

М.А. Роговець, к.т.н., доц.
*Житомирський військовий інститут ім. С.П. Корольова
Національного авіаційного університету*

ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНО-ІМІТАЦІЙНОЇ МОДЕЛІ ПІДГОТОВКИ ОПЕРАТОРІВ ПОСТІВ РАДІОЧАСТОТНОГО МОНІТОРИНГУ

У статті запропоновано методологічний підхід щодо програмної реалізації інформаційно-імітаційної моделі підготовки операторів складних ергатичних систем з метою автоматизації процесу отримання ними практичних навичок роботи на апаратурі.

Постановка проблеми у загальному вигляді. Сучасні тенденції розвитку радіоелектронних засобів та умов їх застосування вимагають підвищення рівня фахової підготовки спеціалістів радіочастотного моніторингу (РМ) всіх рівнів [1–4]. В свою чергу особлива увага приділяється підготовці операторів постів спостереження, які мають розглядатись як ергатичні системи. При цьому ускладнення завдання якісної підготовки операторів зумовлюється переозброєнням, добуваючи підрозділи новими зразками техніки, які кардинально відрізняються від існуючих як за будовою, так і за принципами застосування. Це вимагає розробки нових підходів до підготовки операторів постів. Однак суттєвим обмежуючим фактором набуття нових якісних практичних навичок операторами є недостатня кількість нових засобів моніторингу, а отже не можуть використовуватися виключно з навчальною метою.

Одним зі шляхів розв'язання даної проблеми є створення програмно-апаратних тренажерів на основі сучасних мультимедійних технологій.

Аналіз останніх досліджень і публікацій. Пошук нових підходів і прийомів підвищення рівня навченості операторів складних ергатичних систем викладено у роботах [2–11]. Так у [2–4] проведено аналіз загальних методологічних і методичних основ теоретичної та практичної підготовки даних фахівців. Принципам використання персональних електронно-обчислювальних машин (ПЕОМ) для підготовки зв'язківців, операторів радіолокаційних систем управління, складних технічних систем присвячено роботи [2–8], де вказується на необхідність розробки і впровадження в навчальний процес програмно-тренажних комплексів. Проблеми їх модернізації та створення висвітлені у роботі [10], в якій звертається увага на невідповідність існуючих засобів навчання вимогам навчального процесу та низький рівень його методичного забезпечення. Однією із причин цього є недостатня відповідність інформаційно-імітаційної моделі (ІМ) підготовки операторів вимогам практики.

Враховуючи визначені у [10] недоліки та спираючись на [2–8], запропоновано для підготовки операторів постів РМ використовувати цільові програмні тренажери засобів РМ, в основу яких покладено ІМ підготовки. Науковий підхід щодо обґрунтування необхідності та створення даної моделі підготовки викладено у [2].

Спираючись на результати проведеного аналізу сформульовано **мету статті**, що полягає у розробці методологічного підходу програмної реалізації ІМ підготовки операторів складних ергатичних систем. Для досягнення поставленої мети введено обмеження, що дана модель розроблена для посту РМ на базі радіоприймального пристрою (РПрП) короткохвильового діапазону.

Викладення основного матеріалу. Програмна реалізація ІМ підготовки операторів постів РМ по суті є спеціалізоване програмне забезпечення (СПЗ), що призначене для забезпечення навчання операторів основам управління РПрП, а також для контролю рівня знань, умінь та навичок роботи на посту.

Узагальнений алгоритм СПЗ розділено на такі основні складові:

1. Навчання операторів за допомогою навчальних алгоритмів.
2. Самостійна робота з метою закріплення теоретичних знань та практичних навичок, набутих при навчанні.
3. Контроль дій оператора по підготовці пристрою до роботи, встановленню органів управління у вихідне положення, визначення режимів роботи, контролю функціонування в даних режимах.

Схема узагальненого алгоритму розробленого програмного забезпечення зображена на рисунку 1.

Розглянемо блоки даного алгоритму більш детально.

Блок 1. Позначає перехід до підпрограми навчання, в якій реалізовано навчальні алгоритми отримання навичок ручного управління приймачем.

Блок 2. Операція перевірки логічного виразу – чи пройшов оператор навчання. Якщо “Так” – то перехід до самостійної роботи з РПрП, якщо “Ні” – повернення до повторного відпрацювання попереднього алгоритму навчання.

Блок 3. Самостійне виконання раніше засвоєних навичок та умінь, без втручання викладача та без допомоги навчальних алгоритмів. Алгоритм самостійної роботи призначений для відпрацювання всіх набутих навичок в режимі вільного користування програмою “Робота з приймачем”.

Блок 4. При самостійній роботі з програмою без системи підказок оператор фактично здійснює самоконтроль рівня отриманих навичок та умінь. Якщо “Так” – перехід до контролю, якщо “Ні” – повернення до повторного відпрацювання алгоритмів навчання або до самостійної роботи зі СПЗ.

Блок 5. Контроль навичок та умінь операторів. З цією метою викладачем задається завдання на виконання певних практичних операцій щодо управління приймачем. Оцінювання здійснюється безпосередньо викладачем.

Блок 6. Визначення рівня засвоєння оператором теоретичних знань та практичних навичок. Якщо рівень достатній (результат “ТАК”) – перехід до роботи на реальній апаратурі, якщо “НІ” – повернення до повторного відпрацювання алгоритмів навчання.

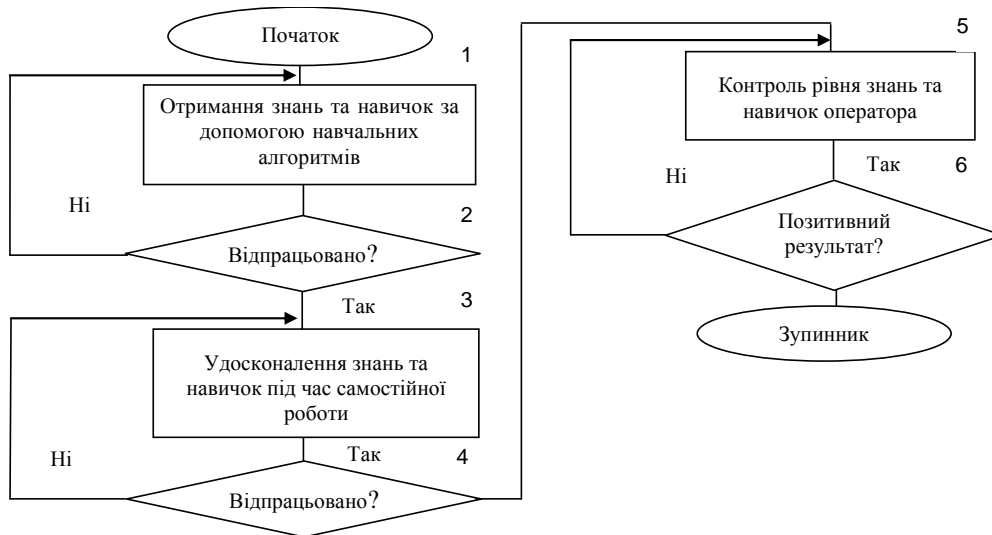


Рис. 1. Схема узагальненого алгоритму СПЗ підготовки операторів постів РМ

При розробці СПЗ підготовки операторів постів РМ на базі РПрП доцільно використовувати метод імітаційного комп'ютерного моделювання. Програма імітує зовнішній вигляд і функціональність радіоприймача, всіх його органів управління та відображення.

Виконується це таким чином: змодельований приймальний пристрій на окремій формі. Функціональність пристрою моделюється так, щоб зміну органу управління можна було б здійснювати за допомогою маніпулятора типу “миша” або з використанням іплікатора сенсорного екрана. Для моделювання зовнішнього вигляду всі статистичні елементи передньої панелі блоку чи пристрою виконуються у графічному редакторі і зберігаються в файлі, який завантажуватиметься в об’єкт Image та відобразатиметься на компоненті JPanel1. Таким чином, формується фон або задній план, а органи управління або динамічні елементи передніх панелей перерисовуються програмним шляхом на компонентах JPanel1 та JPanel2. Для цього записано потрібний програмний код у відповідних функціях обробки подій цих компонентів.

Для створення інтерфейсу програми та деяких складових частин використано програмний пакет SWING, який є бібліотекою машинно-незалежних класів для створення інтерфейсу користувача. Це дозволяє:

- використовувати більш багатий і зручний набір елементів інтерфейсу користувача;
- створювати програмне забезпечення, яке практично не залежне від платформи, на якій буде виконуватись програма, відповідно зменшується ймовірність виникнення помилок, що характерні для конкретної платформи;

– забезпечувати програміста і користувача однотипними засобами для роботи на різних платформах.

Структурно СПЗ складається з таких модулів:

ImageTestApp.java. Модуль форми головного (диспетчерського) вікна програми, призначений для ініціалізації вікна верхнього рівня (фрейму), підключення програмних класів та запуску в складі програми.

ImageTestView.java. Головний клас програми, призначений для виводу зображення РПрП (статичних та динамічних об’єктів), ініціалізації графічних та звукових ресурсів програми, сприйняття дій користувача та відповідної зміни параметрів. Модуль взаємодіє з усіма модулями-класами, що належать до складу програми.

Mu_Mixer.java. Клас, що призначений для низькорівневої обробки аудіофайлів імітування радіограм радіофіру в форматі WAV PCM у режимі прямого доступу до аудіобуфера. Забезпечує засоби відкриття файлу, його програшу, регулювання гучності та надає доступ до одnobайтового масиву звукових вибірок.

Nois.java. Клас, що призначений для низькорівневої обробки аудіофайлів імітування шуму та завад радіофіру в форматі WAV PCM у режимі прямого доступу до аудіобуфера.

Knopka.java. Клас, що призначений для реалізації алгоритму обробки події натиснення на програмні кнопки.

Main_menu.java. Клас, що призначений для реалізації алгоритму імітування роботи програмного меню радіоприймального пристрою.

Ruchka.java. Клас, що призначений для реалізації алгоритму обробки події крутіння програмної ручки в режимі “Робота з приймачем”.

Ruchka_study.java. Клас, що призначений для реалізації алгоритму обробки події обертання програмної ручки в режимі “Навчання”.

Spectr.java. Клас, що призначений для реалізації алгоритму імітування роботи рідинно-кристалічного дисплею (РКД) приймача: його відображення та динамічної зміни.

Spectr_MRK.java. Клас, що призначений для реалізації алгоритму імітування відображення на РКД спектрограми прийнятого сигналу.

ImageTestAboutBox.java. Клас, що призначений для ініціалізації та відображення програмного довідкового вікна.

Взаємозв’язок між програмними модулями зображено на рисунку 2.

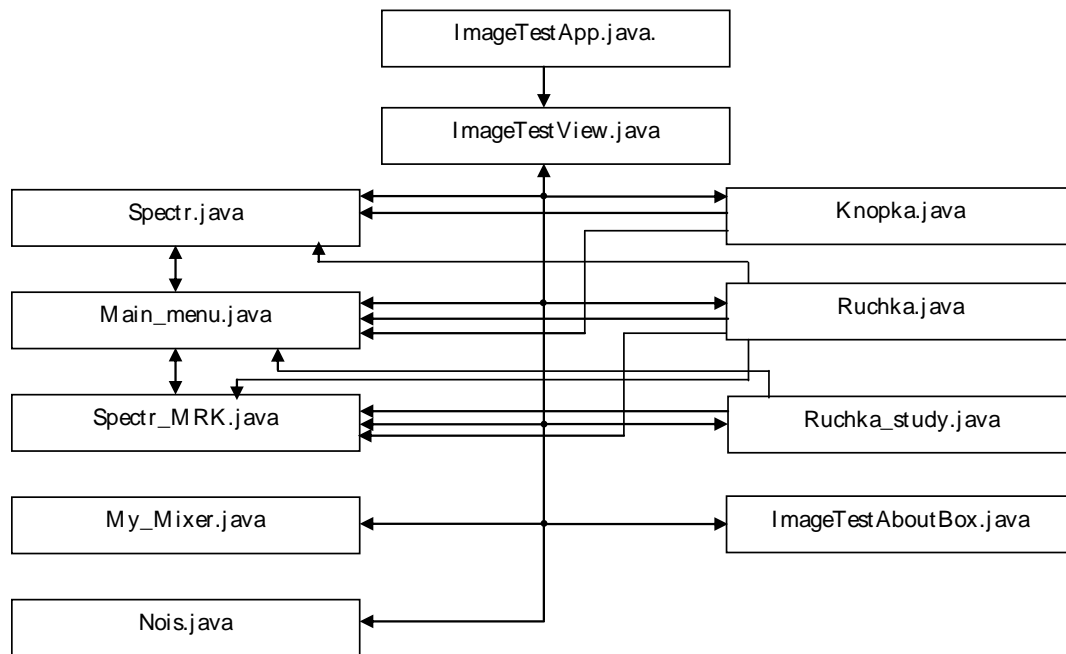


Рис. 2. Взаємозв’язок класів спеціалізованого програмного забезпечення

Розглянемо дані модулі більш детально.

Після запуску файла програми Proot.jar (java-архів) здійснюється ініціалізація фрейму та встановлюються взаємозв’язки з програмними класами, що належать до складу СПЗ. Безпосередньо запуск програми здійснюється після виконання інструкції launch (ImageTestApp.class, args), яка міститься у функції main класу ImageTestApp.java. Після чого здійснюється виконання алгоритмів, що належать до класу ImageTestView.java.

Виконанню модуля ImageTestView.java передусє ініціалізація змінних і констант, якими оперує розроблена програма. Після цього здійснюється ініціалізація та відображення всіх графічних компонентів програмного інтерфейсу шляхом виконання інструкції initComponents().

Вибір користувачем режиму функціонування забезпечується виконанням алгоритму обробки головного меню програми, що описаний функціями jMenuItemActionPerformed та jMenuItem2ActionPerformed головного класу програми.

Відображення динамічних елементів передньої панелі приймача здійснюється програмним шляхом, тобто використовуючи стандартні команди побудови геометричних фігур. Наприклад, для відображення програмної кнопки заданого розміру (44×44 пікселя) із заданими координатами ($x = 727$; $y = 58$) необхідно виконати такі команди:

```

g=jPanel1.getGraphics(); // отримання графічного специфікатора;
Graphics2D g2=(Graphics2D)g; // креслення двомірних фігур;
g.setColor(Color.GRAY); // встановлення кольору об’єкта;
Ellipse2D kn1=new Ellipse2D.Double(727, 58, 44, 44); // виклик класу еліпс;
g2.draw(kn1); // креслення еліпса (кола).
  
```

Імітування натискання програмної кнопки здійснюється за рахунок короточасної зміни забарвлення ореола кнопки на червоний, а після відпускання лівої кнопки “миши” колір ореола набуває початкового значення. Далі, відповідно до призначення кнопки, викликається та чи інша процедура обробки події.

Для відображення програмних регуляторів використовуються стандартні команди побудови геометричних фігур, а саме: використовуючи клас `Ellipse2D`, задаються відповідні параметри для кола. Для створення ефекту обертання на фоні регулятора окреслюється коло меншого діаметра та відмінного кольору. При цьому коловий рух малого кола здійснюється так, що координати його не виходять за межі регулятора (кола з більшим радіусом). Наступні координати кола, що обертається, розраховуються за формулами:

$$X = r \cos \varphi + x; \quad (1)$$


$$Y = r \sin \varphi + y, \quad (2)$$

де X , Y – наступні координати об’єкта; r – радіус об’єкта; φ – кут, на який здійснюється зміщення центра кола; x , y – координати, відносно яких здійснюється обертання.




У разі обертання користувачем колеса прокрутки маніпулятора “миша” програмно генерується подія `java.awt.event.MouseWheelEvent` і відпрацьовується алгоритм за таким програмним кодом:

```
{if(evt.getWheelRotation())>0) // якщо обертання колеса догори:
```

```
{
    g.setColor(Color.DARK_GRAY);
    g.fillOval(x2_2,y2_2,50,50);
    g.setColor(Color.BLACK);
    int r=50;
    fi2_2=fi2_2+0.1;
    x2_2=(int) Math.round(r*Math.cos(fi2_2))+415;
    y2_2=(int) Math.round(r*Math.sin(fi2_2))+460;
    g.fillOval(x2_2,y2_2,50,50);
}else // якщо обертання колеса донизу
{
    g.setColor(Color.DARK_GRAY);
    g.fillOval(x2_2,y2_2,50,50);
    g.setColor(Color.BLACK);
    int r=50;
    fi2_2=fi2_2-0.1;
    x2_2=(int) Math.round(r*Math.cos(fi2_2))+415;
    y2_2=(int) Math.round(r*Math.sin(fi2_2))+460;
    g.fillOval(x2_2,y2_2,50,50);
}
}
```

Згідно з Керівництвом із застосування РПрП, управління функціями приймача можна здійснювати через системне меню, виклик якого здійснюється натисненням програмної кнопки . Особливостями системного меню РПрП є:

– розгалужена система меню;

– навігація здійснюється обмеженою кількістю програмних кнопок (, , , .

З метою програмної реалізації алгоритму імітування роботи РКД в режимі MENU застосовано технологію умовних переходів `switch-case`. При цьому деякі елементи меню передбачають структуру виду (`Main_menu.java`):

```
switch(...)
{case 0:
    {switch(...)
        { case 0:
            .....
            break;}
        break; } }
```


Для відображення РКД використовуються стандартні команди побудови геометричних фігур. Алгоритм графічного відображення РКД та його динамічної зміни реалізовано у класах `Spectr.java` та `Spectr_MRK.java`. При цьому у класі `Spectr.java` реалізовано алгоритм імітування відображення осцилограми прийнятого сигналу:

```
int fs=7000;
for(x_sp=130;x_sp<410;x_sp++)
```

```

    {
        y_sp=(float)(Math.cos(2*Math.PI*fs*i_sp)+Math.cos(2*Math.PI*110*fs*i_sp1)+
        Math.cos(2*Math.PI*50*fs*i_sp2))*13;
        g2.setColor(Color.GREEN);
        line.setLine(x_sp, y_sp+200, x_sp, (Math.round(y_sp)+200));
        g2.draw(line);
        i_sp=(float) (i_sp + 0,00000003);//0,00000003
        i_sp1=(float) (i_sp1 + 0,00000003);
        i_sp2=(float) (i_sp2 + 0,00000003);
    }

```

У класі Spectr_MRK.java реалізовано алгоритм імітування відображення спектрограми прийнятого сигналу та маркерів у режимі вмикання маркера. Для відображення на характеристиці сигналу маркерів натискають кнопку .

Імітування побудови спектрограми сигналу здійснено з використанням методу швидкого перетворення Фур'є.

Слід зазначити, що реалізація алгоритмів імітування відображення спектрограми та осцилограми здійснюється в окремих потоках.

Для імітування радіоефіру в процесі роботи радіоприймального пристрою у класах Nois.java та My_Mixer.java реалізовано алгоритми звукового супроводження. Для цього використано багатопоточний підхід, що дозволило розподілити ресурс оперативної пам'яті ПЕОМ. При цьому бітовий потік, що зчитується із звукового файлу (ПРУТ\Proot\Sound*.wav) розглядається як об'єкт типу Clip. Залежно від частоти, на яку настроюється програмний імітатор, завантажується той чи інший звуковий файл. Запуск файлів імітування шуму та корисного сигналу ефіру реалізовано шляхом одночасного їх запуску в різні потоки. Наступний фрагмент коду ілюструє алгоритм їх запуску:

```
clip.start(); while(!Nois_Break) {clip.loop(count_loop);}
```

Переведення СПЗ в режим "Навчання" здійснюється шляхом вибору опції "Робота з приймачем" → "Навчання". При цьому відпрацьовується алгоритм, що реалізовано у функції jMenuItem2ActionPerformed класу ImageTestView.java. З цією метою на компонент jPanel2 завантажується зменшене зображення передньої панелі РПрП String filename="Proot/src/imagetest/resources/morda3.png". Графічні динамічні елементи передньої панелі програмно прорисовуються окремими стандартними геометричними фігурами. Для відображення програмних кнопок у режимі навчання використовуються алгоритми, що реалізовані у функції кнопка_press_study класу Кнопка.java, а програмні регулятори реалізовані в окремому класі Ruchka_study.java. В процесі функціонування програми в даному режимі до текстового поля графічного компонента JTextArea завантажуються відповідні підказки у вигляді текстових повідомлень.

Розглянемо принцип експлуатації СПЗ за призначенням. Після вмикання ПЕОМ та завантаження операційної системи необхідно запустити на виконання файл "Proot.jar". Звернення до програми відбувається за допомогою активних вікон.

Після запуску програми розгортається початкове вікно. Для вибору режиму роботи СПЗ необхідно увійти до опції головного меню "Робота з приймачем" (рис. 3).

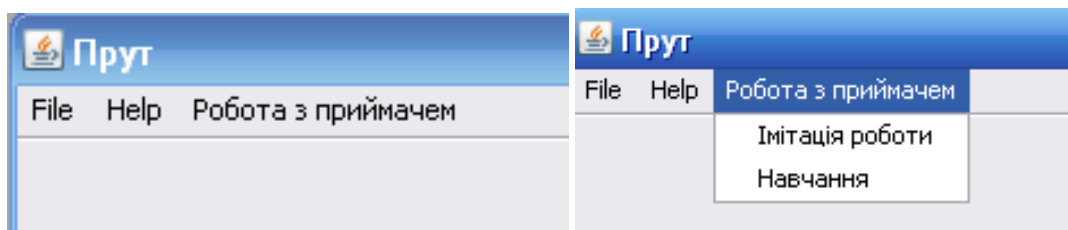


Рис. 3. Головне меню програми

При виборі опції "Робота з приймачем" → "Імітація роботи" на екранну форму завантажується зображення передньої панелі приймача "ВОСХОД" (файл \ПРУТ\Proot\src\imagetest\resources\ morda1.png) (рис. 4).

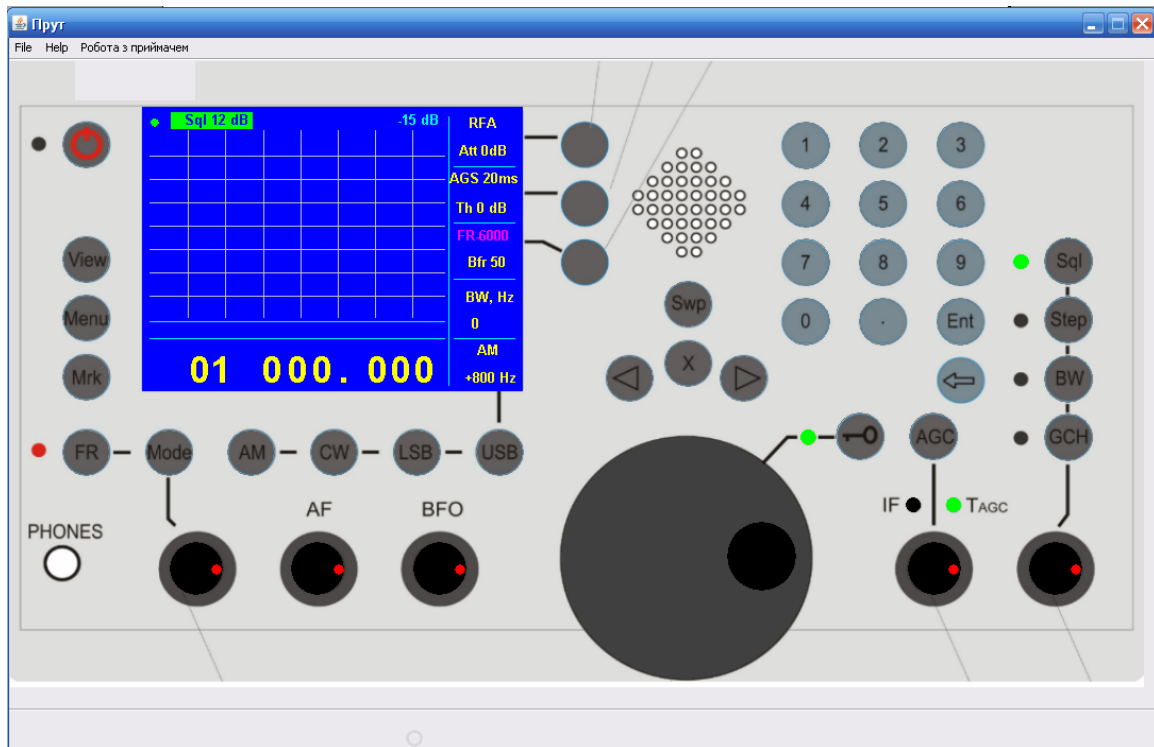



Рис. 4. Вікно програми в режимі "Імітація роботи"

Для початку імітування роботи приймача необхідно натиснути програмну кнопку . При цьому завантажуються відповідно до частоти настройки медійний файл (`\ПРУТ\Proot\Sound*.wav`). Процедури обробки звукових файлів знаходяться у класах `\ПРУТ\Proot\build\classes\imagetest\My_Mixer.class` та `Nois.class`.

Керування програмною моделлю здійснюється маніпулятором "миша" в момент, коли курсор знаходиться у межах екранної області об'єкта. Якщо курсор знаходиться над програмною кнопкою, то активізація здійснюється шляхом натискання лівої кнопки маніпулятора "миша" (відповідні процедури описані у класі `\ПРУТ\Proot\build\classes\imagetest\Кнопка.class`). Якщо курсор знаходиться над програмною ручкою керування здійснюється шляхом обертання ролика маніпулятора (відповідні процедури описані у класі `\ПРУТ\Proot\build\classes\imagetest\Ruchka.class`).

При виборі опції "Робота з приймачем" → "Навчання" на екранну форму завантажуються зображення передньої панелі РПрП (файл `\ПРУТ\Proot\src\imagetest\resources\morda3.png`) (рис. 5).

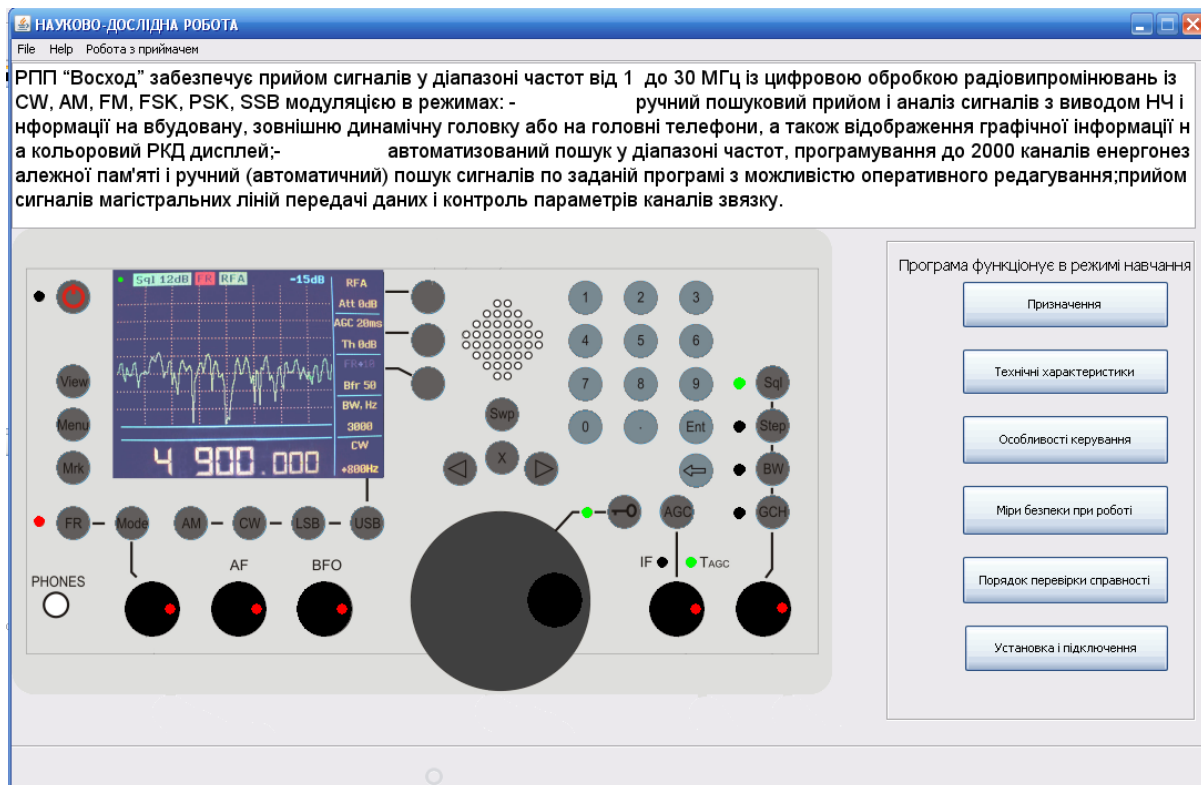


Рис. 5. Вікно програми в режимі "Навчання"

В режимі "Навчання" при натисканні програмних кнопок (крутіння програмних ручок) до інформаційного поля автоматично виводиться відповідне інформаційне повідомлення. Логіка процедур управління органами функціонування подібні до режиму "Імітація роботи" та описані у класах `Ruchka_study.class` та `Knopka.class`.

Під час завантаження програми вхідні дані можна поділити за такими критеріями:

- елементи інтерфейсу програми;
- звукові файли із записами роботи працюючих радіостанцій;
- дані, що містяться в ініціалізаційному файлі (системному реєстрі операційної системи);
- файл даних, що характеризує модель радіоелектронної обстановки.

Елементи інтерфейсу програми – це растрові зображення передніх панелей, перемикачів, шкал, ручок та інших елементів управління, які зберігаються у графічних файлах формату BMP або JPG, або PNG. Ці зображення завантажуються у конструкторах форм в пам'ять, а при виконанні програми копіюються на форму.

У ініціалізаційному файлі (системному реєстрі) містяться дані щодо відпрацювання звукових ефектів, а також початкові установки програми.

У файлі, що характеризує модель радіоелектронної обстановки, зберігаються значення частот, які потім завантажуються в асоціативний контейнер, при цьому відповідним чином сортуючись з метою їх найшвидшого пошуку.

Під час роботи програми, відпрацювання оператором того чи іншого навчального питання вхідними даними є безпосередньо дії оператора, які запам'ятовуються відповідно до алгоритму контролю дій оператора. Фіксація вводу або виконання тієї чи іншої дії оператора відбувається у відповідній функції обробки події того компонента або елемента управління, над яким виконувалась ця дія. При цьому кожній дії присвоєно свій ідентифікаційний номер або дескриптор дії. Проте ці вхідні дані характерні лише для контролюючої частини програмного продукту.

Використання програми забезпечує отримання наступних вихідних даних процесу імітування роботи радіоприймального пристрою. Що може бути представлений у формі:

- графічний – шляхом побудови осцилограм імітованих сигналів, програмного прорисовування зміни станів функціональних органів управління (кнопок, ручок, індикаторів тощо);
- медійний – шляхом відтворення звукових файлів (`(\ПР\УТ\Прот\Sound*.wav)`);
- текстовий – виводом текстових інформаційних повідомлень.

Висновки. Програмна реалізація інформаційно-імітаційної моделі підготовки операторів постів радіомоніторингу здатна забезпечити автоматизацію процесу навчання без використання реальної

апаратури. При цьому винятком є етап доведення умінь та навичок роботи з приймачем до автоматизму, оскільки він вимагає набуття моторно-рухових навичок.

Отриманий науково-практичний результат має стати підґрунтям для подальших досліджень у напрямку моделювання та алгоритмізації процесу підготовки чергової зміни приймального центру (комплексу) радіочастотного моніторингу.

Список використаної літератури:

1. Державне підприємство “Український державний центр радіочастот”. Офіційне інтернет-представництво [Електронний ресурс]. – Режим доступу : <http://www.ucrf.gov.ua/>.
2. Інформаційно-імітаційна модель підготовки операторів постів радіочастотного моніторингу / М.А. Роговець, Р.В. Дзюбчук, О.Ю. Бусько, І.І. Опанасюк // Проблеми створення, випробування, застосування та експлуатації складних інформаційних систем : зб. наук. пр. – Житомир : ЖВІ НАУ, 2011. — Вип. 4. — С. 78–86.
3. Ягунов В.В. Теорія і методика військового навчання : монографія / В.В. Ягунов. — К. : Тандем, 2000. — 380 с.
4. Балановський П.К. Використання ЕОМ для практичної підготовки фахівців зв'язку / П.К. Балановський, А.П. Глушко, С.А. Литвин // Військова освіта : зб. наук. пр. – Харків : Основа, 2000. — № 8. — С. 220–223.
5. Дзюбчук Р.В. Алгоритм підготовки операторів постів радіомоніторингу до роботи на приймальному пристрої AR 5000 А з використанням програми імітатора / Р.В. Дзюбчук // Системи обробки інформації. – Харків, 2008. – Вип. 3 (70). – С. 39–43.
6. Тренажерно-імітаційний комплекс на базі ЕОМ / Ю.М. Башкиров, І.В. Єременко, Н.А. Стасевич, А.І. Бобунов // Військова освіта : зб. наук. пр. — Харків : Основа, 2000. — № 8. — С. 80–82.
7. Чабаненко П.П. Моделювання навчання операторів систем озброєння і військової техніки / П.П. Чабаненко. – К. : Науково-методичний центр військової освіти МО України, 2003. – № 11. – С. 165–173.
8. Романов А.Н. Тренажеры для подготовки операторов РЛС с помощью ЭВМ / А.Н. Романов. – М. : Воениздат, 1980. – 126 с.
9. Дзюбчук Р.В. Використання програм-імітаторів для вирішення проблем практичної підготовки спеціалістів з експлуатації складних інформаційно-технічних систем / Р.В. Дзюбчук // Кібернетика і системний аналіз : зб. наук. пр. – Харків : ХУПС, 2008. – Вип. 3 (18). – С. 74–78.
10. Руснак І.С. Проблеми модернізації та створення тренажно-моделювальних комплексів військового призначення / І.С. Руснак, В.Л. Шевченко // Наука і оборона. – 2002. – № 1. – С. 32–39.
11. Матвієвський О.М. Методичний підхід до обґрунтування характеристик тренажних засобів і систем / О.М. Матвієвський, О.В. Герасименко, Ю.М. Щєбланін // Наука і оборона. – 2005. – № 1. – С. 59–65.

РОГОВЕЦЬ Максим Анатолійович – кандидат технічних наук, доцент кафедри Житомирського військового інституту ім. С.П. Корольова Національного авіаційного університету.

Наукові інтереси:

– інформаційні системи спеціального призначення.

Тел.: (067)225–51–18.

E-mail: mrogovets@mail.ru

Стаття надійшла до редакції 25.06.2012

Роговиць М.А. Програмна реалізація інформаційно-імітаційної моделі підготовки операторів постів радіочастотного моніторингу

Роговец М.А. Программная реализация информационно-имитационной модели подготовки операторов постов радиочастотного мониторинга

Rogovets M.A. Programmatic realization to informatively-imitation models for preparation of the radiofrequency monitoring posts operators

УДК 378.169.3:621.396.62

Программная реализация информационно-имитационной модели подготовки операторов постов радиочастотного мониторинга / М.А. Роговец

В статье предложен методологический подход относительно программной реализации информационно-имитационной модели подготовки операторов сложных эрготических систем с целью автоматизации процесса получения ими практических навыков работы на аппаратуре.

УДК 378.169.3:621.396.62

Programmatic realization to informatively-imitation models for preparation of the radiofrequency monitoring posts operators / M.A. Rogovets

In the article methodological approach is offered in relation to programmatic realization to the informatively-imitation models of the complex ergotice systems operators preparation with the purpose of process automation for reception of practical skills to work on an apparatus by them.