

each of the methods has its advantages and disadvantages. Each of the methods can be used alone or in combination. The strengths and weaknesses of these alternatives to a chemical experiment are considered, taking into account the results of conversations with applicants for education.

The theoretical material of the article is illustrated with freeze frames from students' video reports. Students made these video reports when they studied various chemical disciplines. The pictures show how to independently study and investigate the properties of organic and inorganic substances in a home chemical laboratory.

The authors are aware that the study does not reveal all aspects of the indicated problem of including a chemical experiment in the educational process in the context of organizing distance learning. Therefore, it is planned to conduct the following scientific research on the topic.

Keywords: educational process, teaching chemistry, distance learning, chemistry experiment, video experiments, interactive demonstrations, virtual chemistry experiments, home chemistry experiments.

УДК 372.8 (004)

DOI 10.5281/zenodo.6618615

В. М. Базурін

ORCID 0000-0002-6614-4889

Київський національний
торгівельно-економічний університет

МЕТОДИКА ВИВЧЕННЯ ОСНОВНИХ АЛГОРИТМІЧНИХ КОНСТРУКЦІЙ З ВИКОРИСТАННЯМ ЗАСОБІВ КОМП'ЮТЕРНОЇ ГРАФІКИ МОВИ PYTHON

Проблема навчання учнів базової середньої школи є актуальною в даний час. Успішність навчання учнів основ програмування значною мірою залежить від мотивації учнів. На підвищення рівня мотивації учнів до вивчення програмування впливає низка чинників, у тому числі й інтерактивний зміст навчання. Python є популярною мовою програмування у даний час і вивчається в загальноосвітній школі. Python має вбудовані засоби комп'ютерної графіки. Ці засоби вивчаються у шкільному курсі інформатики. Мета статті – розкрити особливості методики навчання основних алгоритмічних конструкцій учнів 7 класу з використанням засобів комп'ютерної графіки мови Python. Розроблена методика спирається на сталу послідовність вивчення алгоритмічних конструкцій і типів даних, але використовує для цього засоби комп'ютерної графіки. Результати роботи програми учні можуть наочно спостерігати на екрані комп'ютера. Це сприяє швидкому виявленню неточностей і помилок в алгоритмах. Методика побудована за принципом «від простого до складного» і розрахована на учнів 7 класу.

Передумовами застосування даної методики є вивчення основних засобів і методів комп'ютерної графіки на мові Python: модуля turtle, методів модуля turtle.

У процесі створення зображень та анімацій за допомогою мови Python учнями використовуються основні алгоритмічні конструкції: слідування, розгалуження, повторення. Під час створення програм учні повинні використовувати також підпрограми, списки, рядкові дані і складати підпрограми для обробки таких типів даних, як рядки і списки.

Розроблену методику було апробовано на заняттях гуртка «Інформатика» з учнями 6-7 класів. У результаті було відмічено зростання мотивації учнів до вивчення програмування і успішне засвоєння ними основних алгоритмічних конструкцій, а також їх реалізації на мові Python.

Для застосування даної методики достатньо рівня методичної компетентності більшості вчителів інформатики і наявних програмно-технічних засобів.

Ключові слова: учні, комп'ютерна графіка, програмування, алгоритмічна конструкція, Python.

Постановка проблеми. Чинним Державним стандартом базової середньої освіти з інформатики [2] передбачено вивчення основ програмування учнями загальноосвітніх шкіл, починаючи з 7 класу. Проте в процесі вивчення основ програмування виникає низка проблем, пов'язаних з такими чинниками: рівня особистісного розвитку учнів; рівня сформованості логічного та алгоритмічного мислення; мотивації учнів; змісту і послідовності навчального матеріалу.

Не є секретом те, що успішність вивчення програмування значною залежить від мотивації учнів. Особливу роль відіграє мотивація для обдарованих учнів. Саме тому необхідно застосовувати засоби, які сприяють зростанню мотивації учнів: інтерактивні засоби, відповідний зміст навчання та інші.

Аналіз актуальних досліджень. Проблема навчання учнів і студентів основ програмування є актуальною і досліджується широким колом науковців в Україні і за кордоном.

Так, S. Combéfis, G. Beresnevičius, V. Dagienė досліджують ефективність ігрового підходу до вивчення програмування [9]. У їх дослідженні проаналізовано існуючі класифікації навчальних ігор з програмування, розкрито можливості ігрових онлайн-платформ, призначених для вивчення програмування. Науковці зазначають, що застосування таких ігрових платформ, як Lightbot, сприяє формуванню в учнів логічного та алгоритмічного мислення. Далі учні створюють власні ігри з використанням середовищ Scratch і Flowlab. Як зазначають науковці, застосування ігрових платформ значною мірою підвищує мотивацію учнів до вивчення програмування, сприяє формуванню в них алгоритмічного мислення.

Інші вчені, C. S. Prat, T. M. Madhyastha, M. J. Mottarella, C. H. Kuo аналізують чинники, які впливають на швидкість вивчення мови Python за методикою, запропонованою Code academy: ставлення до мови, математичні здібності, пізнавальні здібності [21]. Водночас дане дослідження було здійснено з студентами 18-35 років, отже, його результати можуть бути обмежено застосовані для вивчення програмування учнями 12-13 років.

Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., & Lahmine, S. досліджували проблему навчання основ програмування шляхом створення ігор у середовищі Scratch [19]. Дослідники вказують на ефективність навчання основ програмування у середовищі Scratch порівняно з навчанням програмування на мові Pascal. Як зазначають вчені, навчання програмування шляхом створення ігор у середовищі Scratch сприяє зростанню мотивації учнів до навчання і, як наслідок, більш ефективному формуванню уміння програмувати.

Науковці I. Drosos, P. J. Guo, C. Parnin демонструють дослідження ефективності навчання учнів програмування у середовищі Python Tutor [12]. Дослідники вважають, що основними чинниками, які викликають розчарування у програмістів-початківців: синтаксичні помилки, використання нішевих мовних функцій і розуміння коду високої складності. На основі аналізу результатів дослідження можна зробити висновок про те, що мова програмування, яка вивчається, повинна мати порівняно простий синтаксис, прості алгоритмічні конструкції,

J. Nouri, L. Zhang, L. Mannila, E. Norén, досліджували основні стратегії навчання програмування, які використовуються шведськими вчителями: аналогове програмування, програмування за допомогою роботів і додатків, блочне програмування і текстове програмування [18].

J. Figueiredo, N. Gomes, F. J. García-Peñalvo, вивчали проблему навчання студентів програмування на курсах, вони формулюють такі вимоги до системи завдань з програмування: вправи з програмування повинні бути створені без використання різних технологій, не бути електронним курсом [13].

N. Dass, J. Kim, S. Ford, S. Agarwal, D. H. Chau, досліджують питання навчання основ програмування засобами доповненої реальності [11].

F. L. Khaleel, N. S. Ashaari, T. S. Wook, досліджували проблему застосування ігрових методів у навчанні програмування на мовах C, C++, Java [15].

Дослідження Y. Bosse, M. A. Gerosa присвячено з'ясуванню основних труднощів у процесі вивчення логіки комп'ютерного програмування [7]. Дослідники побудували

педагогічну модель, яка сприятиме акцентуванню уваги учнів на заняттях, а для вчителів – застосовувати методи і інструменти для підтримки навчання.

У центрі уваги L. Goosen, D. Van Heerden, [14] знаходяться освітні технології і їх вплив на розв'язування студентами різних завдань у галузі ІКТ.

Проблемам навчання учнів початкової школи програмування за допомогою візуальних середовищ присвячено дослідження K. Asad, M. Tibi, J. Raiyn, [6]. Дослідниками була розроблена технологія навчання програмування за допомогою візуального 2D 3D-кодування, розробкою курсів «Код з Анною та Ельзою» і «Академія черепах», в яких використовується мова програмування LOGO. Експеримент було проведено серед учнів 4-5 класів, яким пропонувалися завдання трьох категорій: мотиваційні, змагальні і випробувальні.

Дослідження, проведені F. L. Khaleel, N. S. Ashaari, T. S. M. T. Wook, Ismail, ставили метою визначення ефективності ігрового підходу до навчання основ програмування [16]. Дослідники визначають гейміфікацію навчання програмуванню як засіб формування інтересу учнів до вивчення програмування.

Дослідження O. Revelo-Sánchez, C. A. Collazos-Ordóñez, J. A. Jiménez-Toledo розкриває проблеми, пов'язані з організацією спільної роботи (Collaborative Work) у курсі програмування [22]. Спільна робота (Collaborative Work) є актуальною дидактичною технологією, яка широко застосовується у навчанні програмування, інформатики та інших предметів.

У дослідженні A. Theodoropoulos, A. Antoniou, G. Lepouras [24], визначається вплив серйозних ігор на процес навчання програмування учнів середньої школи за допомогою онлайн-середовищ.

У дослідженні J.C. Paiva, J. P. Leal, R. A. Queirós, розкриваються концепція, архітектура і особливості побудови середовища програмування Enki [20].

Інші науковці, T. Tang, S. Rixner, J. Warren [23], описують середовище навчання основ програмування, розкривають особливості мислительних процесів, які лежать в основі побудови цього середовища. Об'єктом дослідження є бібліотека графічного інтерфейсу для інтерактивного програмування на мові Python.

Об'єктом дослідження таких науковців, як C. S. Chang, C. H. Chung, J. A. Chang, є ігрові технології навчання програмування у вищих закладах освіти [8]. Результати даного дослідження дають підстави стверджувати, що застосування ігрових технологій сприяє зростанню рівня мотивації, а отже, й підвищенню ефективності навчання програмування студентів вищих навчальних закладів.

Проблема застосування у навчанні програмуванню інтерактивних середовищ присвячене дослідження J. M. Costa, G. L. Miranda [10]. Науковці пропонують методику навчання основ програмування з використанням середовища Alice, що сприяє формуванню у учнів інтересу до програмування і, як підсумок, – підвищенню ефективності навчання.

Як зазначають R. Layona, B. Yulianto, Y. Tunardi [17], однією з проблем у навчанні програмування є відсутність інтерактивного навчального змісту. Дослідники сконцентрували свою увагу на питанні розробки навчального відео на платформі HTML5 вчителями інформатики і визначенні впливу інтерактивного відео на процес навчання програмування.

Отже, проблеми мотивації навчання і застосування інтерактивних методик досліджуються широким колом учених і є актуальними на даний час.

Мета статті – розкрити особливості методики навчання основних алгоритмічних конструкцій учнів 7 класу з використанням засобів комп'ютерної графіки мови Python.

Виклад основного матеріалу. Проблеми навчання основ програмування, зокрема, на мові Python, знаходяться в центрі уваги таких дослідників, як С.С.Жуковський, С.В.Матвійчук [3], Н.В.Дегтярьова, В.В.Макарова [0], Н.В.Морзе [4] та інших.

Вивчення основ програмування здійснюється відповідно до навчальної програми з інформатики для 5-9 класів. Послідовність вивчення основних тем алгоритмізації і програмування розкривається в підручнику [5].

У вказаних підручниках пропонується така послідовність вивчення основ програмування на мові Python:

1. Мова програмування. Середовище програмування.
2. Величини. Змінні. Вказівка присвоєння.
3. Додаткові модулі.
4. Текстові величини та операції над ними.
5. Опрацювання величин логічного типу. Команда розгалуження.
6. Реалізація алгоритмів повторення мовою програмування.

Дана послідовність вивчення основ програмування є цілком логічною і послідовною. Проте існує інший варіант навчання основ програмування, який, на нашу думку, забезпечує більший інтерес учнів до вивчення основ програмування. Розглянемо цей варіант детальніше.

Практика свідчить, що для учнів 6-7 класів найбільш цікавим моментом вивчення основ програмування на мові Python є комп'ютерна графіка, яка реалізується за допомогою модуля Turtle. Цей модуль є вбудованим і не потребує встановлення. Підключення даного модуля до програми і наступне виведення вікна на екран не викликає жодних технічних або програмних ускладнень.

Пропонований науковцями і авторами підручників зміст теми «Елементи комп'ютерної графіки» включає основні функції і методи модуля turtle і використання алгоритмічної структури «слідування» і зрідка – цикл. Проте у подальшому, під час вивчення наступних тем, їх вивчення можна візуалізувати. Так, алгоритмічна структура «повторення» набуде живого, наочного вигляду у тому випадку, коли її застосувати для побудови малюнків. Так само такий тип даних, як списки, можна використати під час задання координат точок, розмірів фігур, кольорів фігур тощо. Рядки можна застосувати для введення даних і програмування послідовності команд.

Керуючись принципом «від простого до складного», пропонуємо таку послідовність вивчення комп'ютерної графіки:

- 1) вивчення основних методів і функцій об'єкта turtle;
- 2) малювання фігур з використанням алгоритмічної структури «слідування»;
- 3) вивчення алгоритмічної структури «повторення»;
- 4) малювання фігур, в яких зустрічається алгоритмічна структура «повторення», і кількох однотипних фігур;
- 5) використання циклу для малювання кількох однотипних фігур;
- 6) створення підпрограми для малювання фігури;
- 7) використання підпрограм для малювання кількох фігур;
- 8) зчитування даних із списку (кольору, розмірів), використання розгалужень;
- 9) задання траєкторії руху черепашки за допомогою списку і рядка;
- 10) зчитування з файлів даних для малювання фігур.

Вказана послідовність містить більшість тем з основ програмування, які вивчаються у 7 класі. Проілюструємо дану методику прикладами створення програм.

Найпростіший варіант – лінійний алгоритм. При цьому всі інструкції записуються у тій же послідовності, в якій повинні виконуватися. Наприклад, намалюємо квадрат:

```
import turtle
L=90
turtle.forward(L)
turtle.left(90)
turtle.forward(L)
turtle.left(90)
turtle.forward(L)
turtle.left(90)
turtle.forward(L)
turtle.left(90)
```

Аналізуємо з учнями дії, які повторюються. Третій і четвертий рядки програми повторюються чотири рази. Такий же самий квадрат можна намалювати за допомогою алгоритмічної конструкції «цикл»:

```
import turtle
```

```
L=90
for i in range(4):
    turtle.forward(L)
    turtle.left(90)
```

Далі аналізуємо текст програми. Учні повинні прийти до висновку, що використання циклу з параметром надає можливість скорити код програми і зробити його зрозумілішим.

Всі команди, які необхідні для малювання квадрата, можна записати у підпрограму. Після того, як учнів ознайомили з поняттям підпрограми, слід навести приклади підпрограм і їх запис на мові Python. Далі доцільно запропонувати їм створити процедуру, яка малюватиме квадрат:

```
def drawQ(len):
    turtle.pendown()
    for i in range(4):
        turtle.forward(len)
        turtle.left(90)
    turtle.penup()
```

У цьому випадку підпрограма отримує один аргумент – довжину сторони квадрата. Якщо квадрат повинен бути зафарбованим, то слід змінити код:

```
def drawQ(len, col):
    turtle.color(col)
    turtle.begin_fill()
    turtle.pendown()
    for i in range(4):
        turtle.forward(len)
        turtle.left(90)
    turtle.end_fill()
    turtle.penup()
```

Якщо треба намалювати кілька квадратів, розміщених за певним порядком (або розміри яких змінюються за певним законом), то можна використати цикл:

```
import turtle
def drawQ(len):
    turtle.pendown()
    for i in range(4):
        turtle.forward(len)
        turtle.left(90)
    turtle.penup()
L=30
turtle.penup()
for j in range(3):
    drawQ(L)
    turtle.forward(50)
```

Далі, по мірі вивчення наступних тем (наприклад, списків), кольори квадратів можна записати у список і передавати колір елемента списку у підпрограму. Для цього слід ознайомити учнів з поняттям списку, вивчити з ними правила задання списків, а також звернення до елементів списку. Для звернення до елемента списку можна використати цикл з параметром не лише у звичайному форматі, а й цикл з параметром у «пітонівському» стилі.

Приклад програми, яка використовує цикл з лічильником у звичайному стилі:

```
import turtle
def drawQ(len, col):
    turtle.color(col)
    turtle.begin_fill()
    turtle.pendown()
    for i in range(4):
```

```
turtle.forward(len)
turtle.left(90)
turtle.end_fill()
turtle.penup()
L=30
turtle.penup()
colors=['red','green','blue']
for j in range(3):
    drawQ(L,colors[j])
turtle.forward(50)
```

Слід звернути увагу учнів на такий момент: які дії слід виконати, щоб збільшити кількість квадратів, які малює програма? Доцільно запропонувати учням збільшити кількість квадратів різного кольору, які зображує програма на екрані. Учні повинні проаналізувати розроблений алгоритм і перелічити дії, які їм довелося виконати: 1) збільшити кількість кольорів у списку; 2) збільшити кількість повторів у циклі for. Далі доцільно поставити учням запитання: а чи можна зменшити кількість цих дій, тобто, якось автоматизувати обробку списку? Як зробити так, щоб достатньо було б лише дописати колір у список?

Після чого варто заслухати пропозиції учнів. Це цікаве питання, воно оживляє роботу учнів на уроці, спонукає їх думати і шукати варіанти розв'язання задачі.

Після заслуховування варіантів, запропонованих учнями, слід пояснити особливості використання циклу for для обробки списків. Це особливості циклу for у мові Python. У цьому в якості розміру використовується назва списку. Приклад програми:

```
import turtle
def drawQ(len, col):
    turtle.color(col)
    turtle.begin_fill()
    turtle.pendown()
    for i in range(4):
        turtle.forward(len)
        turtle.left(90)
    turtle.end_fill()
    turtle.penup()
L=30
turtle.penup()
colors=['red','green','blue']
for c in colors:
    drawQ(L,c)
turtle.forward(50)
```

Після написання коду слід запропонувати учням доповнити список спочатку одним кольором, потім – ще кількома. Учні повинні прийти до висновку, що для цього достатньо лише дописати назви відповідних кольорів у список.

Цікавим варіантом використання списків є програмування виконавця на різні напрями руху. Для цього з учнями слід повторити алгоритмічну конструкцію «розгалуження», а потім поставити задачу: «Запрограмувати рух черепашки вправо, вліво і вперед, якщо команди записуються у список: F – рух вперед, L – поворот вліво на 90°, R – поворот вправо на 90°». Учні повинні самостійно скласти спочатку функцію, яка виконувати рух черепашки. Орієнтовний код цієї функції такий:

```
def ruh(com:str):
    if com=='F':
        turtle.forward(30)
    elif com=='L':
        turtle.left(90)
    else:
        turtle.right(90)
```

Повний текст програми орієнтовно такий:

```
import turtle
def ruh(com:str):
if com=='F':
turtle.forward(30)
elif com=='L':
turtle.left(90)
else:
turtle.right(90)
commands=['F','L','F','R','F','R','F','L','F']
for c in commands:
ruh(c)
```

Варто запропонувати учням запрограмувати дії виконавця, щоб програма могла намалювати пунктир. Для цього слід доповнити функцію командами: U – підняти перо, D – опустити перо.

Після того, як учні проаналізували код програми, перевірили, як функціонує програма, доцільно поставити перед ними питання: а чи можна записати ці команди у рядку? Учні повинні знову проаналізувати код підпрограми і прийти до висновку: 1) підпрограма отримує рядкову величину (команду); 2) слід створити рядок з командами і перевірити на практиці дієвість програми.

При цьому текст підпрограми не змінюється. Достатньо змінити лише основний код:

```
cdm='FLFRFRFLF'
for c in cdm:
ruh(c)
```

Далі доцільно запропонувати учням удосконалити програму: 1) розширити перелік команд виконавця; 2) додати підпрограму, яка змінює колір ліній у залежності від номера кроку.

Запропоновану методику було апробовано на практиці, під час навчання основ програмування учнів гуртка «Інформатика». Під час занять гуртка було з'ясовано, що використання цієї методики сприяє зростанню інтересу учнів до вивчення основ програмування.

Висновки та перспективи подальших наукових розвідок. Розроблену методику навчання основ програмування можна назвати інтерактивною. Вона не є чимось абсолютно новим, але використовує вбудовані засоби мови Python для роботи з комп'ютерною графікою. Для використання даної методики існують такі об'єктивні передумови:

- 1) наявність вбудованих засобів мови Python для роботи з комп'ютерною графікою і їх можливості для створення малюнків та анімацій;
- 2) використання основних засобів і алгоритмічних конструкцій мови Python;
- 3) наявні компетентності більшості учителів інформатики, на нашу думку, є цілком достатніми для навчання учнів основ програмування за даною методикою.

Перспективами подальших досліджень є:

- 1) розробка системи завдань з комп'ютерної графіки для мови Python з використанням основних алгоритмічних конструкцій;
- 2) практична апробація запропонованої методики з учнями 6-7 класів закладів загальної середньої освіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ / REFERENCES

1. Дегтярьова, Н. В., Макарова, В. В. (2018). Лабораторний практикум як форма організації навчальної діяльності учнів 7 класів. Фізико-математична освіта, 1(15), 181–186. (Dehtiarova, N. V., Makarova, V. V. (2018). Laboratory workshop as a form of organization of educational activities of 7th grade students. Physical and mathematical education, 1(15), 181–186).

2. Державний стандарт базової середньої освіти (2020). Режим доступу: https://osvita.ua/legislation/Ser_osv/76886/ (State standard of basic secondary education (2020). URL: https://osvita.ua/legislation/Ser_osv/76886/).
3. Жуковський, С. С., Матвійчук, С. В. (2016). *Задачі III етапу Всеукраїнської олімпіади з інформатики в Житомирській області у 2016 році та рекомендації щодо їх розв'язування*. Комп'ютер в сім'ї та школі, 8, 39–44. (Zhukovskiy, S. S., Matviichuk, S. V. (2016). Tasks of the third stage of the All-Ukrainian Olympiad in Informatics in Zhytomyr region in 2016 and recommendations for their solution. Computer in the family and school, 8, 39–44).
4. Морзе, Н. В. (2003). *Методика навчання інформатики*. Київ: Навчальна книга. (Morze, N. V. (2003). Methods of teaching computer science. Kyiv.: Navchalna knyha).
5. Морзе, Н. В., Барна, О. В. (2020). *Інформатика. Підручник для 7 класу закладів загальної середньої освіти*. Київ: Оріон. (Morze, N. V., Barna, O. V. (2020). Informatics. Textbook for 7th grade secondary schools. Kyiv: Orion).
6. Asad, K., Tibi, M., Raiyn, J. (2016). Primary School Pupils' Attitudes toward Learning Programming through Visual Interactive Environments. *World journal of education*, 6 (5), 20–26.
7. Bosse, Y., Gerosa, M. A. (2017). Why is programming so difficult to learn? Patterns of Difficulties Related to Programming Learning Mid-Stage. *ACM SIGSOFT Software Engineering Notes*, 41(6), 1–6.
8. Chang, C. S., Chung, C. H., Chang, J. A. (2020). Influence of problem-based learning games on effective computer programming learning in higher education. *Educational Technology Research and Development*, 68(5), 2615–2634.
9. Combéfis, S., Beresnevičius, G., Dagiene, V. (2016). Learning programming through games and contests: overview, characterisation and discussion. *Olympiads in Informatics*, 10(1), 39–60.
10. Costa, J. M., Miranda, G. L. (2017). Relation between Alice software and programming learning: A systematic review of the literature and meta-analysis. *British Journal of Educational Technology*, 48(6), 1464–1474.
11. Dass, N., Kim, J., Ford, S., Agarwal, S., Chau, D. H. (2018). Augmenting coding: Augmented reality for learning programming. In *Proceedings of the Sixth International Symposium of Chinese CHI (Apr, 2018)*. (pp. 156–159).
12. Drosos, I., Guo, P. J., Parnin, C. (2017). HappyFace: Identifying and predicting frustrating obstacles for learning programming at scale. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. (Oct, 2017). (pp. 171–179). IEEE.
13. Figueiredo, J., Gomes, N., García-Peñalvo, F. J. (2016). Ne-course for learning programming. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*. (Nov, 2016). (pp. 549–553).
14. Goosen, L., Van Heerden, D. (2017). Beyond the horizon of learning programming with educational technologies. In *Proceedings of the South Africa International Conference on Educational Technologies*. (pp. 78–90).
15. Khaleel, F. L., Ashaari, N. S., Wook, T. S. M. T. (2019). An empirical study on gamification for learning programming language website. *Jurnal Teknologi*, 81(2).
16. Khaleel, F. L., Ashaari, N. S., Wook, T. S. M. T., Ismail, A. (2017). Methodology for developing gamification-based learning programming language framework. In *2017 6th international conference on electrical engineering and informatics (iceei)*. (Nov, 2017). (pp. 1–6). IEEE.
17. Layona, R., Yulianto, B., Tunardi, Y. (2017). Authoring tool for interactive video content for learning programming. *Procedia computer science*, 116, 37–44.
18. Nouri, J., Zhang, L., Mannila, L., Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1–17.
19. Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., Lahmine, S. (2015). Learning basic programming concepts by creating games with scratch programming environment. *Procedia-Social and Behavioral Sciences*, 191, 1479–1482.

20. Paiva, J. C., Leal, J. P., Queirós, R. A. (2016). Enki: A pedagogical services aggregator for learning programming languages. In Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education. (Jul, 2016). (pp. 332–337).
21. Prat, C. S., Madhyastha, T. M., Mottarella, M. J., Kuo, C. H. (2020). Relating natural language aptitude to individual differences in learning programming languages. Scientific reports, 10 (1), 1–10.
22. Revelo-Sánchez, O., Collazos-Ordóñez, C. A., Jiménez-Toledo, J. A. (2018). Collaborative work as a didactic strategy for teaching/learning programming: a systematic literature review. TecnoLógicas, 21(41), 115–134.
23. Tang, T., Rixner, S., Warren, J. (2014). An environment for learning interactive programming. In Proceedings of the 45th ACM technical symposium on Computer science education. (Mar, 2014). (pp. 671–676).
24. Theodoropoulos, A., Antoniou, A., Lepouras, G. (2016). How do different cognitive styles affect learning programming? Insights from a game-based approach in Greek schools. ACM Transactions on Computing Education (TOCE), 17(1), 1–25.
25. Hill, C. (2020). Learning scientific programming with Python. Cambridge University Press.

Базурин В. Н. Методика изучения основных алгоритмических конструкций с использованием средств компьютерной графики Python.

Аннотация. Проблема обучения учащихся базовой средней школы актуальна в настоящее время. Успешность обучения учащихся основ программирования во многом зависит от мотивации учащихся. На повышение уровня мотивации учащихся к изучению программирования следует ряд факторов, в том числе и интерактивное содержание обучения. Python является популярнейшим языком программирования в настоящее время и изучается в общеобразовательной школе. Python имеет встроенные компьютерные графики. Эти средства изучаются в школьном курсе информатики. Цель статьи – раскрыть особенности методики обучения основным алгоритмическим конструкциям учащихся 7 класса с использованием средств компьютерной графики языка Python. Разработанная методика опирается на установившуюся последовательность изучения алгоритмических конструкций и типов данных, но использует для этого средства компьютерной графики. Результаты работы программы учащиеся могут наблюдать на экране компьютера. Это способствует быстрому выявлению неточностей и ошибок в алгоритмах. Методика построена по принципу «от простого к сложному» и рассчитана на учащихся 7 класса. Предпосылками применения данной методики является изучение основных средств и методов компьютерной графики на языке Python: модуля turtle, методов модуля turtle. В процессе создания изображений и анимаций с помощью языка Python учащимися используются основные алгоритмические конструкции: следование, разветвление, повторение. При создании программ учащиеся должны использовать также подпрограммы, списки, строковые данные и составлять n . Предпосылками применения данной методики является изучение основных средств и методов компьютерной графики на языке Python: модуля turtle, методов модуля turtle. В процессе создания изображений и анимаций с помощью языка Python учащимися используются основные алгоритмические конструкции: следование, разветвление, повторение. При создании программ учащиеся должны использовать подпрограммы, списки, строковые данные и составлять подпрограммы для обработки таких типов данных, как строки и списки. Разработанная методика была апробирована на занятиях кружка «Информатика» с учениками 6-7 классов. В результате был отмечен рост мотивации учащихся к изучению программирования и успешному усвоению ими основных алгоритмических конструкций, а также их реализации на языке Python. Для применения данной методики достаточно уровня методической компетентности большинства учителей информатики и имеющих программно-технических средств.

Ключевые слова: учащиеся, компьютерная графика, программирование, алгоритмическая конструкция, Python.

Bazurin V. M. Methods of studying basic algorithmic constructions using Python computer graphics.

Summary. The problem of teaching elementary school students is relevant today. The success of teaching students the basics of programming largely depends on the motivation of students. There are a number of factors that increase the level of motivation of students to study programming, including the interactive content of learning. Python is a popular programming language today and is taught in high school. Python has built-in computer graphics tools. These tools are studied in the school course of computer science. The purpose of the article is to reveal the peculiarities of the method of teaching the basic algorithmic constructions of 7th grade students using the means of computer graphics of the Python language. The developed technique is based on a constant sequence of studying algorithmic constructions and data types, but uses computer graphics for this purpose. Students can visually observe the results of the program on a computer screen. This facilitates the rapid detection of inaccuracies and errors in algorithms. The method is built on the principle of "simple to complex" and is designed for 7th grade students.

Prerequisites for the application of this technique are the study of basic tools and methods of computer graphics in Python: turtle module, turtle module methods. In the process of creating images and animations using Python, students use basic algorithmic constructions: following, branching, repetition. When creating programs, students should also use routines, lists, string data, and compose routines to process data types such as strings and lists.

The developed technique was tested in the classes of the circle "Informatics" with students of 6-7 grades. As a result, there was an increase in students' motivation to learn programming and their successful mastery of basic algorithmic constructions, as well as their implementation in Python.

To apply this technique, the level of methodological competence of most computer science teachers and available software and hardware is sufficient.

Key words: students, computer graphics, programming, algorithmic construction, Python.

УДК 378.046.4:[37.012-37.014+37.022+37.026]

DOI 10.5281/zenodo.6630533

I. М. Мігельман

ORCID 0000-0002-9817-6690

О. І. Папач

ORCID 0000-0002-8960-5457

Комунальний заклад вищої освіти
«Одеська академія неперервної освіти
Одеської обласної ради»

**ДЕЯКІ ДИНАМІЧНІ ХАРАКТЕРИСТИКИ МЕТОДИЧНОЇ КОМПЕТЕНТНОСТІ
ВЧИТЕЛЯ В КОНТЕКСТІ ПІСЛЯДИПЛОМНОЇ ОСВІТИ**

У статті висвітлюється методика та результати дослідження деяких динамічних характеристик методичної компетентності вчителів, які підвищували кваліфікацію на базі Одеської академії неперервної освіти протягом 2020-2022 рр. Його особливістю стало поєднання кейс-методик з пошуково-емпіричним «якісним» методом фокус-груп. Було виділено кластер значущих дванадцяти характеристик методичних компетентностей, спільних для вчителів споріднених галузей середньої освіти: математичної, природничої та інформатичної. Для різних груп респондентів обирався набір з 3-5 характеристик сформованого кластера. На рефлексивному етапі курсів проводилось експрес-обговорення цих характеристик та анонімно самооцінювання вчителями рівня володіння методичними компетентностями в розрізі вибраних характеристик (за спрощеною шкалою суб'єктно-об'єктного ставлення до методичного досвіду щодо кожної характеристики). Динамічний статус характеристик забезпечувався розгорткою за педагогічним стажем та можливістю дослідників формувати для кожної групи набір характеристик залежно від тематики курсів і