Unterricht der Grundlagen vom Programmieren mit Hilfe der Blöcke geeignet ist. Es ist für alle Alterskategorien passend. Dieses Spiel entwickelt bei den Kindern algorithmisches Denken, obwohl es mit der Programmiersprache nicht gerade arbeitet. LightBot ist in den 28 Sprachen erreichbar inklusive Tschechisch. Zur Verfügung ist auch als eine App für kluge Handys mit System Android und iOS (Yaroslavski, D., 2014).

Die Aufgabe dieses Spiels ist, den kleinen Roboter auf der Landkarte durchfahren und dabei alle Lichtfelder aus der blauen Farbe in die gelbe anschalten. Die Spieler können zwischen zwei Roboter auswählen, für die Mädchen gibt es Roboter in Violettfarbe und für die Jungen blaue. LightBot hat drei Schwierigkeitsniveaus - vom Grundniveau bis das dritte schwierigste Niveau.

Die einzelnen Level nachfolgen und deshalb sollte man sie nicht überspringen. Nach jedem erfolgreich absolvierten Level entspermt das andere.(LightBot, 2018).

Bild 8 – Umgebung von Programm LightBot (LightBot)

3.5 Codemoji

Codemoji ermöglicht den Kindern der 1. - 8. Klasse die Grundlagen vom Programmieren und Kodierung der Webseiten mit Hilfe der Sprachen HTML, JavaScript usw. Programmieren verläuft sehr einfach und unterhaltend und ermöglicht den Schülern ihre eigene Webseiten, Animationen und viele andere zu bilden. Codemoji benutzt man im Unterricht wegen dem schnellen Lernen und weil die unnötige Codesyntax eliminiert wird. Der Lehrstoff wird in den Kursen organisiert, wo die Studenten in den Lektionen nach ihrem eigenen Tempo lernen. Dank Codemoji ist der Unterricht spannend auch unterhaltend und dabei die Sichtbaustellen benutzt (Codemoji, 2018).

Bild 9 – Umgebung von Programm Codemoji (Codemoji)

Ende

In dem Beitrag behandelten wir die Programmiersprache Scratch und seine Möglichkeiten. Scratch setzt sich ein Ziel, die Kinder der ersten Stufen der Grundschule die Grundlagen der Algorithmisierung und Computergrafikbildung kennenzulernen. Programmiersprache Scratch gehört in die Gruppe der sogenannten Kinderprogrammiersprachen. Scratch wird von Anfang an so konstruiert und gebildet, daß die Kinder Programmieren lernen. Scratchumgebung wird vorgeschlagen, damit es einfach und leicht beherrschbar war. Atraktivität der Programmierumgebung Scratch im Unterricht besteht in der visuellen Programmbildung mit Hilfe Blöckenzusammenfalten. Damit verschwindet für die Kinder die langweilige Korrigierung der Tippfehler, die Bewachung des fehlenden Strichpunkte und den andren Befehlen, die mit den üblichen Programmiersprachen verbunden sind und deshalb ist Scratch sehr passend für die Anwendung in den Grundschulen.

Weitere Programmierumgebung, die wir beschreiben, gibt es nur in Englisch, was sehr positiv zum Informatikunterricht aber auch Fremdsprache beiträgt. Außer Tynker sind die anderen Umgebungen in den Sprachen JavaScript, Python und HTML. Alle sind für die Kinder von 5-16 Jahren geeignet und jeder Benutzer kann sich auswählen, welche ihm besser passen wird. CodeCombat bietet das Spiel des Adventuretyps an, wo der Spieler anstatt der direkten Beherrschung, die Figur programmiert. In Code Avengers und Codemoji lernen die Kinder die Webseiten kodieren, Appes, Spiele usw. LightBot ist ein Videospiel mit den Grundlagen von Programmieren.

Zum Schluss können wir anführen, daß Scratch und seine andere Möglichkeiten sehr passend zur Entwicklung des Informatikdenkens und damit auch zum Informatikunterricht beitragen können. Informatikunterricht hat in der tschechischen Schulen schon eine längere Tradition. Anfang der Eingliederung des informatischen Inhalts in die Lernplansstruktur neben der traditionellen Fächern wie zum Beispiel Mathematik, Chemie, Biologie, Sport waren überhaupt nicht einfach, sie kommen bis zum Jahr 1991. Schon damals genehmigte das Ministerium für Schulwesen, Jugend und Sport der Tschechischen Republik den Lehrplan der Grundschule für den Fach Informatik. Obwohl es sich um einen fakultativen Fach handelte, wurde er in damaliger Zeit sehr gut inhaltlich verarbeitet (srov. Dostál, J., Wang, X., 2017).

Literatur

- Klupal, L. (2016). Aplikace: ScratchJr. *Apple ve školství*. [cit. 2018-03-18]. Dostupné z: <http://avs.vyuka.info/aplikace/scratchjr>
- Brázdilová, M. (2017). Apple společně s firmou Tynker chtějí naučit programovat ve Swiftu i děti v mateřské škole. *LETEM SVĚTEM APPLE*. [cit. 2018-08-08]. Dostupné z: <https://www.letemsvetemapple.eu/2017/05/02/apple-spolupracuje-firmou-tynker-spolecne-nauci-programovat-ve-swiftu-i-deti-materske-skolce/>
- Code Avengers [online]. [cit. 2018-08-08]. Dostupné z: <https://www.codeavengers.com/>
- CodeCombat [online]. [cit. 2018-08-09]. Dostupné z: <https://codecombat.com/>
- Codemoji [online]. [cit. 2018-08-09]. Dostupné z: <https://www.codemoji.com/>
- Dostál, J. & Wang, X. (2017). The Curriculum Content of Informatics as a Teaching Subject at Basic Schools in the Czech Republic between 1996 and 2005. In *EDULEARN17 Proceedings*. Madrid: International Association of Technology, Education and Development. s. 9525-9534. doi 10.21125/edulearn.2017.0796
- Lightbot [online]. [cit. 2018-08-09]. Dostupné z: <http://lightbot.com/>
- Lightbot: Creating and Running a Program (2017). In *Lightbot*. [cit. 2018-08-09]. Dostupné z: http://lightbot.com/Lightbot_BasicProgramming.pdf
- Programování ve Scratchi [online]. [cit. 2018-03-03]. Dostupné z: <https://programovanivescratchi.webnode.cz/programovaci-jazyk-scratch/>
- Resnick, M., Maloney, J. & Eastmond, E., eds. (2009). Scratch: programming for all. In *Communications of the ACM*, 52(11). doi: 10.1145/1592761.1592779
- Scratch [online]. [cit. 2018-08-08]. Dostupné z: https://scratch.mit.edu/projects/editor/?tip_bar=home
- ScratchJr - About [online]. [cit. 2018-03-18]. Dostupné z: <http://scratchjr.org/about/info>
- ScratchJr: Coding for Young Kids [online]. [cit. 2018-03-04]. Dostupné z: <https://www.kickstarter.com/projects/2023634798/scratchjr-coding-for-young-kids>
- Scratch Program. Scratch Wiki [online]. [cit. 2018-03-03]. Dostupné z: https://wiki.scratch.mit.edu/wiki/Scratch_Program
- Scratch wiki [online]. [cit. 2018-03-06]. Dostupné z: <https://wiki.scratch.mit.edu/wiki/Pyonkee>
- Snap! – důstojný nástupce programovacího jazyka Scratch [online]. In *root.cz*. [cit. 2018-03-04]. Dostupné z: <https://www.root.cz/clanky/snap-dustojny-nastupce-programovaciho-jazyka-scratch/>
- Strategie digitálního vzdělávání do roku 2020. (2014). MŠMT. [online]. Dostupné z <http://www.msmt.cz/file/34429/>
- Tynker: Coding for kids [online]. [cit. 2018-08-08]. Dostupné z: <https://www.tynker.com/>
- Úvod do jazyka Scratch. (2013). In *zdrojak.cz*. [cit. 2018-03-03]. Dostupné z: <https://www.zdrojak.cz/clanky/uvod-jazyka-scratch/>

Yaroslavski, D. (2014). Developer spotlight. In *openfl.org*. [cit. 2018-08-09]. Dostupné z: <http://www.openfl.org/blog/2014/11/08/developer-spotlight-danny-yaroslavski/>

Kontaktadresse:

Hana Bučková, Mgr.,

Katedra technické a informační výchovy, Pedagogická fakulta UP, Žižkovo nám. 5, 771 40

Olomouc, ČR, tel.: 00420 585 635 820, e-mail: hana.buckova02@upol.cz