

### Навчання технологій програмування магістрів інформатики

**Анотація:** У статті аналізується необхідність вдосконалення класичного курсу програмування для майбутніх учителів інформатики відповідно до сучасного стану розвитку галузі програмування. Обґрунтовується можливість впровадження в навчальний процес курсу „Технології програмування”, призначеного для підготовки магістрів інформатики першого року навчання. Вивчення курсу повинне забезпечити формування у студентів базових знань, навичок, вмінь та досвіду роботи, які стосуються сучасних методологій та технологій розробки програмних засобів. На прикладі однієї теми в статті описані можливі шляхи набуття практичних навичок, умінь та досвіду роботи студентів з окремих питань тестування програмних засобів.

**Ключові слова:** технології програмування, магістри інформатики, тестування за методом чорної скриньки.

Курс програмування є одним з основних курсів предметної підготовки майбутнього вчителя інформатики. Тому його зміст, форми, методи і засоби навчання повинні відповідати сучасному стану мов, методів і технологій програмування, а також орієнтуватися на перспективу їх розвитку. Однак практика показує, що рівень підготовки майбутніх учителів інформатики, зокрема магістрів інформатики в галузі програмування, не є високим і потребує вдосконалення відповідно до сучасного стану розвитку галузі програмування. Опитування студентів-магістрів факультету інформатики НПУ імені М.П. Драгоманова показало, що недостатньо розглянутими в галузі програмування є питання, які стосуються вивчення етапів життєвого циклу розробки програмних засобів, сучасних методологій розробки програмних засобів, сучасних форм програмування тощо.

В класичний курс програмування майбутніх учителів інформатики не ввійшли такі важливі питання, які стосуються тестування програмного засобу, вибору парадигми програмування відповідно до поставленої задачі, розробки технічного завдання стосовно програмного засобу та інші.

Зазначимо, що деякі питання, які стосуються тестування програмного засобу, розробки технічного завдання, сучасних форм програмування тощо входять до шкільного стандарту з інформатики рівня стандарт, деякі до академічного і/або поглибленого рівня. Отже вивчення їх майбутніми учителями інформатики є вкрай необхідним.

Аналіз проблем інформатичної освіти, дослідження теоретичних і методичних аспектів навчання інформатики знайшли відображення в працях А.П. Єршова, С.А. Бешенкова, В.В. Губарева, М.І. Жалдака, В.В. Лаптева, М.П. Лапчика, В.М. Монахова, Н.В. Морзе, О.О. Ракітіної, С.А. Ракова, Ю.С. Рамського, З.С. Сейдаметової, С.О. Семерікова, М.Л. Смутьсон, О.М. Спіріна, Ю.В. Триуса, Г.Ю. Цибко, С.М. Яшанова та інших.

Коротко проаналізуємо досягнення вітчизняних і зарубіжних науковців в галузі алгоритмізації і програмування, а також відзначимо визначних вітчизняних і зарубіжних вчених, які здійснили вагомий внесок в процес включення основ програмування до змісту загальної середньої освіти в Україні в 60-тих роках минулого століття [1,2,3].

Створенням перших навчальних посібників з програмування займалися вчені В.М. Монахов, С.І. Шварцбурд, В.Г. Ашкінузе, А.Л. Брудно, Р.С. Гутер, П.Т. Резніковський та ін. Розробкою методики навчання програмування в машинних кодах займалися В.М. Монахов, Р.С. Гутер, П.Т. Резніковський та ін. Алгоритмічні мови досліджували І.М. Антипов, С.А. Абрамов, В.В. Щенніков, Г.А. Звенигородський та ін.

Загальноосвітні аспекти навчання програмування і питання взаємозв'язків навчання програмування і математики досліджували В.М. Монахов, С.І. Шварцбурд, методичні аспекти добору засобів описування алгоритмів – Н.Б. Бальцюк, П.С. Кацева, А.Ф. Касторнов, Е.І. Кузнецов, можливості вивчення елементів програмування для машин Поста і Тюрінга - В.А. Успенський, В.М. Касаткін, О.А. Кузнецов, І.А. Переход.

Базовими питаннями формування алгоритмічної культури учнів в процесі навчання математики і програмування займалися В.М. Монахов, М.П. Лапчик, Л.П. Червочкіна.

Питаннями методики використання в процесі навчання математики і програмування мікрокалькуляторів займалися І.М. Антипов, В.Г. Болтянський, М.І. Жалдак, Ю.С. Рамський, С.І. Шварцбурд, М.І. Шкіль та ін.

Всесвітньо відомими вважаються роботи з програмування А.Я. Архангельського, Н. Вірта, Е. Дейкстра, С. Ж. Дейта, А.І. Кітова, Д.Е. Кнута, А. Маккі, С. Макконелла, Р.У. Себеста та ін.

Сьогодні інформатика стала однією з лідерів серед навчальних дисциплін. В середніх навчальних закладах за обсягом годин вона поступається лише математиці та українській мові. Також в рамках вивчення інформатики збільшено години на навчання змістової лінії алгоритмізація та програмування. Все це вимагає від вчителя інформатики ґрунтовної обізнаності в галузі програмування.

Отже, підсумовуючи сказане, приходимо до розуміння необхідності підсилення підготовки майбутніх вчителів інформатики в галузі алгоритмізації і програмування. Таким чином підвищена актуальність даного питання зумовили впровадження курсу „Технології програмування”, призначеного для підготовки магістрів – майбутніх учителів інформатики першого року навчання, які навчаються за напрямом підготовки «Інформатика». Цей курс належить до нормативної частини циклу науково-предметної підготовки. Його вивчення повинне забезпечити формування у студентів базових знань, які стосуються теоретичних основ сучасних технологій програмування, узагальнення та систематизацію знань з технологій проектування програмних засобів, сучасних методологій розробки програмного забезпечення, набуття студентами практичних навичок та умінь проектування програмних засобів на всіх етапах життєвого циклу розробки програмних засобів.

Основними завданнями вивчення дисципліни “Технології програмування” є:

- ✓ Розкрити місце і значення теоретичних основ технологій розробки програмних засобів у методології навчання програмування майбутніх учителів інформатики.
- ✓ З’ясувати переваги та недоліки сучасних методологій програмування, різних парадигм програмування відповідно до поставленої задачі.
- ✓ Розширити знання студентів про:
  - тенденції розвитку сучасних технологій програмування і програмного забезпечення;
  - особливості використання технології функціонального програмування у різних системах програмування;
  - особливості використання технології об’єктно-орієнтованого програмування у різних системах програмування;
  - особливості використання сучасних методів програмування під час розв’язування практичних задач.
- ✓ Ознайомити студентів з:
  - загальними принципами розробки програмних засобів;
  - етапами життєвого циклу розробки програмних засобів;
  - сучасними методологіями розробки програмних засобів: Rapid Application Development, Microsoft Solutions Framework, ітераційна модель, eXtreme Programming, Scrum, Kanban, Rational Unified Process, розробка програмних засобів за міжнародними стандартами;
  - сучасними формами програмування: одиноке, парне, командне програмування.
- ✓ Сформувати уміння:
  - розробки програмних засобів на етапі планування, кодування і тестування програмних засобів;
  - здійснювати тестування програмного засобу за методами білої, чорної, сірої скриньки;
  - розробляти і описувати технічне завдання програмного засобу, створення інструкцій для розробника і для користувача;
  - доцільно обирати систему (парадигму) програмування відповідно до поставленої задачі;
  - розробляти програмні засоби, використовуючи технології функціонального програмування;
  - розробляти програмні засоби, використовуючи технології об’єктно-орієнтованого програмування.

Розглянемо детальніше тему «Тестування програмного засобу за методом чорної скриньки», яка вивчається студентами в рамках даної дисципліни. Мета навчання даної теми - проаналізувати різні підходи до тестування програмного засобу, сформувати навички тестування програмного засобу за методом чорної скриньки.

В рамках даної теми студентам необхідно засвоїти наступні поняття:

- тестування програмного засобу;
- налагодження програмного засобу;
- принцип оптимальної стратегії розробки тестів;
- види тестування;
- рівні тестування;
- метод функціонального тестування (чорна скринька).

А також вміти:

- здійснювати тестування за методом випадкового тестування;

- здійснювати тестування за методом еквівалентного розподілу;
- здійснювати тестування за методом аналізу граничних умов;
- здійснювати тестування за методом причинно-наслідкових зав'язків;
- здійснювати тестування за методом припущення помилки.

Під час опанування даної теми студентам необхідно виконати наступні індивідуальні завдання.

1. Ознайомитися з теоретичним матеріалом до теми.
2. Відповідно до номера в журналі обрати варіант задачі для тестування програмного засобу за методом чорної скриньки.
3. Розробити алгоритм розв'язування обраної задачі (у вигляді блок-схеми).
4. Подати алгоритм через опис програми будь-якою мовою програмування (перевага надається вже вивченим мовам програмування).
5. Розробити тести за методом еквівалентного розподілу:
  - a. виокремити класи еквівалентності і занести їх у відповідну таблицю (табл. 1):

**Таблиця 1**

№	Вхідні значення	Класи коректних даних	Класи некоректних даних

Наведемо приклад виконання даного завдання.

**Задача.** Додати два цілі числа. Кожен з доданків – двозначне ціле число.

Класи еквівалентності занесені до табл. 2:

**Таблиця 2**

№	Вхідні / вихідні значення	Класи коректних даних	Класи некоректних даних
1	Перший доданок, другий доданок	від -99 до -10 від -9 до -1 0 від 1 до 9 від 10 до 99	>99 <-99 1.5 Q
3	Сума	від -198 до -100 від -99 до -1 0 від 1 до 99 від 100 до 198	>198 <-198 'некоректні дані'

- b. використати виокремлені класи для побудови тестів. Протестувати розроблений програмний засіб і заповнити таблицю (табл. 3):

**Таблиця 3**

Номер тесту	Призначення тесту	Вхідні значення	Очікуваний результат	Реакція програми	Висновок

6. Розробити тести за методом аналізу граничних умов:
  - a. для виокремлених класів еквівалентності вказати граничні й спеціальні значення і заповнити відповідну таблицю (табл. 4).

**Таблиця 4**

№	Вхідні значення	Класи коректних даних	Класи некоректних даних	Граничні і спеціальні значення

Наведемо приклад виконання даного завдання для попередньої задачі.

Класи еквівалентності вже занесені до табл. 2. Визначимо граничні і спеціальні значення (табл. 5).

**Таблиця 5**

№	Вхідні / вихідні значення	Граничні і спеціальні значення
1	Перший доданок, Другий доданок	0, 1, -1, 9, -9 10, -10 99, -99 100, -100
3	Сума	(-99, -99) (-49, -51) (99, -99) (49, 51)

- b. використати виокремлені граничні і спеціальні значення для побудови тестів. Протестувати розроблений програмний засіб і заповнити таблицю (табл. 6).

**Таблиця 6**

Номер тесту	Призначення тесту	Вхідні значення	Очікуваний результат	Реакція програми	Висновок

7. Розробити тести за методом причинно-наслідкових зв'язків:

- a. на основі аналізу семантичного (сислового) змісту задачі побудувати таблицю істинності (True/False), в якій послідовно перебирати всі можливі комбінації причин і визначити наслідки кожної комбінації причин (табл. 7).

**Таблиця 7**

№	Умова 1	Умова 2...	... Умова N	Результат

Наведемо приклад виконання даного завдання для попередньої задачі (табл. 8).

**Таблиця 8**

№	Перший доданок	Другий доданок	Результат
1	0	0	False
2	0	1	False
3	1	0	False
4	1	1	True

- b. використати таблицю істинності для побудови тестів. Протестувати розроблений програмний засіб і заповнити таблицю (табл. 9).

**Таблиця 9**

Номер тесту	Призначення тесту	Вхідні значення	Очікуваний результат	Реакція програми	Висновок

Наведемо приклад виконання даного завдання для попередньої задачі (табл. 10).

**Таблиця 10**

Номер тесту	Призначення тесту	Вхідні значення	Очікуваний результат	Реакція програми	Висновок
1	для перевірки правильності введених значень	(300,-300)	False	False	програма виконується коректно
2	для перевірки правильності введених значень	(300,25)	False	False	програма виконується коректно
3	для перевірки правильності введених значень	(25,-300)	False	False	програма виконується коректно
4	для перевірки правильності введених значень	(25,25)	True	True	програма виконується коректно

8. Розробити тести за методом припущення помилки. За необхідності додати невстановлені досі тести за методом припущення помилки, заповнити стосовно них таблицю (табл. 11).

**Таблиця 11**

Номер тесту	Призначення тесту	Вхідні значення	Очікуваний результат	Реакція програми	Висновок

Наведемо приклад виконання даного завдання для попередньої задачі (табл. 12).

**Таблиця 12**

Номер тесту	Призначення тесту	Вхідні значення	Очікуваний результат	Реакція програми	Висновок
1	для перевірки правильності введених значень	(a,b)	False	False	програма виконується коректно
2	для перевірки результату обчислень	(25,25)	True (50)	True (50)	програма виконується коректно

9. Визначити загальну кількість тестів і оформити звіт з лабораторної роботи, в якому мають міститися:
- умова задачі;
  - алгоритм програми;
  - код програми;
  - для кожного з чотирьох методів – відповідні таблиці;
  - сумарна зведена таблиця стосовно всіх тестів;
  - висновок про роль тестування з використанням методу чорної скриньки і можливості його застосування;
  - загальний висновок про переваги і недоліки даного методу.

Для дистанційної підтримки навчання було розроблено електронний курс «Технології програмування», в якому містяться лекційні, лабораторні та додаткові матеріали, а також тестовий контроль [5].

Як показує практика, в сучасних умовах неможливо забезпечити якісну підготовку вчителя інформатики без цілісного уявлення про методологію і технологію розробки програмних засобів. Тому підготовка майбутніх фахівців з інформатики повинна включати формування необхідних знань і умінь, що стосуються галузі алгоритмізації і програмування, зокрема технологій програмування.

#### Список використаних джерел

- Жалдак М. І. Комп'ютерно-орієнтовані системи навчання – становлення і розвиток / М.І. Жалдак // Науковий часопис НПУ імені М. П. Драгоманова. Серія 2 : Комп'ютерно-орієнтовані системи навчання. – 2010. – №. 9. – С. 3-9.
- Історія кафедри теоретичних основ інформатики Національного педагогічного університету ім. М.П.Драгоманова [Електронний ресурс] – Режим доступу: <http://ktoi.npu.edu.ua/uk/istoriia-kafedry>.
- Жалдак М.І. 20 років становлення і розвитку методичної системи навчання інформатики в школі та педагогічному університеті / М.І. Жалдак, Н.В. Морзе, Ю.С. Рамський // Науковий часопис НПУ імені М.П. Драгоманова. – К.: НПУ ім. М.П.Драгоманова, 2006. – № 4 (11). – С. 3-13.
- Жалдак М.І. Проблеми інформатизації навчального процесу в школі і в вузі / М.І. Жалдак // Сучасні інформаційні технології в навчальному процесі: Зб. наук. пр. – К.: КДПІ імені М.П. Драгоманова, 1991. – С. 3-16.
- Технології програмування [Електронний ресурс].- Режим доступу: <http://www.moodle.fi.npu.edu.ua/course/view.php?id=92>

#### Изучение технологии программирования магистрами информатики

*М.А. Умрик*

**Аннотация:** В статье анализируется необходимость совершенствования классического курса программирования для будущих учителей информатики в соответствии с современным состоянием развития области программирования. Обосновывается возможность внедрения в учебный процесс курса "Технологии программирования", предназначенного для подготовки магистров информатики первого года обучения. Изучение курса должно обеспечить формирование у студентов базовых знаний, навыков, умений и опыта работы, касающиеся современных методологий и технологий разработки программных средств. На примере одной темы в статье описаны возможные пути приобретения практических навыков, умений и опыта работы студентов по отдельным вопросам тестирования программных средств.

**Ключевые слова:** технологии программирования, магистры информатики, тестирование методом черного ящика.

#### Preparation of informatic masters: scientific researches via ICT

*М.А. Umryk*

**Resume:** The article analyzes the need to improve the classical course of programming for future teachers of informatics in accordance with the current state of development of the programming industry. The possibility of introducing the course "Programming Technologies", intended for the preparation of masters of computer science in the first year of study, is substantiated in the educational process. The example of one topic in the article describes the possible ways of acquiring practical skills, skills and experience of students on specific issues of software testing.

**Keywords:** technologies of programming, masters of informatics, black box testing.