

DOI: <https://doi.org/10.18454/COMP.2024.1.5>

ТЕНДЕНЦИИ РАЗВИТИЯ БАЗОВОГО КОМПЬЮТЕРНОГО ОБРАЗОВАНИЯ В УНИВЕРСИТЕТАХ

Обзор

Долинский М.С.^{1,*}

¹ORCID : 0000-0002-3057-4051;

¹Гомельский государственный университет имени Ф. Скорины, Гомель, Беларусь

* Корреспондирующий автор (dolinsky[at]gsu.by)

Аннотация

Базовое компьютерное образование в университетах испытывает огромные проблемы. С одной стороны, очень быстро увеличивается объём знаний, которым должен обладать выпускник университета. С другой стороны, контингент студентов сильно разнится по уровню подготовки и мотивации и уровень этой дифференциации постоянно растёт. Как следствие, увеличивается сложность обучения и процент недоучившихся. Учёные и педагоги ищут решение этих проблем по следующим направлениям: пересмотр необходимых к получению в университете знаний в сторону реальности их получения в отведённое время; использование новых информационных технологий для упрощения процесса обучения и повышения его качества; разработка новейших методик обучения, учитывающих реалии. В данной работе представлен стратегический документ в области компьютерного образования в университетах – Computing Curriculum 2020, а также обзор направлений развития базового компьютерного образования, таких как: обучение с помощью искусственного интеллекта, виртуальные лаборатории, микропроцессорные комплекты и робототехника, WEB-системы для дистанционного и смешанного обучения, разработка мобильных приложений, визуальное программирование, геймификация, архитектура компьютеров, языки программирования, технологии обучения. Кроме того, автор приводит свой опыт и видение преподавания базового компьютерного образования в университетах.

Ключевые слова: обзор литературы, базовое компьютерное образование в вузах.

TENDENCIES IN THE DEVELOPMENT OF BASIC COMPUTER EDUCATION IN UNIVERSITIES

Review article

Dolinsky M.S.^{1,*}

¹ORCID : 0000-0002-3057-4051;

¹Francisk Skorina Gomel State University, Gomel, Belarus

* Corresponding author (dolinsky[at]gsu.by)

Abstract

Basic computer education in universities is facing enormous difficulties. On the one hand, the amount of knowledge that a university graduate must possess is increasing very rapidly. On the other hand, the population of students varies greatly in terms of their level of preparation and motivation, and the level of this differentiation is constantly growing. As a consequence, the complexity of learning and the percentage of undergraduates are increasing. Scientists and educators are looking for solutions to these problems in the following directions: revision of the knowledge required to be obtained at the university in the direction of the reality of obtaining them in the given time; use of new information technologies to simplify the learning process and improve its quality; development of new teaching methods that take into account the realities. This article presents a strategic document in the field of computer education in universities – Computing Curriculum 2020, as well as an overview of the directions of development of basic computer education, such as: learning with artificial intelligence, virtual laboratories, microprocessor suites and robotics, WEB-systems for distance and blended learning, mobile application development, visual programming, gamification, computer architecture, programming languages, learning technologies. In addition, the author provides their experience and vision of teaching basic computer education in universities.

Keywords: literature review, basic computer education in universities.

Введение

Под базовым компьютерным образованием автор статьи, как и разработчики документа Computing Curricula 2020 [5], понимает совокупность предметов, изучаемых, как правило, на первом/втором курсах университетов и посвящённых внутреннему устройству, а также организации и функционированию компьютеров, включая программирование на ассемблере, и основам алгоритмизации и программирования на языке высокого уровня. Эта совокупность предметов определяется как базовая (ядро/core в терминологии Computing Curricula) поскольку изучение всех других компьютерных дисциплин базируется на знаниях, умениях и навыках, полученных в ходе изучения этой совокупности предметов.

Цель исследования – изучение и освещение тенденций в развитии обучения базовым компьютерным дисциплинам, таким как «Архитектура компьютеров» и «Алгоритмизация и программирование». Задачи исследования – анализ и обобщение мирового опыта обучения указанным базовым компьютерным дисциплинам, а также изложение авторского опыта работы в этом направлении. Практическая значимость заключается в экономии огромного количества времени

всех читателей обзорной статьи, которое потратил автор на поиск и фильтрацию нужной информации, выстраивание систематического взгляда на поднятую тематику.

Othman [35] отмечает, что программирование компьютеров – один из базовых предметов технических университетов, однако результативность обучения студентов продолжает снижаться по следующим причинам по мнению студентов: использование обычных статических материалов, обучение в больших группах студентов, концентрация преподавателей на синтаксисе языков программирования вместо обучения решению задач, неверные подходы преподавателей к обучению и программированию, трудность в изучении предлагаемого языка программирования.

Belmar [2] сделал обзор проблем и достижений в обучении программированию и компьютерному мышлению по континентам: Европа, Северная Америка, Океания, Азия, Латинская Америка, Африка, отметив что в двух последних имеется значительное количество проблем. Он же кратко обозначил цель обучения программированию и компьютерному мышлению как развитие пяти навыков:

- абстракция (определение какую информацию о сущности/объекте необходимо сохранить, а какую игнорировать);
- обобщение (формулирование решения в общих терминах так, чтобы оно могло быть применено для решения различных задач);
- декомпозиция (разбиение сложных задач на меньшие подзадачи, которые легче понять и решить);
- алгоритмизация (определение шаг за шагом множества операций/действий для того, чтобы решить задачу);
- отладка (локализация и устранение ошибок).

Shanmugam [39] выполнил обзор современных методов обучения программированию, включая традиционные подходы, обучение по запросам, проектное обучение, обучение в группах, онлайн-обучение, «перевернутый урок», адаптивное обучение.

Основные результаты

2.1. Computing Curricula 2020

31 декабря 2020 года две крупнейшие международные организации, объединяющие компьютерных специалистов: Association for Computing Machinery (ACM) и Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS) выпустили совместный 205-страничный документ Computing Curricula 2020 (CC 2020) [5], определяющий стратегию компьютерного образования по семи направлениям: Computer Engineering, Computer Science, Cybersecurity, Information Systems, Information Technology, Software Engineering, Data Science (under development).

Ключевое отличие CC 2020 от предыдущей версии документа CC 2005 – переход от знаний к компетенциям. Вот как это объясняется в CC 2020:

Достигнут консенсус, что карьерный успех специалиста требует три вещи:

- знание – «знать-что» – формулирование ключевых концепций и содержания образования;
- умения – «знать-как» – способность решать задачи, достигая результата;
- диспозиции – «знать-почему» – интеллектуальные, социальные и моральные тенденции.

Таким образом, Компетенция = Знания + Умения + Диспозиции.

CC2020 формулирует будущее компьютерного образования, следуя четырём принципам.

1. Сохранять и поддерживать идеи компьютерного образования как сейчас, так и в будущем во всем мире.
2. Охватывать тенденции из индустрии, научных исследований, всех спектров общества.
3. Быть расширяемым и поддерживать существующие, зарождающиеся и будущие тенденции в компьютерном образовании.

4. Быть гибким, чтобы достигать глобального принятия и адаптации к множеству образовательных контекстов.

Компьютерное образование включает семейство следующих областей:

- **Computer engineering** (разработка компьютеров) развивается от electrical engineering (электротехника);
- **Computer science** (компьютерная наука) эволюционирует в академическую дисциплину;
- **Information systems** (информационные системы) расширяются, поскольку компьютеры стали основой для организационных процессов и рабочих сред;
- **Information technology** (информационная технология) возникает как новая дисциплина, которая способствует построению и поддержке компьютерных инфраструктур;
- **Software engineering** (разработка программного обеспечения) развивается как дисциплина, базирующаяся на “Computer science” и “Computer engineering”.

В качестве ядра обучения компьютерным знаниям выделяются такие компоненты как «Архитектура компьютеров» и «Языки программирования». Отдельно в CC 2020 уделено внимание обучения компьютерным технологиям в начальной и средней школе.

Далее представлены характерные примеры работы в этих направлениях различных организаций и авторов в последние годы.

2.2. Обучение с помощью искусственного интеллекта

Oosterwyk et al. [34] отмечает, что AI-программы, такие как ChatGPT в состоянии выполнять задания вводных курсов по программированию вместо студентов, при этом ChatGPT понимает не только текстовые описания заданий, но и программы, написанные на различных языках программирования таких как Python, Java, C++ и др. Соответственно, перед преподавателями стоит задача изменения соответствующих заданий.

Jacques et al. [29] представил своё видение обучения вводных курсов по программированию с помощью искусственного интеллекта (ИИ). Среди прочего, он предлагает давать задания на использование ИИ генерирующего программы по различным представлениям задач: текстовые представления (алгоритмы, написанные на языке программирования или псевдокоде), визуальные представления (например, UML-диаграммы), числовые и формульные

представления. Другой вариант заданий – генерация нескольких решений одной и той же задачи с последующим сравнительным анализом реализаций. Третий – объяснение студентом решений, сгенерированных ИИ.

Bengtsson и Kaliff [3] предлагают использовать ChatCPT для оценивания программ, разработанных студентами, отмечая, что исторически это делалось либо проверкой учителем вручную, либо автоматической проверкой по тестам.

Denny [10] отмечает, что традиционные подходы к обучению программированию фокусируются на практике написания кода. Но с появлением AI-генераторов кода требуется изменять эти подходы в сторону развития навыков формирования эффективных текстовых запросов к AI-генераторам кодов.

Zheng [47] описывает использование ChatGPT в области data science (науки о данных), указывая в качестве достоинств такого подхода персонализированное обучение, понимание концепций, генерация и объяснение кода, оценивание. Data science, по его мнению, включает три существенных обязательных курса – базы данных, анализ данных, машинное обучение.

Farah [26] представил авторский ChatBot, который помогает студентам осваивать оформление программ на языке Python в соответствии со стандартом PEP-8.

2.3. Виртуальные лаборатории

Zargham и Cheng [45] представляют новую платформу и метод кодирования. Ученики 5-12 классов используют среду виртуальной реальности для написания кода для создания интересного мира, в частности американских горок мечты. Во время написания кода они наблюдают за развитием своих американских горок в пошаговой 3D-среде. После завершения проектирования американских горок они могут развернуть его на смартфоне. Затем они могут кататься на американских горках в виртуальной среде, поместив телефон в гарнитуру виртуальной реальности. Разработанная и реализованная платформа виртуальной реальности выходит за рамки простых визуальных стимулов и позволяет учащимся принимать непосредственное участие и испытывать свой дизайн в осязаемой, интерактивной и оперативной системе.

Xu [8] описывает виртуальную лабораторию, основанную на использовании аппаратной платформы DSO38Lab, обеспечивающей сбор данных и программном обеспечении, поддерживающим графический дисплей, отладку программ, генерацию сигналов, временной и частотный анализы, цифровую фильтрацию и многое другое.

2.4. Микропроцессорные комплекты и робототехника

Платформа Arduino стала чрезвычайно популярной на всех уровнях технического обучения в последнее десятилетие. В дополнение к дешёвому аппаратному обеспечению платформа предлагает бесплатную среду разработки. Также имеются дополнительные аппаратные модули, дополнительные программные модули (библиотеки) и учебные материалы. Важным дополнением к платформе Arduino является среда симуляции и отладки TinkerCAD, а также возможность взаимодействия с программным обеспечением MATLAB-Simulink. Černák [6] приводит методологию совместного использования Arduino, TinkerCAD, MATLAB-Simulink при обучении студентов робототехнике.

Lewis [30] описывает методологию проектного обучения на базе микропроцессорного комплекта Arduino, в рамках которой студенты разрабатывали проекты управления ветряной мельницей и системы фильтрации воды.

Cruz [9] описывает использование в учебном процессе системы на кристалле ESP8266 и ESP32. Они совместимы с Arduino, имеют изменяемое программное обеспечение и низкую стоимость. В ходе занятий студенты осваивают разработку программ с использованием WiFi и BlueTooth.

Zimmermann [49] приводит пример обучения студентов визуальному программированию гуманоидных роботов. Такое обучение очень интересно, особенно начинающим. Обычно гуманоидный робот состоит из сенсоров и актуаторов, таких как тело, голова, две руки, две ноги.

Vrbanić и Kocijančić [41] обсуждают как эффективнее обучать начинающих программированию микроконтроллеров – в тестовой или графической среде, сравнивая достоинства и недостатки обоих подходов.

2.5. WEB-системы для дистанционного и смешанного обучения

Mamman [31] описывает WEB-платформу для обучения программированию на языке Python, разработанную жителей таких стран как Нигерия, Судан, Гана и Камерун, говорящих на языке Hausa. WEB-платформа обеспечивает создание и наполнение учебных курсов, форум для общения студентов и преподавателей и выдачу сертификатов о прохождении курсов.

Aldalur [1] предлагает свою WEB-платформу WEbQuest для изучения программирования на C в среде Visual Studio.

Zhuang [48] представил WEB-платформу OPTES (Online Programming Training Estimation System) которая обеспечивает интегрированную среду для назначения внеклассных заданий по программированию, обеспечивающую также отслеживание результативности студентов в течение семестра.

Nguyen и Ha [33] разработали систему ProgEdu для автоматического оценивания работ студентов по программированию, которая близка к разработке реального программного обеспечения, помогая студентам понять фундаментальные концепции и получить опыт работы в современной среде разработки программного обеспечения.

Fonseca [27] описывает авторскую WEB-платформу для обучения студентов программированию логических контроллеров, в ближайших планах развитие для программированию IoT (интернета вещей) с поддержкой Arduino и микроконтроллеров ESP32 и STM32 и процессоров цифровой обработки сигналов.

Zhang [46] представляет практику использования платформы EduCoder для обучения программированию. Отмечается её существенная помощь в трёх аспектах процесса обучения: организация занятий, проектирование онлайн экспериментов, отслеживание успеваемости студентов. Важные результаты – снижение нагрузки на учителя, повышение эффективности и качества обучения.

2.6. Разработка мобильных приложений

Özdemir [36] отмечает, что имеется значительное количество языков программирования и средств разработки мобильных приложений. И подробно описывает одно из них – Kulika, которое использовал в обучении студентов, поскольку оно эффективно поддерживает все стадии разработки: анализ, проектирование, разработку, тестирование и отладку.

2.7. Визуальное программирование

Cheng [8] приводит мнение, что традиционные текстовые языки программирования создают трудности в обучении начинающих. Как альтернатива предлагается использование визуальных языков программирования. Приводится пример использования визуальных языков программирования для формирования оценивающих тестов.

2.8. Геймификация

Zakiah [44] приводит обзор средств геймификации при изучении языков программирования, включающих инкорпорацию таких игровых механик как баллы, бейджи, таблицы лидеров. Такая обратная связь и ощущение прогресса поощряют студентов интенсифицировать обучение.

Pusztai [37] описывает авторский опыт внедрения геймификации в университетские курсы «Алгоритмы и структуры данных» и «Методы программирования».

2.9. Архитектура компьютеров

Quaratiello [38] описывает авторский курс по ассемблеру и низкоуровневому C, введённый в MIT с 2022 года на специальностях Electrical Engineering and Computer Science, и Computer Science and Engineering.

2.10. Языки программирования

Grindei [28] описывает авторский подход к обучению программированию и языкам программирования C, C++, C# и Python с использованием специальных инженерных задач для студентов.

Mosobalaje и Orodu [32] описывают курс программирования на Python, включающем ввод/вывод, объекты (значения, переменные, ключевые слова), структуры условий и повторения, функции, списки, кортежи и словари, а также применение программ в различных предметных областях.

Birkenkrahe [4] представил авторский подход к изучению дисциплин, связанных с “data science”, в рамках которых изучаются языки программирования R, Python, C++, SQL.

Cervoni и Basseur [7] описывают методологию обучения языку программирования PROLOG.

2.11. Технологии обучения

Wittie [43] описывает пример применения методологии “Teaching for Mastery” для обучения программированию. Суть её заключается в том, что студент переходит к изучению следующей темы только после того, как он освоил предыдущую, и каждый студент двигается по учебному материалу с собственной скоростью. Основой обучения является подход flipped classroom («перевёрнутый класс»). Теоретические сведения содержатся в подготовленных кратких (3-12 минут) видео-лекциях. Для проверки знаний студентов используется система автоматической проверки. Для нуждающихся студентов проводятся еженедельные консультации.

Vakaliuk [40] представляет методику обучения программированию с использованием Massive Open Online Courses (MOOC): Sololearn, CodesDope, CodinGeek, Javatpoint, Pskills, Codesrecker.

Авторский опыт

Автор в течение нескольких десятилетий занимается обучением программированию школьников и студентов, а также обучением основам цифровой техники и архитектуре компьютеров студентов факультета математики и технологий программирования, специальностей «Программное обеспечение», «Информатика и технологии программирования», «Прикладная информатика». Для повышения эффективности этого обучения автором были разработаны специальные методики, а для их полноценной реализации было разработано специальное программное обеспечение под руководством автора.

3.1. Инструментальная система дистанционного обучения DL.GSU.BY

Система HLCCAD (High-Level Chip Computer-Aided Design System) позволяет интерактивно и визуально рисовать функциональные схемы цифровых устройств, моделировать их работу на входных тестах (задаваемых интерактивно или в файлах), анализировать результаты моделирования, находить и исправлять ошибки [15]. Система HLCCAD обеспечивает автоматическое сравнение результатов моделирования с эталонными результатами, подготовленными автором задания. Таким образом, сообщая студенту о правильности/неправильности выполнения задания. Кроме того, система HLCCAD интегрирована с системой DL.GSU.BY на основании чего в соответствующих учебных курсах задания предъявляются студентам через сайт DL и через него же студенты отправляют на проверку свои схемы. Результат проверки схемы системой HLCCAD сообщается студенту и вносится в таблицу результатов.

3.2. WINTER- среда отладки программ для микропроцессоров/микроконтроллеров

Система Winter позволяет писать и отлаживать ассемблерные программы для микроконтроллеров и микропроцессоров Intel 8051, Intel 8086, Atmel AT90S2313, Atmel AT90S2323, Motorola M68HC05 and Motorola M68HC08, TMS370 [14]. Система Winter также позволяет настраивать себя на другие модели микропроцессоров/микроконтроллеров. Система Winter интегрирована с системой DL.GSU.BY на основании чего в соответствующих учебных курсах задания предъявляются студентам через сайт DL и через него же студенты отправляют на проверку свои программы. Результат проверки программы системой WINTER сообщается студенту и вносится в таблицу результатов.

3.3. Методика обучения студентов основам цифровой техники

Основные принципы методики обучения основам цифровой техники [14], [11] и архитектуры компьютеров [16]:

- практико-ориентированный подход – основная цель курса – научить студентов проектировать схемы и писать программы на ассемблере и C;
- автоматическая проверка решений (схем и программ);

- активное использование средств моделирования, отладки и визуализации в целях повышения качества усваивания материала;
- значительное количество (тысячи) обучающих заданий различной сложности, поддерживающих персонализированное обучение.

3.4. Методика обучения студентов-первокурсников программированию

Основные принципы методики обучения студентов-первокурсников программированию [25], [13]:

- персонализированный подход к обучению (выбор студентом):
 - языка программирования Pascal, C, C++, C#, Java, Python;
 - уровня сложности заданий в текущем обучении.
- практико-ориентированный подход – основная цель курса – научить студента писать и отлаживать программы на выбранном им языке программирования;
- автоматизированные объективные контроль и оценивание знаний;
- основные функции преподавателя:
 - научить студента учиться;
 - оказывать индивидуальную помощь студентам;
- наличие всего необходимого материала в учебном курсе на различных уровнях абстракции.

В качестве оценки эффективности предложенных программных средств и методик обучения можно привести следующие соображения:

- инструментальная система дистанционного обучения, автоматически выдающая условия заданий и проверяющая решения студентов, а также выполняющая некоторые обучающие функции, резко интенсифицирует учебный процесс, позволяя выполнять больше работы в течение отведённого времени, тем самым повышая качество обучения;
- средства симуляции и отладки программ и схем цифровых устройств обеспечивающие визуализацию, улучшают восприятие и понимание учебного материала;
- вся система обучения положительно воспринимается студентами в учебном процессе;
- более подробная аргументация приведена в статьях, на которые приведены ссылки при упоминании программных средств и методик;
- прямым подтверждением эффективности предложенных программных средств и методик является также и наличие многочисленных публикаций автора в англоязычных специализированных журналах: “WSEAS Transactions on Advances in Engineering Education”, “International Journal of Education and Learning Systems”, “International Journal of Circuits and Electronics”, “Journal of Information Technology and Digital World”, “Global Journal of Computer Science and Technology Interdisciplinary”.

3.5. Методика обучения программированию школьников-старшеклассников (классы 9-11)

Автор абсолютно солидарен с [5] в том, что, для того чтобы получить в вузе качественных студентов, надо работать со школьниками, и занимается этим всю свою жизнь. Основные принципы методики обучения программированию школьников-старшеклассников [23]:

- подготовка к победам в олимпиадах по информатике: международных, национальных, региональных. Уровень достижений зависит не только и не столько от способностей, сколько от вложенного труда и потраченного времени;
- последовательное обучение от простого к сложному с перманентной проверкой результативности обучения – еженедельной в тренировочных олимпиадах и ежемесячной в раундах на Codeforces.com;
- значительное количество индивидуальной самостоятельной работы школьника.

3.6. Методика обучения программированию в средней школе (классы 5-8)

Для того чтобы старше школьники могли успешно выступать в международных и национальных олимпиадах, они должны начинать занятия, как минимум, в средней школе. Основные принципы обучения программированию средних школьников [12]:

- персонализированное адаптивное обучение;
- регулярное участие в тренировочных и официальных олимпиадах;
- развитие самостоятельности и креативности.

3.7. Методика обучения программированию в начальной школе (классы 1-4)

В отличие от доминирующего подхода блочного визуального программирования с начинающими (не только в младшей, но даже и в средней школе), автор придерживается противоположной точки зрения, что начинать обучение можно и нужно с текстового программирования, а все возникающие проблемы решать с помощью соответствующего методического и программного обеспечения [19], [20], [21]. Более того, под руководством автора разработаны и ежегодно проводятся специальные олимпиады по программированию для учеников начальной школы [18]. Для выравнивания уровней подготовки первоклассников, а также для возможности заниматься с дошкольниками был разработан и используется специальный курс «Учимся думать» [24].

В качестве аргументов в пользу эффективности предлагаемых подходов к обучению можно привести следующие:

- школьники ученики автора завоевали более 20 медалей на международных олимпиадах школьников по информатике (IOI);
- автор регулярно публикует статьи о своей работе в журнале “Olympiad in Informatics”;
- автор неоднократно становился победителем конкурса по методике преподавания информатике школьникам, который проводится журналами «Информатика и образование» и «Информатика в школе».

Заключение

В данной работе представлен обзор направлений развития базового компьютерного образования, таких как: обучение с помощью искусственного интеллекта, виртуальные лаборатории, микропроцессорные комплекты и робототехника, WEB-системы для дистанционного и смешанного обучения, разработка мобильных приложений,

визуальное программирование, геймификация, архитектура компьютеров, алгоритмизация и языки программирования, технологии обучения.

Кроме того, автор приводит свой опыт и видение преподавания базового компьютерного образования в университетах, рассматривая такие вопросы как специально разработанное программное обеспечение и собственные методики преподавания основ цифровой техники и программирования.

В качестве основных выводов можно предложить следующие: методы и средства повышения качества обучения базовым компьютерным дисциплинам развивались, развиваются и будут развиваться, адаптируясь к текущим реалиям.

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы на английском языке / References in English

1. Aldalur I. Improving Programming Learning in Engineering Students through Discovery Learning / I. Aldalur and X. Sagarna // IEEE Revista Iberoamericana de Tecnologías del Aprendizaje (Early Access). — DOI: 10.1109/RITA.2023.3301409
2. Belmar H.M. Review on the Teaching of Programming and Computational Thinking in the World / H.M. Belmar // Front. Comput. — Sci. 4, 997222. — DOI: 10.3389/fcomp.2022.997222
3. Bengtsson D. Assessment Accuracy of a Large Language Model on Programming Assignments Degree Project in Computer Science and Engineering / D. Bengtsson, A. Kaliff. — 2023. — URL: <https://www.diva-portal.org/smash/get/diva2:1779792/FULLTEXT01.pdf> (accessed: 01.11.2023)
4. Birkenkrahe M. Teaching Data Science with Literate Programming Tools / M. Birkenkrahe // Preprints. — 2023. — 2023071847. — DOI: 10.20944/preprints202307.1847.v1
5. CC 2020: Computing Curricula 2020: Paradigms for Global Computing Education. — URL: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf> (accessed: 01.11.2023)
6. Čermák R. Teaching Mechatronics, Robotics and Automation with Arduino EDULEARN23 Proceedings / R. Čermák, V. Mašek, D. Fait. — P. 6497-6502. — DOI: 10.21125/edulearn.2023.1720
7. Cervoni L. PROLOG, a Technological Approach to Teaching: the Case of Mathematics and Problem Solving International Conferences Educational Technologies 2022 / L. Cervoni and J. Brasseur // Sustainability, Technology and Education. — 2022. — URL: https://www.iadisportal.org/components/com_booklibrary/ebooks/202210R027.pdf (accessed: 01.11.2023)
8. Cheng Y.P. Applying Computational Thinking and Formative Assessment to Enhance the Learning Performance of Students in Virtual Programming Language / Y.P. Cheng, S.C. Cheng, M. Yang [et al.] In: Huang, Y.M., Rocha, T. (eds) // Innovative Technologies and Learning. ICITL 2023. Lecture Notes in Computer Science. — vol. 14099. — Springer, Cham. — DOI: 10.1007/978-3-031-40113-8_13
9. Cruz N.C. Low-Cost Chip Programming for Teaching and Learning Digital Skills / N.C. Cruz, J.F. Estrada, M. Lupión [et al.] — EDULEARN23 Proceedings. — P. 2662-2671. — URL: <https://library.iated.org/view/CRUZ2023LOW> (accessed: 01.11.2023)
10. Denny P. Promptly: Using Prompt Problems to Teach Learners How to Effectively Utilize AI Code Generators / P. Denny, A. Luxton-Reilly, J. Leinonen [et al.] // arXiv:2307.16364v1 [cs.HC] 31 Jul 2023. — URL: <https://arxiv.org/pdf/2307.16364.pdf> (accessed: 01.11.2023)
11. Dolinsky M. One Approach to Study the Topic "Logic and Combinational Circuits" / M. Dolinsky // WSEAS Transactions on Advances in Engineering Education. — 2023a. — 20. — 14-24. — URL: <https://wseas.com/journals/articles.php?id=7602> (accessed: 01.11.2023)
12. Dolinsky M. Secondary School Programming Olympiads in Gomel Region (Belarus) / M. Dolinsky // Olympiads in Informatics. — 2023b. — 17. — 131-142. — URL: https://ioinformatics.org/journal/v17_2023_131_142.pdf (accessed: 01.11.2023)
13. Dolinsky M. Teaching Algorithms and Programming First Year University Students on Base of Distance Learning System DL.GSU.BY / M. Dolinsky // WSEAS Transactions on Advances in Engineering Education. — 2022a. — 19. — 52-57. — URL: <https://wseas.com/journals/articles.php?id=5611> (accessed: 01.11.2023)
14. Dolinsky M. Experience of Blended Learning the Fundamentals of Digital Electronics for First/Second Year University students On Base of Distance Learning System DL.GSU.BY / M. Dolinsky // International Journal of Education and Learning Systems. — 2022b. — 7. — 59-642. — URL: <https://www.iaras.org/iaras/home/caijels/experience-of-blended-learning-the-fundamentals-of-digital-electronics-for-first-second-year-university-students-on-base-of-distance-learning-system-dl-gsu-by> (accessed: 01.11.2023)
15. Dolinsky M. Tool HLCCAD for Blended Learning the Fundamentals of Digital Electronics / M. Dolinsky // International Journal of Circuits and Electronics. — 2022c. — 7. — 47-55. — URL: <https://www.iaras.org/iaras/home/computer-science-communications/caijce/tool-hlccad-for-blended-learning-the-fundamentals-of-digital-electronics> (accessed: 01.11.2023)

16. Dolinsky M. Experience of Blended Learning in the Subject – Architecture of Computers / M. Dolinsky // Journal of Information Technology and Digital World. — 2022d. — 4(3). — 167-182. — URL: <https://irojournals.com/itdw/article/view/4/3/4> (accessed: 01.11.2023)
17. Dolinsky M. Instrumental System of Distance Learning DL.GSU.BY and Examples of its Application / M. Dolinsky // Global Journal of Computer Science and Technology Interdisciplinary. — 2022e. — 22(1). — 45-53. — URL: https://globaljournals.org/GJCST_Volume22/6-Instrumental-System-of-Distance-Learning.pdf (accessed: 01.11.2023)
18. Dolinsky M. Primary School Programming Olympiads in Gomel Region (Belarus) / M. Dolinsky // Olympiads in Informatics. — 2022f. — 16. — 107-123. — URL: https://ioinformatics.org/journal/v16_2022_107_123.pdf (accessed: 01.11.2023)
19. Dolinsky M. The Technology of Differentiated Instruction in Text Programming in Elementary School Based on the Website dl.gsu.by / M. Dolinsky, M. Dolinskaya // Olympiads in Informatics. — 2020. — 14. — 37-46. — URL: https://ioinformatics.org/journal/v14_2020_37_46.pdf (accessed: 01.11.2023)
20. Dolinsky M. Training in Writing the Simplest Programs from Early Ages / M. Dolinsky, M. Dolinskaya // Olympiads in Informatics. — 2019. — 13. — 21-30. — URL: https://ioinformatics.org/journal/v13_2019_21_30.pdf (accessed: 01.11.2023)
21. Dolinsky M. How to Start Teaching Programming at Primary School / M. Dolinsky, M. Dolinskaya // Olympiads in Informatics. — 2018. — 12. — 13-24. — URL: https://ioinformatics.org/journal/v12_2018_13_24.pdf (accessed: 01.11.2023)
22. Dolinsky M. A New Generation Distance Learning System for Programming and Olympiads in Informatics / M. Dolinsky // Olympiads in Informatics. — 2017. — 11. — 29-39. — URL: https://ioinformatics.org/journal/v11_2017_29_39.pdf (accessed: 01.11.2023)
23. Dolinsky M. Gomel Training School for Olympiads in Informatics / M. Dolinsky // Olympiads in Informatics. — 2016. — 10. — 237-247. — URL: https://ioinformatics.org/journal/v10_2016_237_247.pdf (accessed: 01.11.2023)
24. Dolinsky M. Technology for the Development of Thinking of Preschool Children and Primary School Pupils / M. Dolinsky // Olympiads in Informatics. — 2014. — 8. — 63-68. — URL: https://ioinformatics.org/journal/v8_2014_63_68.pdf (accessed: 01.11.2023)
25. Dolinsky M. An Approach to Teaching Introductory-Level Computer Programming / M. Dolinsky // Olympiads in Informatics. — 2013. — 7. — 14-22. — URL: <https://ioinformatics.org/journal/INFOL114.pdf> (accessed: 01.11.2023)
26. Farah J. Harnessing Rule-Based Chatbots to Support Teaching Python / J. Farah, S. Ingram, A. Purohit [et al.] // Programming Best Practices ernational Conference on Interactive Collaborative Learning (ICL2023), Madrid, Spain, September 26-29, 2023. — URL: <https://infoscience.epfl.ch/record/303628> (accessed: 01.11.2023)
27. Fonseca I. Web-based Platform and a Methodology to Teach Programming Languages in Electrical Engineering Courses / I. Fonseca, N.C. Martins and F.A. Lopes // 2022 31st Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEIE), Coimbra, Portugal, 2022, 1-6. — DOI: 10.1109/EAEIE54893.2022.9820594
28. Grindei L. Project-oriented Approach in Teaching Programming to the First Year Undergraduate Students in Electrical Engineering / L. Grindei, C. Constantinescu, A. Bojita [et al.] // 2023 10th International Conference on Modern Power Systems (MPS), Cluj-Napoca, Romania, 2023, 1-4. — DOI: 10.1109/MPS58874.2023.10187591
29. Jacques L. Teaching CS-101 at the Dawn of ChatGPT / L. Jacques // ACM Inroads. — 2023. — 14(3):40–46. — DOI: 10.1145/3595634
30. Lewis J.E. Comparing First-Year Student Programming Confidence Perceptions Between Different Hands-On Projects / J.E. Lewis, N. Hawkins, B.S. Robinson // Paper presented at 2023 ASEE Annual Conference & Exposition, Baltimore, Maryland. — URL: <https://peer.asee.org/43274> (accessed: 01.11.2023)
31. Mamman M. Computerized E-learning Platform for Teaching Python Programming Using Hausa Language / M. Mamman, R.S. Rdadi, M.A. Sadiq // World Journal of Advanced Engineering Technology and Sciences. — 2023. — 09(02): 116–123. — DOI: 10.30574/wjaets.2023.9.2.0214
32. Mosobalaje O.O. Introducing Python Coding to Petroleum Engineering Undergraduates: Excerpts from a Teaching Experience / O.O. Mosobalaje and O.D. Orodu // Paper presented at the SPE Nigeria Annual International Conference and Exhibition, Lagos, Nigeria, July 2023. — DOI: 10.2118/217148-MS
33. Nguyen A. Teaching DevOps and Software Engineering Practices Using an Automated Programming Assessment System / A. Nguyen, V. Ha. — 20 Jul. 2023. — URL: <https://journal.tvu.edu.vn/index.php/journal/article/view/2104> (accessed: 01.11.2023)
34. Oosterwyk G. Beyond the Hype: A Cautionary Tale of ChatGPT in the Programming Classroom / G. Oosterwyk, P. Tsibolane, P. Kautondokwa [et al.] // Conference: SACLA2023. — URL: https://www.researchgate.net/publication/372495260_Beyond_the_Hype_A_Cautionary_Tale_of_ChatGPT_in_the_Programming_Classroom/link/64ba588fc41fb852dd8d4b11/download (accessed: 01.11.2023)
35. Othman J. Factors Affecting Engineering Students in Learning the Programming Subject at UiTM Pulau Pinang: A Study on Educators' Ability / J. Othman, S.B. Mahlan, R. Kadar [et al.] // International Journal of Academic Research in Progressive Education and Development. — 2023. — 12(2). — 944–953. — DOI: 10.6007/IJARPED/v12-i2/17359
36. Özdemir İ.H. Mobile Programming with Kuika: A Course Design / İ.H. Özdemir, T. Kışla and S.S. Tito [et al.] // 2023 32nd Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEIE), Eindhoven, Netherlands, 2023, 1-6. — DOI: 10.23919/EAEIE55804.2023.10181754
37. Pusztai K. Education supported by games even in university? / K. Pusztai // IntechOpen. — DOI: 10.5772/intechopen.1002208

38. Quaratiello G. An Introductory Low-level Programming Course for Students with a Python Background Master thesis / G. Quaratiello // *Engineering in Electrical Engineering and Computer Science*, Massachusetts Institute of Technology, June 2023. — URL: <https://dspace.mit.edu/handle/1721.1/151672> (accessed: 01.11.2023)
39. Shanmugam D.B. Approaches to Teaching Programming: a comprehensive review and analysis / D.B. Shanmugam, Dr.N. Vijayalakshmi, N. Revathi // chapter in book "Research in Multidisciplinary Subjects (Volume-2)" // under ed. of Biplab Auddy. — 2023.
40. Vakaliuk T. The Use of ICT of Educational Purposes in the Independent Work of Students in the Study of the Basics of Programming by Future Software Engineers / T. Vakaliuk, O. Chyzhmotria, D. Antoniuk [et al.] // In *Proceedings of the 2nd Myroslav I. Zhaldak Symposium on Advances in Educational Technology (AET 2021)*. — 644-657. — DOI: 10.5220/0012066700003431
41. Vrbanič F. Strategy for Learning Microcontroller Programming – a Graphical or a Textual start? / F. Vrbanič, S. Kocijančič // *Educ Inf Technol*. — 2023. — DOI: 10.1007/s10639-023-12024-9
42. Xu X. Teaching Reform of Virtual Instrument Technology and Application Course Based on OBE Concept / X. Xu // *Open Access Library Journal*. — 2023. — 10. — 1-9. — DOI: 10.4236/oalib.1110491
43. Wittie L. Incorporating a Teach for Mastery System in a Computer Science Course 2023 / L. Wittie // *ASEE Annual Conference & Exposition*, Baltimore, Maryland. — URL: <https://peer.asee.org/43624> (accessed: 01.11.2023)
44. Zakiah N. A Review of Gamification for Learning Programming Language / N. Zakiah, D. Khalid, W. Hidayah. — URL: <https://www.icbe.my/wp-content/uploads/2023/06/5-1.pdf> (accessed: 01.11.2023)
45. Zargham M. Virtual Reality Programming Platform for Grades 5-12 / M. Zargham & J. Cheng // In T. Bastiaens (Ed.), *Proceedings of EdMedia + Innovate Learning* (pp. 1067-1073). Vienna, Austria: Association for the Advancement of Computing in Education (AACE). Retrieved August 7, 2023. — URL: <https://www.learntechlib.org/primary/p/222616/> (accessed: 01.11.2023)
46. Zhang S. Exploring the Applications of EduCoder Platform in Blended Teaching for Computer Major / S. Zhang, J. Yang, X. Sang // *Journal of Education and Educational Research*. — 2023. — 4(2). — URL: <https://drpress.org/ojs/index.php/jeer/article/view/10819/10528> (accessed: 01.11.2023)
47. Zheng Y. ChatGPT for Teaching and Learning: An Experience from Data Science Education / Y. Zheng // In *The 23rd Annual Conference on Information Technology Education (SIGITE '23)*. — URL: <https://arxiv.org/pdf/2307.16650.pdf> (accessed: 01.11.2023)
48. Zhuang Y. OPTES: A Tool for Behavior-based Student Programming Progress Estimation / Y. Zhuang, L. Wang, M. Zhang [et al.] // *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. — 122-131. — DOI: 10.1109/COMPSAC57700.2023.00025.
49. Zimmermann M. Teaching Visual Programming: Humanoid Robot Programming as a Case Study / M. Zimmermann // *EDULEARN23 Proceedings*. — 6143-6149. — DOI: 10.21125/edulearn.2023.1598