професійна підготовка фахівців з фізики". – К.: НПУ ім. М.П.Драгоманова, 1998. – Частина І. – С.15-19.

The flow diagram of innovative didactics process of professionally-methodical preparation of teacher of physics is offered

Key words: elements of knowledge's, structure, levels of knowledge's.

Отримано: 18.05.2006.

УДК 37.026.9+681.3+37.01:007+518.5

О.П. Лінник[1], Н.В. Моісеєнко[2], В.М. Євтєєв[2], І.О. Теплицький[2], С.О. Семеріков[2]

[1]Інститут повітряного транспорту Національного авіаційного університету, м. Кривий Ріг
[2]Криворізький державний педагогічний університет

ОБ'ЄКТНО-ОРІЄНТОВАНЕ МОДЕЛЮВАННЯ У ПІДГОТОВЦІ МАЙБУТНІХ УЧИТЕЛІВ ФІЗИКИ

Стаття присвячена досвіду впровадження технології комп'ютерного моделювання у курсі «Об'єктно-орієнтоване програмування» для студентів фізико-математичних факультетів педагогічних університетів.

Ключові слова: комп'ютерне моделювання, Python, об'єктно-орієнтоване програмування, бібліотека VPython, методична система навчання.

Постановка проблеми. Природний зв'язок об'єктно-орієнтованого програмування (ООП) та моделювання був відображений вже у першій мові ООП – Simula 67, сама назва якої походить від слова *simulation* – моделювання. Останні роки автори регулярно публікують у педагогічних виданнях матеріали за цією тематикою ([2-4] та інші). Зокрема, в [3] йшлося про впровадження наскрізної програми систематичного навчання моделювання на всіх спеціальностях педагогічних ВНЗ, що мають спеціалізацію "Основи інформатики". Пропонований нижче матеріал є розвитком [3;7].

Метою пропонованого матеріалу є обговорення структури та змісту курсу ООП для майбутніх вчителів фізики.

Основна частина. Традиційно в комплексі навчальних дисциплін, що вивчаються за спеціалізацією "Основи інформатики", виділяють дві складові – теоретичну та практичну (технологічну). Якщо зміст першої є усталеним, то в межах другої спостерігається надвисокий рівень варіативності (інколи навіть в межах одного ВНЗ), що суттєво зменшує мобільність студентів та суперечить вимогам Болонської декларації до уніфікації підготовки спеціалістів з метою створення єдиного освітнього простору [2].

Однією з проблем, що постають перед викладачами інформатики, є необхідність швидкого реагування на зміни у цій галузі з подальшою адаптацією програмно-методичного забезпечення відповідних курсів. При цьому часто не розрізняються суттєві зміни, що вимагають модифікації окремих положень теоретичної складової курсу, та несуттєві, пов'язані переважно з випуском чергової версії одного із програмних засобів, застосовуваних для підтримки курсу інформатики [4].

Головним положенням розроблюваної нами методичної системи є те, що інформатика у вищій школі – фундаментальна дисципліна, ядро якої є усталеним та інваріантним для всіх напрямків підготовки. Саме цілі та зміст навчання є тими системоутворюючими компонентами, що визначають методи, форми та засоби навчання (зокрема, програмні). Такий підхід спрямований на подолання негативної тенденції штучного прив'язування змісту курсу інформатики до використовуваного програмного забезпечення (та, відповідно, неадекватного збільшення ролі технологічної складової).

З метою зменшення залежності від використовуваної операційної системи (ОС) доцільно застосовувати кросплатформенні інтерпретатори мов програмування. В курсі ООП для студентів-фізиків ми зупинилися на Python – простій та потужній ООП-мові, що ідеально підходить для швидкої побудови прототипів (т.з. «одноразового програмування») [1]. Інтерпретатор Python вільно поширюється і доступний на різноманітних ОС.

Інтерпретатор Python легко розширюється новими функціями і типами даних, реалізованих іншими мовами. Одним з модулів, що розширюють можливості роботи Python з 3D-графікою засобами бібліотеки OpenGL, є VPython. При використання цього модуля властива Python можливість швидкого прототипування програм дає змогу, не відволікаючи на деталі реалізації інтерфейсу користувача, зосередитися на сутності модельованого явища.

У навчальному посібнику [1] описаний процес побудови 3D-моделей ряду фізичних процесів: поступального рівномірного та рівноприскореного руху; руху тіла, кинутого під кутом до горизонту; руху центра мас системи двох тіл; обертання тіл різної форми. Малий спектр розглянутих моделей визначається орієнтацією автора посібника на курс загальної фізики технічних вузів, у той час як на фізичних спеціальностях класичних та педагогічних університетів він повинен бути істотно розширений.

Лекційний курс ООП включає в себе наступні теми:

1. Основні концепції ООП *2 год.*
2. Основи синтаксису Python *2 год.*
3. Об'єкти модуля VPython *3 год.*
4. Моделі атомів H, He, Li *2 год.*
5. Більярдна модель ідеального газу *2 год.*
6. Молекулярна динаміка *3 год.*
7. Динаміка Сонячної системи *4 год.*
8. Моделювання спіральних галактик *2 год.*
9. Застосування модуля VPython в програмах на C++ *2 год.*

Вивчення ООП починається з огляду його основних складових – об'єктно-орієнтованого аналізу та проектування, які, у відповідності до [8], є основами об'єктно-орієнтованого моделювання.

22-годинний лекційний курс супроводжується комплексом лабораторних робіт, в яких вимагається не стільки створити готовий програмний продукт, використовуваний в якості лекційної демонстрації, скільки виконати деталізований опис процесу створення моделі з розкриттям фізичного змісту модельованого явища, побудувати робочий прототип програми та провести на ньому ряд експериментів.

При вивченні сьомої теми в якості допоміжного програмного забезпечення ми застосовуємо VPNBody – колекцію модулів мовою програмування Python, створена Родні Даннінгом [9] та локалізована нами. Назва VPNBody є поєднанням Візуального Python (VP) та N-частковий. Комплекс призначений для моделювання систем під дією сили тяжіння, що складаються з гравітаційно домінуючого об'єкта (зірка) і декількох менших об'єктів (планети).

Для використання VPNBody необхідний інтерпретатор Python і модуль VPython. Якщо вони встановлені, запуск системи здійснюється вибором файлу *vpnb.py* (*рис. 1*).

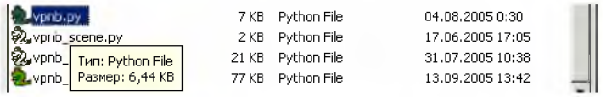


Рис. 1. *Запуск системи*

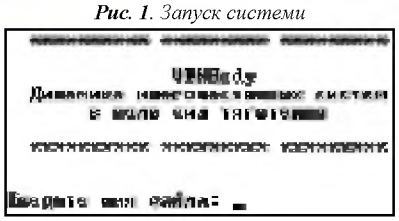


Рис. 2. *Початковий екран системи*

Увівши у запрошенні `inner_planets.txt`, одержимо наступне вікно (*рис. 3*):

Рис. 3. *Моделювання руху внутрішніх планет*

Модуль VPython дає наступні можливості керування вікном анімації:

1. Призупинити і відновити анімацію – натискання лівої кнопки миші.
2. Поворот сцени (системи координат) – переміщення миші при натиснутій правій кнопці.
3. Зміна розміру сцени – утримуючи натиснутими ліву і праву кнопки миші, тягти мишу до верху екрану.
4. Завершення моделювання – натиснути клавішу Escape.

При масштабуванні та обертанні сцени можна побачити тонкі лінії, що йдуть від Сонця – це необов'язкові координатні вісі (*рис. 4*).

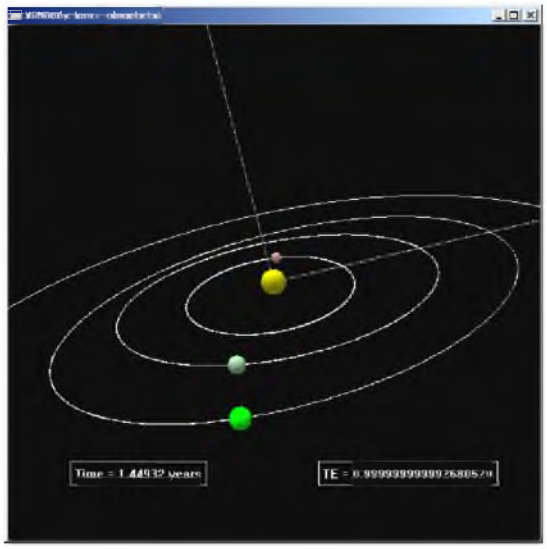


Рис. 4. *Моделювання внутрішніх планет після обертання і зміни розмірів сцени*

На *рис. 4* Сонце розташоване в центрі поля зору. Внизу: ліворуч – модельний час (у роках), праворуч – повна механічна енергія системи в одиницях початкової повної механічної енергії. Чим більш це число відрізняється від 1, тим менш точні результати моделювання.

На моделювання динаміки Сонячної системи у пропонованому курсі відведено найбільший час не через складність матеріалу (адже дана задача може бути розв'язана і засобами, доступними допитливому школяреві [5-7]), а внаслідок багатства та наочності моделей руху. Люди завжди були заворожені небом – колективне людське прагнення зрозуміти рух планет було тією рушійною силою, що сформувала сучасну фізику й астрономію. Сьогодні інтерес до динаміки Сонячної системи залишається сильним і серед професійних учених, і в колі аматорів. Наприклад, недавнє відкриття планет навколо інших зірок підняло важливе питання про те, чи могла б землеподібна планета рухатися по орбіті іншої зірки в межах критичної зони, в якій вода перебуває в рідкому стані на поверхні планети. VPNBody забезпечує зручний та ефективний вступ до цієї області наукових досліджень.

Як зазначалося вище, VPNBody призначений для моделювання систем, що складаються з гравітаційно домінантного об'єкта, по орбіті якого рухаються декількома менших об'єктів. Очевидний приклад такої системи – батьківська зірка і дочірні планети, астероїди, комети. Інший приклад – система планета-місяць.

У Сонячній системі зірка і планети взаємодіють за законом всесвітнього тяжіння:

$$F_{ij} = G\frac{m_i m_j}{r_{ij}^2}. \qquad (*)$$

В усіх моделях нас цікавить еволюція системи у часі. Взагалі, існує два типи інформації, що ми намагаємося добути з моделі:

1) нас можуть цікавити точні відносні положення планет у даний момент часу,
2) ми можемо бути зацікавлені стабільністю і довгостроковою тривалою еволюцією орбіт – чи може, приміром, землеподібна планета ε-Андромеди залишатися в межах населеної зони настільки довго, щоб там виникло життя?

Тип інформації, що ми його хочемо одержати, визначає методи, які використовуються для дослідження системи.

Еволюція системи визначається обчисленням траєкторії для кожної планети за рівнянням (*). Є багато способів зробити це, проте всі вони можуть бути поділені на дві великі категорії. У першій категорії – алгоритми Рунге-Кутта (РК). Алгоритм РК високого порядку може використовуватися для відстеження руху планет з майже машинною точністю (принаймні, певний час). Помилка, пов'язана з алгоритмами РК, не обмежена, і тому ці алгоритми не можуть звичайно використовуватися для довгострокового моделювання системи. В другій категорії – симплектичні алгоритми. На відміну від РК-методів, симплектичні алгоритми не зберігають повну механічну енергію системи, але енергетична помилка обмежена. Таким чином, симплектичний алгоритм може використовуватися для еволюції системи протягом майже необмеженого часу (можливі обмеження зумовлюються помилками округлення). Взагалі, РК-методи використовуються, коли ми хочемо знати відносні положення планет на деякий момент часу в найближчому майбутньому, а симплектичні методи використовуються, коли ми хочемо знати, чи буде орбіта даної планети стійкою протягом кількох мільйонів років.

Для одержання траєкторії кожного об'єкта системи з рівняння (*) ми повинні знати *початкові умови* системи. Існує кілька способів їхнього задання. Перший – розташування планет на одній з осей («парад планет») і задання для кожної з них швидкості [6-7]. Але більш зручно застосовувати *орбітальні елементи*. Орбітальні елементи планети описують її розміри (одне число), форму орбіти (одне число), орієнтацію (три числа) орбіти в просторі, і поточне положення об'єкта на орбіті.

Перший орбітальний елемент – велика піввісь a еліптичної орбіти планети відносно зірки.

Другий орбітальний елемент – ексцентриситет e планетної орбіти. Ексцентриситет описує форму орбіти ($e = 0$ визначає ідеальне коло, граничне значення ексцентриситету $e = 1$).

Третій орбітальний елемент – нахил i, що вимірюється відносно заданої опорної площини. У Сонячній системі опорною вважається площина орбіти Землі і нахили решти планетних орбіт вимірюються відносно неї. *Рис. 5* показує орбіти Меркурія і Землі. Нахил орбіти Меркурія відносної земної – i орбіти Меркурія.

Рис. 5. *Нахил орбіти Меркурія відносно земної $i = 7°$*

В анімаціях VPNBody опорна площина – завжди початкова площина сцени (площина комп'ютерного монітора).

Четвертий орбітальний елемент – довгота точки сходження Ω. Точка сходження (висхідний вузол) – точка в опорній площині, у якій орбіта планети при русі нагору перетинає опорну площину. Довгота точки сходження – кут, вимірюваний в опорній площині проти годинникової стрілки від осі *x* до точки сходження. В Сонячній системі вісь *x* вказує на точку весняного рівнодення – позицію в небі, що її займає Сонце у перший день весни.

П'ятий орбітальний елемент – напрямок на перицентр ω. Перицентр – точка зближення на найкоротшу відстань між планетою і зіркою (відповідно точка максимального віддалення називається *апоцентром*). Напрямок на перицентр – кут ω, що вимірюється проти годинникової стрілки в площині орбіти планети від точки сходження до перицентра.

Шостий орбітальний елемент – середня аномалія *M*. Середня аномалія – кут, вимірюваний проти годинникової стрілки в площині орбіти планети від перицентра до позиції *середньої планети*. Реальні планети рухаються по орбіті зі змінною швидкістю; через збереження кутового моменту ця швидкість найбільша в позиції перицентра і найменша в позиції апоцентра. Середня планета рухається по орбіті з *постійною* швидкістю так, що середня планета і реальна планета збігаються в позиціях перицентра й апоцентра.

Приклади моделей (http://semerikov.googlepages.com/vpnbody.rar).

asteroids.txt – включає Сонце, Марс, Юпітер, Сатурн і 15 астероїдів з випадковими орбітальними елементами, розташовані між орбітами Марса і Юпітера (*рис. 6*).

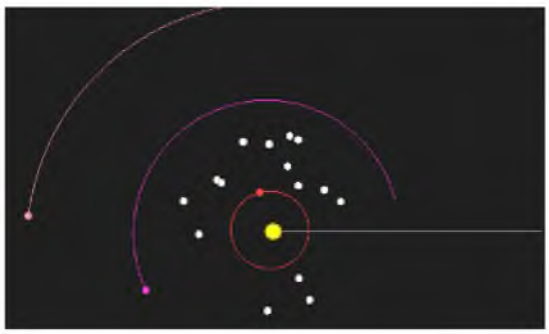
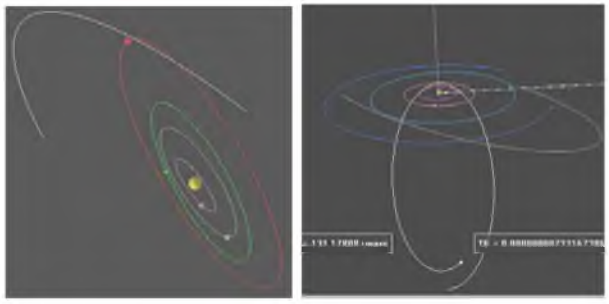


Рис. 6

comet_inner.txt – включає Сонце, Меркурій, Венеру, Землю, Марс і малу комету, що рухається по орбіті з великим ексцентриситетом (*рис. 7а*).

comet_outer.txt – включає Сонце, Юпітер, Сатурн, Уран, Нептун, Плутон і малу комету (*рис. 7б*).



а) *б)*

Рис. 7

compare_a.txt – включає зірку і три планети з ідентичними орбітальними елементами, за винятком півосей. Анімація ефективно показує відношення між півосями й орбітальним періодом (*рис. 8*).

compare_ap.txt – включає зірку і три планети з ідентичними орбітальними елементами, за винятком напрямку на перицентр (*рис. 9*).

compare_e.txt – включає зірку і три планети з ідентичними орбітальними елементами, за винятком ексцентриситету (*рис. 10*). Кожна планета починає рух з напрямку на перицентр, повертаючись за однаковий час. Це модель показує, що між ексцентриситетом і орбітальним періодом немає зв'язку.

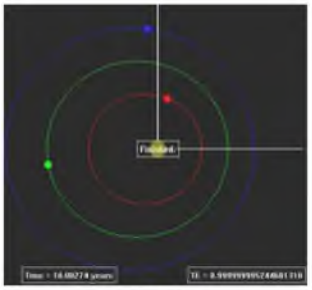


Рис. 8

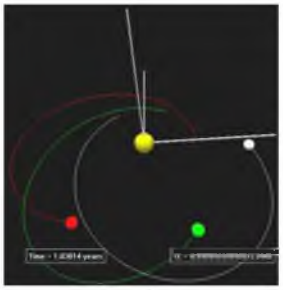


Рис. 9

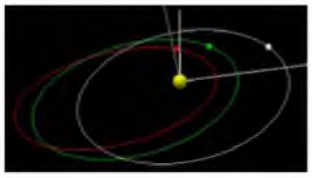


Рис. 10

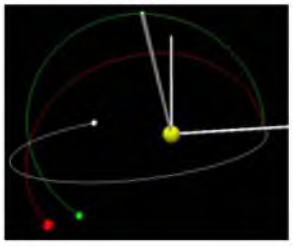


Рис. 11

compare_i.txt – включає зірку і три планети з ідентичними орбітальними елементами, за винятком орбітальних нахилів (*рис. 11*). Два ненульових нахили демонструють геометричний зміст даного поняття. Один з нахилів, більший за 90°, демонструє ретроградну орбіту. Як відомо, в Сонячній системі ретроградну орбіту має Уран.

compare_lan.txt – включає зірку і три планети з ідентичними орбітальними елементами, за винятком довготи точки сходження (*рис. 12*). Довготу точки сходження з усіх орбітальних елементів візуалізувати найважче, тому дана модель буде корисна для розуміння сутності цього поняття.

compare_ma.txt – включає зірку і три планети з ідентичними орбітальними елементами, за винятком початкових середніх аномалій – стартових позицій і швидкостей (*рис. 13*).

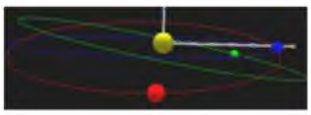


Рис. 12

Рис. 13

draw_full_solar_system.txt – створює картину всіх дев'яти планет Сонячної системи. Радіус Сонця відображається в коректному масштабі (*рис. 14*).

rogue_star_inner.txt – у цій моделі невелика зірка (0,5 маси Сонця) блукає серед внутрішніх планет і спотворює їхньої орбіти (*рис. 15*).

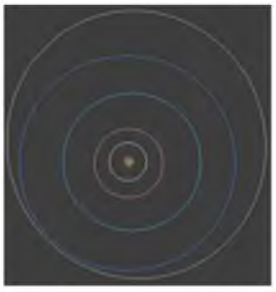


Рис. 14

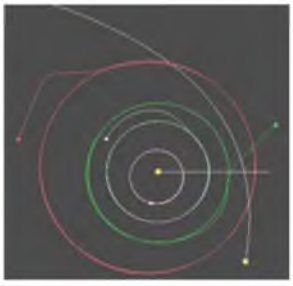


Рис. 15

rogue_star_outer.txt – у цій моделі зірка в 1,5 маси Сонця блукає серед зовнішніх планет і спотворює їхньої орбіти (*рис. 16*).

sun_wobble.txt – включає Сонце, Юпітер і Сатурн. При збільшенні зображення можна спостерігати рух Сонця під дією Юпітера і Сатурна (*рис. 17*).

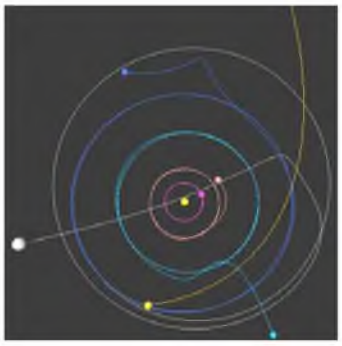


Рис. 16

Рис. 17

Висновки:

1. Стабілізація технологічної складової курсу інформатики сприяє підвищенню рівня теоретичної підготовки, дозволяє створити стабільні підручники, надає широкі можливості по вибору апаратних та програмних засобів навчання інформатики, знижуючи вартість оволодіння ними за рахунок використання ліцензійно чистого вільно поширюваного програмного забезпечення [3].

2. Засобом стабілізації курсу ООП є застосування кросплатформенних інтерпретованих ООП-мов, придатних для швидкого прототипування програм. Традиційно ефективний засіб підвищення практичної значущості виучуваного предмету – використання міжпредметних зв'язків – реалізується через розгляд об'єктно-орієнтованих моделей. Застосування графічної бібліотеки OpenGL через модуль VPython дозволяє у простий спосіб створити високоякісні 3D-анімації.

3. При вивченні теми "Динаміка Сонячної системи" доцільно використати програмний комплекс VPNBody. Сферами застосування VPNBody є лекційні демонстрації планетних орбіт, орбітальних елементів, тривимірного характеру орбіт (орбітних нахилів), планетних конфігурацій (протистоянь, сполучень тощо), ретроградного (зворотного) руху, ідентифікація екзосистем. Моделі, створені за допомогою VPNBody, можуть використовуватися також для лабораторних робіт.

4. Мова програмування Python не дозволяє створити швидкі програми, тому їх ускладненням доцільним є перехід до інших середовищ моделювання (зокрема, мови C++) [5].

Список використаних джерел:

1. *Гетманова Е.Е.* Моделирование физических процессов в VPython: Учебное пособие. – Харьков: ХНУРЭ, 2004. – 82 с.
2. *Поліщук О.П., Семеріков С.О.* Методичні та організаційні проблеми навчання комп'ютерного програмування у вищих навчальних закладах // Теорія та методика навчання математики, фізики, інформатики: Збірник наукових праць. Випуск VI: В 3-х томах. – Кривий Ріг: Видавничий відділ НМетАУ, 2006. – Т. 3: Теорія та методика навчання інформатики. – С.8-11.
3. *Поліщук О.П., Теплицький І.О., Семеріков С.О.* Систематичне навчання моделюванню в підготовці майбутнього вчителя // Комп'ютерне моделювання в освіті: Матер. Всеукр. наук.-метод. семін.: Кривий Ріг, 26 квітня 2006 р. – Кривий Ріг: КДПУ, 2006. – С.48-49.
4. *Семеріков С.О., Теплицький І.О.* Інваріантність до операційної системи та мови програмування як засіб фундаменталізації курсів інформатики у ВНЗ // Інформаційні технології в освіті, науці і техніці: Матеріали V Всеукраїнської конференції молодих науковців ІТОНТ–2006: Черкаси, 3-5 травня 2006 р. – Черкаси: ЧНУ, 2006. – С.140.
5. *Соловйов В.М., Семеріков С.О., Теплицький І.О.* Інструментальне забезпечення курсу комп'ютерного моделювання // Комп'ютер у школі та сім'ї. – 2000. – №2. – С.28-32.
6. *Теплицький І.О.* Елементи комп'ютерного моделювання: Навч. посібник. – Кривий Ріг: КДПУ, 2005. – 208 с.
7. *Теплицький І.О., Семеріков С.О.* Комп'ютерне моделювання руху тіл під дією сили всесвітнього тяжіння // Зб. наук. праць Кам'янець-Подільського держ. ун-ту: Серія педагогічна. Вип. 10: Дидактики дисциплін фізико-математичної та технологічної освітніх галузей. – Кам'янець-Подільський: Кам.-Под. держ. ун-т, ІВВ, 2004. – С.166-172.
8. *Шлеер С., Меллор С.* Объектно-ориентированный анализ: моделирование мира в состояниях. – К.: Диалектика, 1993. – 240 с.
9. *Rodney* Dunning. VPNBody User's Manual (http://panther.bsc.edu/~rdunning).

The article is devoted to methodic teaching of computer modeling in course «Object-oriented technology of programming» at physical-mathematical faculties of the pedagogical schools institutions.

Key words: computer simulation, Python, object-oriented technology of programming, class library VPython, methodical system of training.

Отримано: 7.05.2006.

УДК 53(07)

І.А. Мазурик, С.П. Величко

Кіровоградський державний педагогічний університет ім. В.Винниченка

ДО ПРОБЛЕМИ ВДОСКОНАЛЕННЯ НАВЧАЛЬНОГО ФІЗИЧНОГО ЕКСПЕРИМЕНТУ ЯК ОСНОВНОЇ СКЛАДОВОЇ КОМПЕТЕНТНОСТІ СУЧАСНОГО ВЧИТЕЛЯ ФІЗИКИ

У статті аналізуються дидактичні й ергономічні вимоги та особливості виконання шкільного фізичного експерименту з метою підвищення компетентності сучасного вчителя фізики.

Ключові слова: фізичний експеримент, компетентність, сучасний вчитель, дидактичні вимоги, ергономічні вимоги.

Досвід і практика викладання фізики у різних навчальних закладах має специфічні особливості щодо дотримання дидактичних принципів, оскільки крім теоретичної та практичної частини курс фізики взагалі і зокрема під час викладання певного його змісту в школі включає в себе експериментальну, яка уособлює методику навчання. Фізичний експеримент покликаний "матеріалізувати" теорію, засвідчити її істинність. Втім навчальний фізичний експеримент виявляється не завжди ефективним для доказу основ фізичної теорії. Вирішення проблеми підвищення якості фізичного експерименту криється у специфіці дотримання експериментатором (учителем чи учнем) дидактичних та ергономічних принципів, серед яких особливу увагу приділяють принципам наочності, науковості, достовірності, доступності, послідовності, систематичності.

Зокрема, застосування наочності в навчанні має на меті забезпечити «живе споглядання», воно може розглядатися як перший етап в пізнанні учнями фізичних явищ.

Принцип наочності випливає, з одного боку, із закономірностей процесу пізнання, початковим компонентом якого є споглядання, а з другого боку, у процесі пізнання людина передусім використовує першу сигнальну систему. До того ж треба зважати, що на початковому етапі навчання (в дошкільному і молодшому шкільному віці) мислення дітей характеризується конкретно-образними проявами.

Тому використання наочності у навчанні сприяє поєднанню конкретного з абстрактним, раціонального з ірраціональним, теоретичних знань з практичною діяльністю. Ще Я.А.Коменський стверджував, що все, що тільки можна, надавати для сприймання органами відчуттів; це повинне стати для учителів золотим правилом.

У процесі навчання фізики демонстраційний дослід передає інформацію в основному за допомогою зорових образів, тому забезпечення гарної видимості під час демонструвань – одна з найважливіших вимог до нього. Ігнорування цієї вимоги, як правило, приводить до порушення