

бастамалар, басқалары - шығармашылық өнімдер. Мұғалімнің міндеті – бастаманы баспай, оқушылардың ой-өрісін бағыттау [2].

Идеяның дамуы жобалау кезеңіне көшеді, мұнда студенттер өз өнімінен немесе қызметінен кім пайда көре алады, оның қалай жұмыс істейтіні, оны қалай жүзеге асыруға болатыны туралы ойлана бастайды. Мұнда мақсатты аудиторияны талдау, бәсекелестер мен аналогтарды зерттеу элементтері, сондай-ақ қарапайым ресурстарды жоспарлау және жұмыс кезеңдері пайда болады.

Әрине, мектеп қабырғасында толыққанды бизнес-жоспар талап етілмейді, бірақ жобаны басқарудың негізгі принциптерін (мақсаттар, міндеттер, орындау мерзімдері, қатысушылардың рөлдері) меңгеру керек. Мұғалім студенттерге идеяларды құрылымдауға және жобаның даму болашағын көруге мүмкіндік беретін дизайндық ойлау элементтерін, визуалды карталарды, ақыл-ой карталарын пайдалана алады. Студенттер прототиптерді жасай алады - қағаз макеттерінен графикалық редакторлардағы цифрлық үлгілерге, онлайн қызметтерде немесе тіпті қарапайым мобильді қосымшаларда.

Келесі маңызды кезең – стартап-жобаның тұсаукесері. Бұл үрдіс мектеп оқушыларын көпшілік алдында сөз сөйлеуге, дәлелдеуге, өзіне сенімділікке үйретеді. Тұсаукесер мектептегі іс-шараның бөлігі ретінде де, кеңірек платформада да – мектептік хакатон, идеялар фестивалі, байқау немесе онлайн қорғау түрінде өтуі мүмкін. Ата-аналарды, жергілікті кәсіпкерлерді, техникалық жоғары оқу орындарының студенттерін немесе IT-компаниялардың өкілдерін тарту іс-шараның маңыздылығын арттыруға және оған шынайылық беруге көмектеседі. Қазақстандағы мектеп стартаптарын сәтті қорғау мысалдары көптеген жобалар шынымен жауап тауып, нақты бизнестен немесе жергілікті биліктен қолдау алатынын көрсетеді [3].

Әдістеменің бірдей маңызды бөлігі - жобадан кейінгі рефлексия. Жобаны аяқтағаннан кейін студенттер тек орындалған жұмыс туралы есеп беріп қана қоймай, нақты нені үйренгенін, қандай дағдыларды игергенін, қандай қателіктер жібергенін және болашақта оларды қалай болдырмауға болатынын ойластыруы керек. Бұл сыни тұрғыдан ойлау мен өзін-өзі бағалау дағдыларын қалыптастыруға ықпал етеді.

Әдебиеттер тізімі

- [1] Қараев Ж.А. Информатика пәнін оқытудың ғылыми-әдістемелік негіздері. – Алматы: Рауан, 2019. – 240 б.
- [2] Әбдіғали Ә.Ә. Жобалық оқыту технологиясы: Теориясы мен тәжірибесі. – Нұр-Сұлтан: Фолиант, 2021. – 196 б.
- [3] Бидайбеков Е.Ы., Әбеннова З.А. Оқу үрдісінде ақпараттық технологияларды қолдану. – Алматы: Қазақ университеті, 2020. – 184 б.

МЕТОДИКА ПРЕПОДАВАНИЯ ПРОГРАММИРОВАНИЯ В МЛАДШИХ КЛАССАХ (5–7 КЛАССЫ)

Исаева А.К.¹, Ракишева Н.К.², Секен Е.³, Бейсен М.⁴

¹Қарагандинский университет имени академика Е.А. Букетова

¹E-mail: isa_aiga@mail.ru

Аннотация

В данной статье рассматриваются теоретические и практические аспекты методики преподавания программирования учащимся 5–7 классов. Анализируются современные подходы к обучению программированию в условиях общеобразовательной школы с учетом возрастных и психологических особенностей младших подростков. Обосновывается целесообразность использования визуальных языков программирования на начальном этапе, выявляются преимущества геймификации, проектной деятельности и индивидуализации обучения. Приводятся рекомендации по формированию алгоритмического мышления и последовательного перехода от блочного к текстовому программированию. Статья может быть полезна преподавателям информатики, методистам и разработчикам образовательных программ.

Современная школа сталкивается с необходимостью раннего формирования у обучающихся цифровых и вычислительных навыков. Одним из важнейших компонентов этой подготовки становится обучение программированию уже с младших классов. В условиях стремительного развития цифровых технологий и появления новых профессий, основанных на ИТ-компетенциях, программирование воспринимается не только как прикладной навык, но и как универсальный способ развития логического и алгоритмического мышления. Возраст учащихся 5–7 классов является особенно благоприятным для начала обучения: дети уже обладают достаточной зрелостью для абстрактного мышления, но ещё сохраняют высокую степень вовлечённости и интереса к игровой и исследовательской деятельности.

Методика преподавания программирования в этих классах должна учитывать как психологические особенности учащихся, так и дидактические принципы школьного образования. Прежде всего, необходимо учитывать, что учащиеся младших классов часто обладают наглядно-образным мышлением, слабо развитым абстрактным мышлением, и сравнительно ограниченной способностью к долгосрочной концентрации. Поэтому введение программирования должно происходить в щадящем, поэтапном режиме, с использованием визуальных языков программирования, таких как Scratch, Blockly, Code.org. Эти среды позволяют учащимся воспринимать команды и алгоритмы в виде ярких блоков и образов, а также быстро видеть результат своей работы, что повышает мотивацию и снижает уровень тревожности перед сложными задачами.

На начальном этапе особое значение приобретает игровая и проектная деятельность. Программы, созданные учащимися, могут быть связаны с созданием мультфильмов, простых игр, интерактивных рассказов. Такой подход помогает не только освоить базовые концепции, как последовательность действий, цикл, условие, но и формирует позитивное отношение к программированию как к творческому процессу. В дальнейшем обучение может сопровождаться интеграцией с другими предметами – математикой, естествознанием, русским или казахским языком, что позволяет формировать метапредметные связи и расширять кругозор.

Переход к текстовому программированию (например, на языке Python) должен происходить плавно и только после того, как учащиеся уверенно освоят принципы алгоритмизации. На этом этапе важно формировать умения писать код вручную, понимать синтаксис и

отлаживать программы. Однако и здесь не следует сразу перегружать учащихся техническими деталями. Методически оправданным считается подход, при котором текстовое программирование вводится через знакомые визуальные аналоги. Например, среды программирования типа Tynker и Scratch 3.0 позволяют переключаться между блочным и текстовым представлением, что снижает уровень когнитивной нагрузки.

Большое внимание в методике уделяется развитию навыков решения задач. Здесь важна постановка задач, ориентированных на практическое применение: автоматизация простых действий, решение задач на движение, обработку текста, создание калькуляторов. Такая направленность позволяет ученикам осознавать значимость программирования и видеть его прикладную ценность. Также важно развивать умения работы с ошибками: культура отладки должна прививаться с самого начала как неотъемлемая часть программирования.

Неотъемлемой частью методики является геймификация – использование игровых механик в обучении. Это могут быть как внешние системы мотивации (бейджи, уровни, награды), так и использование игровых платформ (например, CodeCombat, Lightbot), где программирование интегрировано в игровой сюжет. Такие средства способствуют вовлечению учеников и удержанию их внимания на протяжении всего занятия. Кроме того, обучение программированию должно строиться на принципах индивидуализации и дифференциации. Учащиеся обладают разной скоростью освоения материала и интересами, поэтому важно предлагать задания разного уровня сложности и давать возможность выбора задач. Эффективной формой работы являются мини-проекты, которые дети разрабатывают индивидуально или в парах. Это помогает развивать не только ИКТ-компетенции, но и коммуникативные навыки, умение планировать и доводить дело до конца.

Важную роль играют и технические средства: наличие компьютеров, интерактивных досок, стабильного доступа в интернет. Однако не менее важна и методическая подготовка учителя. Педагог должен владеть современными инструментами программирования, уметь адаптировать материалы под возрастные особенности, грамотно управлять цифровой средой и быть готовым к роли наставника, а не только транслятора знаний. В связи с этим возрастает значение повышения квалификации учителей информатики, обмена лучшими практиками и создания методических сообществ.

В заключение следует отметить, что преподавание программирования в 5–7 классах – это не только обучение навыкам работы с кодом, но и формирование у детей гибких, универсальных компетенций XXI века. Умение логически мыслить, структурировать информацию, решать задачи, сотрудничать и учиться через практику – все это закладывается в процессе освоения основ программирования. Методика преподавания должна быть гибкой, доступной, интересной и постоянно обновляемой в соответствии с вызовами времени и потребностями учеников. Только при таких условиях возможно сформировать у школьников устойчивый интерес к ИТ и заложить прочную основу для дальнейшего профессионального роста в области цифровых технологий.

Список литературы

- [1] Босова, Л.Л., Босова, А.Ю. Методика преподавания информатики. – М.: БИНОМ, 2021.
- [2] Поляков, К.Ю., Еремин, Е.А. Основы алгоритмизации и программирования: учебник для общеобразовательных школ. – М.: Просвещение, 2020.

- [3] Grover, S., Pea, R. Computational Thinking in K–12: A Review of the State of the Field. Educational Researcher, 2013.
- [4] Brennan, K., Resnick, M. New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 Annual Meeting of the American Educational Research Association, 2012.
- [5] Wing, J. Computational Thinking. Communications of the ACM, 2006.
- [6] Code.org. CS Fundamentals Curriculum Guide. <https://curriculum.code.org>
- [7] Papert, S. Mindstorms: Children, Computers, and Powerful Ideas. — New York: Basic Books, 1980.

ЗЕРТТЕУШІЛІК ОҚЫТУ ЖӘНЕ ТОПТЫҚ ЖОБАЛАР МЕН ПРЕЗЕНТАЦИЯЛАР ӘДІСТЕРІНІҢ МАТЕМАТИКАНЫ ОҚЫТУДАҒЫ ТИІМДІЛІГІ

Искакова Г.Ш.¹, Шаяхметова Б.К.², Ахметбекова Н.С.³

^{1,2,3} Академик Е.А.Бөкетов атындағы ҚарУ, Қарағанды, Қазақстан

¹E-mail: Iskakova.1975@mail.ru

²E-mail: kazakhzavod@mail.ru

³E-mail: nightxnara@mail.ru

Қазақстан Республикасының білім беру саясаты мен нормативтік құжаттарын, стратегиялық жоспарларын, заңдарын және білім беру әдістемесіндегі заманауи үрдістерді ескерсек ҚР 2007 ж. "Білім туралы" Заңының, жаңартылған нұсқаларымен 6-бап, 2-тармағында: "Білім беру жүйесінің басты міндеті – жеке тұлғаның шығармашылық, рухани және зияткерлік мүмкіндігін дамыту" деп атап өтілген. [1]-[3]

Жоғары оқу орнында математиканы тиімді оқыту үшін қолданылатын әдіс-тәсілдер мектептегі әдістерден өзгеше болуы мүмкін, себебі студенттердің танымдық қабілеті жоғары, және теорияны терең меңгеру талап етіледі. Кеңінен қолданылатын және тиімді саналатын әдістерге дәріс пен практиканы ұштастыра отырып оқыту, яғни интеграцияланған оқытуды айтуға болады. Бұл әдістің негізгі мақсаты теорияны түсіндіріп қана қоймай, оны бірден қолдануға үйрету болып табылады. Теориялық бөлімнен кейін есептер шығару, моделдеу, дәлелдеулер жүргізуді қажет етеді. Зерттеушілік әдісін студенттердің ғылыми жобалар, курстық, дипломдық жұмыстар дайындауында қолдануға болады. Бұл әдістің артықшылығы академиялық сауаттылық пен өз бетінше ізденуді дамытады. Мәселелік оқыту - студентке нақты бір математикалық мәселе немесе есеп беріледі, оны шешу үшін өздігінен теорияны меңгеріп, талдау жасайды. Бұл әдісті әсіресе, дифференциалдық теңдеулер, ықтималдықтар теориясы, математикалық физика пәндерінде қолданған тиімді. Кейс-стади әдісі теорияны нақты өмірмен байланыстырады. Интерактивті әдістер Kahoot, Desmos, GeoGebra, Mathematica, Maple, Moodle платформаларын қолдану арқылы студенттердің сабаққа қызығушылықты арттырады, визуализация мен анимация көмектеседі. Топтық жобалар мен презентациялар студенттерге белгілі бір тақырып беріледі, олар оны зерттеп, сынып алдында қорғайды. Бұл коммуникативтік дағдылар мен логикалық ойлауды дамытады. Флипид класс әдісі - теорияны үйде бейнематериал немесе конспект