

ОСОБЛИВОСТІ НАВЧАННЯ ВЕБ-ПРОГРАМУВАННЯ МОВОЮ JAVASCRIPT СТУДЕНТІВ-МАТЕМАТИКІВ

Статтю присвячено розкриттю особливостей змісту і структури модуля "Веб-програмування мовою JavaScript" для студентів-математиків. У статті розкрито методичні особливості навчання веб-програмування студентів-математиків, висвітлено основні проблеми, які виникають у процесі навчання веб-програмування майбутніх учителів математики та інформатики. Автор пропонує свою концепцію викладання основ веб-програмування мовою JavaScript в курсі "Спеціальний лабораторний практикум з інформатики" під час навчання за освітньо-кваліфікаційним рівнем "Спеціаліст".

Ключові слова: веб-програмування, студенти-математики, мова JavaScript.

Постановка проблеми у загальному вигляді. Бурхливий розвиток інформаційно-комунікаційних технологій, зростання ролі мережі Інтернет у навчанні студентів неминуче впливають і на зміст сучасної інформатичної освіти. Майбутній учитель-інформатики має бути людиною, компетентною у більшості галузей ІКТ, у тому числі і веб-програмуванні. Поява курсу "Основи веб-дизайну" у структурі шкільної інформатичної освіти зумовлює необхідність вивчення основ програмування мовою JavaScript саме студентам-математикам, які мають другу спеціальність "Інформатика". З іншого боку, кількість годин з інформативних дисциплін постійно зменшується. Курс "Веб-програмування" або "Основи веб-програмування" не належить до нормативних дисциплін спеціальності "Математика", а отже, в багатьох педагогічних університетах просто не викладається. Виникає певне протиріччя між наявністю об'єктивних передумов до вивчення курсу веб-програмування в педагогічному університеті і відсутністю окремої дисципліни, в якому він буде вивчатися.

Аналіз останніх праць. Питання вивчення програмування у вищих педагогічних навчальних закладах висвітлюються у працях М. І. Жалдака, Н. В. Морзе, С. О. Семерікова та ін. Проблеми вивчення основ веб-програмування в загальноосвітній школі і педагогічному ВНЗ присвячені дослідження І. С. Іваськіва, Ю. С. Рамського, О. Ю. Ніколаєнка та інших учених [1; 2]. Розглядаючи методичні особливості вивчення даного курсу, специфіку викладання веб-програмування для різних попередніх мов програмування, уточнюючи форми і методи навчання веб-програмування, науковцями недостатньо розкрито один із важливих аспектів даної проблеми, а саме – зміст курсу "Основи програмування мовою JavaScript".

Метою даної статті є обґрунтування змісту курсу веб-програмування мовою JavaScript для студентів-математиків, другою спеціальністю яких є інформатика.

Виклад основного матеріалу дослідження. Одним із шляхів вирішення проблеми вивчення основ веб-програмування у вищому педагогічному навчальному закладі, на нашу думку, є введення елементів веб-програмування у зміст дисциплін інформатичної спрямованості. Крім базового курсу інформатики, студенти-математики (які мають спеціалізацію "Інформатика") вивчають інші дисципліни інформатичної спрямованості: бази даних, вибрані питання інформатики, спеціальний лабораторний практикум з інформатики, адміністрування інформаційних систем, проектування інформаційних систем, практикум із програмування.

Особливої уваги заслуговує дисципліна "Спеціальний лабораторний практикум з інформатики", на вивчення якої навчальним планом спеціальності "Математика" відведено 90 годин, з них 44 години – лабораторні заняття. Половини цієї кількості, на нашу думку, цілком достатньо для формування ключових компетентностей учителя інформатики у галузі веб-програмування мовою JavaScript.

Уточнимо спочатку вимоги до змісту навчального матеріалу. На нашу думку, до навчального матеріалу з веб-програмування мовою JavaScript висуваються такі вимоги:

1. обсяг навчального матеріалу повинен охоплювати всі ті теми, без вивчення яких досить важко зрозуміти особливості даної мови програмування;
 2. навчальний матеріал має бути чітко структурованим, логічно і послідовно викладеним;
 3. зміст навчального матеріалу повинен надавати уявлення про основні об'єкти, властивості і методи JavaScript;
 4. навчальний матеріал повинен бути достатньо деталізованим для забезпечення міцності знань, умінь та навичок студентів;
 5. навчальний матеріал має бути викладеним на доступному для студентів рівні;
 6. наявність задач, розв'язаних із помилками і розв'язаних частково;
- Добираючи навчальний матеріал, ми керувалися наступними принципами:
– від простого до складного;

- наступності і послідовності;
- посиленості;
- зв'язку теорії з практикою;
- індивідуалізації та диференціації навчання.

Розглянемо детальніше, як реалізуються вказані дидактичні принципи у процесі вивчення веб-програмування мовою JavaScript. Під час побудови змісту модуля "Веб-програмування мовою JavaScript" було враховано принцип "від простого до складного". Спочатку розглядається введення-виведення даних як необхідний компонент програмування мовою JavaScript, а на наступних заняттях навчальний матеріал поступово ускладнюється. У той же час навчальний матеріал кожної наступної теми базується на попередніх темах.

На основі аналізу змісту і особливостей програмування мовою JavaScript було зроблено висновок про те, що він має містити 11 лабораторних робіт за такою тематикою:

1. Введення і виведення даних в JavaScript.
2. Розгалуження.
3. Циклічні алгоритми.
4. Масиви.
5. Операції із рядковими змінними.
6. Створення об'єктів користувача.
7. Операції із вікнами.
8. Використання динамічних стилів.
9. Керування документом і рядком статусу.
10. Програмування форм.
11. Створення меню.

Розглянемо особливості навчання студентів-математиків програмування мовою JavaScript.

Як відмічають науковці [1; 2], методика навчання веб-програмування залежить від того, які мови програмування студенти вивчали перед цим:

- одну з процедурних мов програмування (Паскаль, С);
- одну з об'єктно-орієнтованих мов програмування (ObjectPascal, C++, Java).

У першому випадку студентів варто ознайомити з об'єктно-орієнтованою парадигмою програмування, відповідно до якої програма подається у вигляді сукупності об'єктів, кожен із яких є реалізацією деякого класу, а класи утворюють ієрархію за принципами наслідування. Центральним поняттям об'єктно-орієнтованого програмування є об'єкт. Програмний об'єкт володіє деяким зовнішнім виглядом або властивостями, що виражаються у значеннях його змінних, і методами, що задаються у вигляді його процедур. Властивості та методи не існують відокремлено один від одного, а об'єднані разом, утворюючи об'єкт з новою властивістю. Методи "оточують" властивості об'єкта, не дозволяючи напряму звертатися до них або змінювати їх значення. Властивості інкапсульовані в об'єкт і доступ до них здійснюється тільки шляхом методів, які надаються об'єктом. При викладанні матеріалу недоцільно заглиблюватися в механізми успадкування і поліморфізму, які реалізуються об'єктно-орієнтованими мовами програмування. JavaScript реалізує просту об'єктно-орієнтовану модель і для розуміння функціонування і створення сценаріїв достатньо наведеної інформації про програмні об'єкти [1: 179-182].

У другому випадку варто відмітити, що JavaScript базується на спрощеній об'єктно-орієнтованій парадигмі програмування, тому студентам доцільно ознайомитися з особливостями синтаксису цієї мови і можна приступати до реалізації характерних для мови сценаріїв JavaScript програм [1: 183].

Як показує практика, студенти-математики до 5 курсу вивчають і структурні, і об'єктно-орієнтовані мови програмування. Застосувати можна кожен із указаних шляхів навчання програмування. Однак спочатку доцільно уточнити: об'єктно-орієнтовані мови програмування зазвичай вивчаються за допомогою візуальних середовищ програмування. Мова сценаріїв JavaScript не має візуального середовища розробки програм, тому у студентів виникають значні труднощі при переході до створення програм мовою JavaScript. Тому доцільним, на нашу думку, є використання другого і першого варіанту (коли студенти вивчали структурну мову програмування).

Для тих студентів, що раніше вивчали структурні мови програмування, доцільно починати з написання функцій у заголовку документа [1: 184]. Далі варто ознайомити студентів з основними об'єктами JavaScript. Таку саму схему побудови документа варто використовувати і далі: ознайомлення студентів з першою програмою на мові JavaScript, з викликом програми безпосередньо з гіперпосилання, з розміщенням програми у контейнер <SCRIPT> тіла документа, з викликом обробника події Click елемента форми. На наступних етапах завдання ускладнюються [1: 185].

Ускладнення відбувається так: більшість лабораторних робіт містять дві задачі. Розв'язання першої задачі відбувається з використанням вікон prompt, alert, розв'язання другої задачі – з використанням елементів введення форм. На наочному прикладі студенти мають усвідомити відмінності між зчитуванням даних із спливаючих вікон і зчитуванням даних з елементів форми. Аналогічними є

відмінності у виведенні даних у вікно типу alert і текстові поля форм. Ці вміння удосконалюються у процесі кожної лабораторної роботи.

Визначаючи особливості методики навчання веб-програмування студентів, зауважимо, що JavaScript базується на об'єктно-орієнтованому підході, який передбачає оперування не тільки зі змінними, але й з властивостями та методами об'єктів. Як показує практика, саме поняття об'єктно-орієнтованого підходу викликає певні труднощі для розуміння студентами сутності понять "об'єкт", "властивість", "метод", "інкапсуляція" та ін. У процесі вивчення мов структурного програмування (Паскаль, С, С++) основні поняття об'єктно-орієнтованого програмування, здебільшого, залишаються "за кадром". Під час вивчення мови Object Pascal і середовища програмування Delphi студент працює у візуальному середовищі програмування, в якому звертання до об'єктів та їх властивостей відбувається напівавтоматично. Це значно прискорює процес розробки програми, але часто призводить до формування поверхневих уявлень про особливості об'єктно-орієнтованого програмування. Під час роботи над сценарієм на мові JavaScript студентам доводиться писати код програми вручну, набираючи відповідні властивості і методи, їх значення з клавіатури, а для цього їм необхідно чітко знати назви основних об'єктів і їх властивостей, типів спливаючих вікон і т. д. Це призводить до частих помилок і ускладнює вивчення мови сценаріїв JavaScript.

Суттєвим недоліком курсу "Спеціальний лабораторний практикум з інформатики" є відсутність лекцій. Лекції є необхідним компонентом навчального процесу. Через їх відсутність у навчальному плані теоретичний матеріал доводиться пояснювати на лабораторних заняттях, що призводить до скорочення кількості часу, який припадає на виконання практичного завдання. Доопрацьовувати задачі студенти мають вдома, що за відсутності поряд викладача зводить їх зусилля нанівець. Не в силах знайти помилку в коді, вони не виконують одне завдання, що призводить до неможливості виконувати наступне.

Для того, щоб запобігти даним недолікам, доцільно проконтролювати виконання студентами перших лабораторних робіт. Окрім того, спрощує роботу студентів з розробки HTML-коду застосування WEB-редактора, в якому всі ці дії автоматизовані [3: 19]. Більшість WEB-редакторів мають підсвітку коду, що, в свою чергу, спрощує написання коду мовою JavaScript. Водночас більшість таких редакторів мають режим попереднього перегляду, що теж спрощує роботу над програмою.

Окрім спеціалізованого програмного забезпечення, можливо використовувати консоль налагоджування, яка присутня в таких браузерах, як Mozilla Firefox останніх версій.

Практика показує, що найбільш ефективним засобом формування умінь студентів-математиків створювати сценарії мовою JavaScript є задача. Її переваги перед проектом полягають у тому, що задача формулюється конкретно і вимагає конкретного розв'язку. Задача може бути розв'язана правильно або неправильно. При задачному підході спрощується контроль за навчальними досягненнями студентів, водночас економиться аудиторний час, якого зазвичай не вистачає.

На основі аналізу педагогічної практики варто відмітити, що задачний підхід до вивчення програмування мовою сценаріїв JavaScript має низку переваг перед проектним підходом:

1. чіткий поділ навчального матеріалу на порції (задачі);
2. полегшення фіксації навчальних досягнень студентів;
3. економія аудиторного часу.

Для успішного розв'язування задач із програмування у студентів повинно бути розвинуте логічне та алгоритмічне мислення, яке формується у процесі вивчення математичних дисциплін та програмування. На п'ятому курсі у переважній більшості студентів-математиків таке мислення сформоване. Студенти-п'ятикурсники зазвичай мають чітке уявлення про основні алгоритмічні конструкції (слідування, розгалуження, цикл), тому логіка побудови програми зазвичай не є проблемою. У процесі вивчення програмування мовами Паскаль або С++ студенти засвоїли основні алгоритмічні конструкції, тому викладач має тільки нагадати їх. Варто звернути увагу студентів на особливості реалізації вказаних алгоритмічних конструкцій мовою JavaScript, тобто їх синтаксис, який значно відрізняється від синтаксису мови Паскаль. Якщо студенти у минулому вивчали мову С++, то варто звернути їх увагу на відмінності у структурі програми на JavaScript і С++ (у програмі на JavaScript відсутня декларативна частина, оголошення змінних відбувається під час їх ініціалізації, тип змінних не вказується явно, введення і виведення здійснюється у вікна prompt, confirm або елементи форм). У випадку, коли студенти раніше вивчали мову С або мови на її основі, майже не виникає проблем у засвоєнні студентами синтаксису мови сценаріїв JavaScript.

На основі аналізу ефективності роботи студентів було зроблено висновок про те, що оптимальна кількість задач для виконання на лабораторному занятті – дві. Більша кількість задач призводить до того, що студенти часто не встигають виконувати всі задачі, менша – у них з'являється зайвий час, тобто робочий час лабораторного заняття використовується неефективно.

Для дотримання принципу диференціації та індивідуалізації навчання кожен інструкційний лист до лабораторної роботи містить 20 варіантів завдань. Кожен варіант налічує дві задачі, виконувати які студент має самостійно.

Формування у студентів поняття об'єкта, властивостей і методів об'єкта, здебільшого, викликає більше труднощів. Але за умови використання наочних прикладів ці труднощі можуть бути зменшені.

Варто звернути увагу студентів на особливості ієрархії об'єктів у JavaScript: найвищий рівень мають об'єкти класу window, далі слідує об'єкти класу document, які, в свою чергу, включають об'єкти нижчого рівня (форми та елементи форм). Кожен клас об'єктів має свої властивості і свої методи.

Тобто, існують об'єктивні засади для успішного навчання створення сценаріїв мовою JavaScript в умовах вивчення дисципліни "Спеціальний лабораторний практикум з інформатики" студентами-математиками.

Вивчення основ програмування сценаріїв на JavaScript можна здійснювати також за допомогою методу проектів. У такому випадку студент працює над певним проектом, продовжуючи поетапно розробку одного програмного продукту протягом усього модулю. Його переваги перед задачним підходом:

1. проектний підхід сприяє формуванню у студентів умінь планувати свої дії, розробляти і реалізувати логіку проекту, формулювати мету і завдання проекту і т. д.;

2. у процесі здійснення проекту студенти виконують основні етапи дослідження, що сприяє формуванню у них дослідницьких умінь;

3. важливе значення має практична спрямованість проекту.

На практиці доведено, що задачний підхід у навчанні студентів-математиків створення сценаріїв мовою JavaScript має ряд переваг над проектним: розв'язування задач викликає значно менше труднощів, ніж виконання проекту, висуває більш конкретні вимоги до розв'язання. Одна задача – один розв'язок.

Висновки з проведеного дослідження і перспективи подальших розвідок у даному напрямі. На основі вищесказаного можна зробити висновок про те, що навчання основ програмування мовою JavaScript доцільно для випадку, коли студенти вивчали раніше одну зі структурних мов програмування. Найбільш ефективним визнано задачний підхід, тому структуру модуля "Розробка сценаріїв мовою JavaScript" доцільно будувати відповідно до цього підходу.

Перспективою подальших розвідок у даному напрямі є обґрунтування компонентів системи навчання веб-програмування мовою JavaScript.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Рамський Ю. С. Вивчення Web-програмування в школі : [навч. посібник] / [Рамський Ю. С., Іваськів І. С., Ніколаєнко О. Ю.]. – Тернопіль : Навчальна книга – Богдан, 2004. – 200 с.
2. Рамський Ю. С. Методика навчання основ Web-програмування в загальноосвітній школі / Ю. С. Рамський, І. С. Іваськів // Комп'ютер у школі та сім'ї. – 2000. – №1. – С. 7–10 ; 2000. – № 2. – С. 18–23 ; 2000. – № 3. – С. 18–21 ; 2000. – № 4. – С. 16–21 ; 2001. – № 1. – С. 18–21.
3. Базурін В. М. Порівняльний аналіз WEB-редакторів / В. М. Базурін // Комп'ютер у школі та сім'ї. – 2007. – № 6. – С. 19–23.

REFERENCES (TRANSLATED & TRANSLITERATED)

1. Ramskyu Yu. S. Vuvchennya Veb-programuvannya v shkoli [Web-Programming Learning at School] : [navch. Posibnyk] / [Ramskyu Yu. S., Ivaskiv I. S., Nikolaenko O. Yu.]. – Ternopil : Navchalna knyga-Bogdan, 2004. – 200 s.
2. Ramskyu Yu. S. Metodyka navchannya osnov web-programuvannya v zagalnoosvitniy shkoli [Methodology of Teaching of Web-Programming Fundamentals] / Yu. S. Ramskyu, I. S. Ivaskiv // Komputer u shkoli ta simyi [Computer at School and Family]. – 2000. – № 1. – S. 7–10 ; 2000. – № 2. – S. 18–23 ; 2000. – № 3. – S. 18–21 ; 2000. – № 4. – S. 16–21 ; 2001. – № 1. – S. 18–21.
3. Bazurin V. M. Porivnyalnyy analiz WEB-redaktoriv [Comparative Analysis of WEB-Processors] / V. M. Bazurin // Komputer u shkoli ta simyi [Computer at School and Family]. – 2007. – № 6. – S. 19–23.

Матеріал надійшов до редакції 23.01. 2014 р.

Базурин В. Н. Особенности обучения веб-программированию на языке JavaScript студентов-математиков.

Статья посвящена раскрытию особенностей содержания и структуры модуля "Веб-программирование на языке JavaScript" для студентов-математиков. В статье раскрыты методические особенности обучения веб-программирования студентов-математиков, освещены основные проблемы, которые возникают в процессе обучения веб-программированию будущих учителей математики и информатики. Автор предлагает свою концепцию преподавания основ веб-программирования языком JavaScript в курсе "Специальный лабораторный практикум по информатике" во время обучения за образовательно-квалификационным уровнем "Специалист".

Ключевые слова: веб-программирование, студенты-математики, язык JavaScript.

Bazurin V. M. Features of Teaching Web Programming Students-Mathematicians in JavaScript Language.

The article is devoted to opening the content peculiarities and the structure of the module "Web Programming in the JavaScript Language" for students-mathematicians. The methodical features of teaching Web programming of students-mathematicians are exposed in the article, basic problems which arise up in the process of teaching Web programming of future teachers of mathematics and informatics are highlighted. An author offers his conception of learning web-programming bases on the JavaScript language in the course "Special Laboratory Practical Work on Informatics" during studies after the educational-qualifying level "Specialist". The author formulates the requirements to the laboratory of Web programming. The author tells about didactic principles, which are used during the learning Web programming on the JavaScript language: 1) from simple to complex; 2) continuity and consistency; 3) affordability; 4) the connection between theory and practice; 5) individualization and differentiation. V. Bazurin describes main ways of realization of these principles during the studying the Web programming. The author describes the structure and contents of the laboratory practicum, which used the author's model of learning web-programming on JavaScript language. During planning the content of laboratory practice the author used the task (problem) approach, because this approach must be the most effective than other approaches. The task (the problem) is the main component of the laboratory practice. Another approach is the project approach. During the lessons students work on the project. The author concludes that the most effective approach in the study of Web programming is the task approach

Key words: *web-programming, students-mathematicians, JavaScript language.*