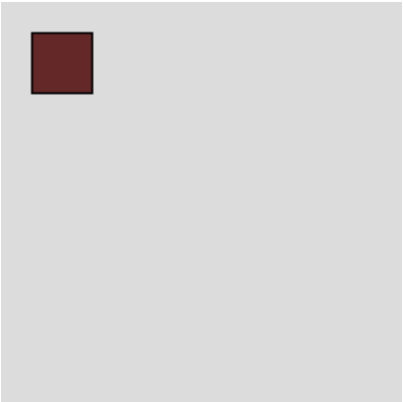
Під час проведення практичних робіт з курсу «Креативне програмування» пропонується наступна методика, яка передбачає наступний ланцюжок дій і переходів : Базова програма → Компетентнісна програма → Варіативна програма → Креативна (авторська) програма.

Базові програми –це програми, які включають у себе приклади коду, що вивчався раніше і на основі яких будуть створюватися інші (більш складніші) програми. Базові програми можуть надаватися вчителем у готовому вигляді (з коротким інструктажем) або створюватися учнями. Вивчений на попередніх уроках навчальний матеріал і набуті компетентності повинні дозволити здобувачам освіти написати базові програми самостійно. Обраний варіант залежить від завдань котрі ставить вчитель: перевірка домашнього завдання, оцінювання, актуалізація опорних знань, мотивація навчальної діяльності тощо.

Наприклад.

<table>
<tr>
<td>  </td>
<td>

```
void setup(){size(200,200);background(220);}

void draw()
{
int a=30;
int c=15;
int x=c;
int y=c;
fill(100,40,40);
stroke(0);
strokeWeight(1);
rect(x,y,a,a);
}
```

</td>
</tr>
</table>

Компетентнісні програми створюються на основі базових шляхом заміни частини коду або додавання фрагментів коду відповідно до діяльнісної складової навчальної програми для набуттям відповідних компетентностей. Дані програми створюються здобувачами освіти під керівництвом вчителя, шляхом послідовних перетворень від простого до складного з відповідними коментарями.

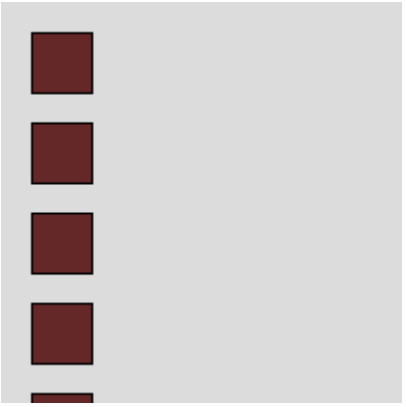
Наприклад.

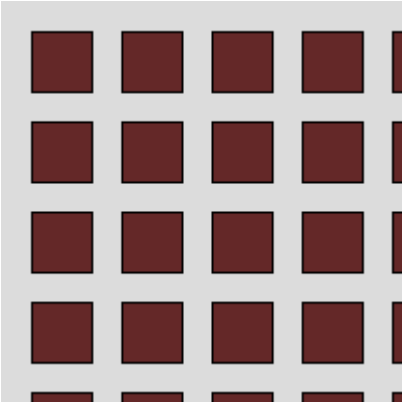
<table>
<tr>
<td></td>
<td>

```
void setup(){size(200,200);background(220);}
```

</td>
</tr>
</table>

	`void draw()`
	`{`
	`int a=30;`
	`int c=15;`
	`int y=c;`
	`fill(100,40,40);`
	`stroke(0);`
	`strokeWeight(1);`
	`for(int x=c;x<200;x=x+a+c){`
	`rect(x,y,a,a);}`
	`saveFrame("fere.bmp");`
	`}`

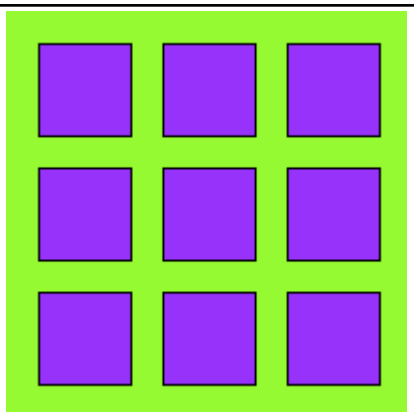
	`void setup(){size(200,200);background(220);}`
	
	`void draw()`
	`{`
	`int a=30;`
	`int c=15;`
	`int x=c;`
	`fill(100,40,40);`
	`stroke(0);`
	`strokeWeight(1);`
	`for(int y=c;y<200;y=y+a+c){`
	`rect(x,y,a,a);}`
	`saveFrame("fere.bmp");`
	`}`

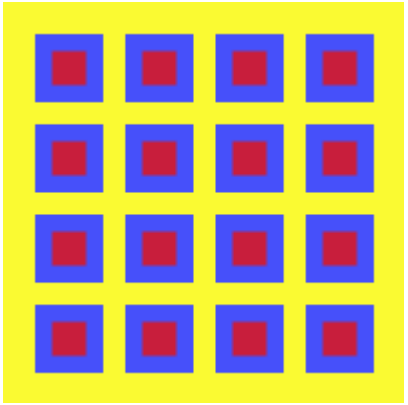
	`void setup(){size(200,200);background(220);}`
	
	`void draw()`
	`{`
	`int a=30;`
	`int c=15;`
	`fill(100,40,40);`
	`stroke(0);`
	`strokeWeight(1);`
	`for(int y=c;y<200;y=y+a+c){`
	`  for(int x=c;x<200;x=x+a+c){`
	`  rect(x,y,a,a);}}`
	`saveFrame("fere.bmp");`
	`}`

Варіативні програми – програми, які створюються здобувачами освіти самостійно на основі компетентнісних шляхом зміни кількісних характеристик (величин) з наступним виконанням для утворення потрібного результату - створення малюнка.

Наприклад.

	`void setup(){size(200,200);background(150,250,50);}`

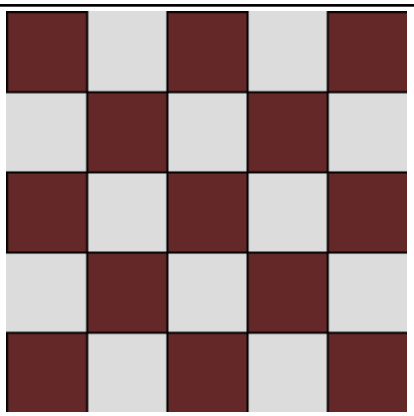
	`void draw()`
	`{`
	`int a=46;`
	`int c=16;`
	`fill(150,50,250);`
	`stroke(0);`
	`strokeWeight(1);`
	`for(int y=c;y<200;y=y+a+c){`
	`  for(int x=c;x<200;x=x+a+c){`
	`  rect(x,y,a,a);}}`
	`saveFrame("fere.bmp");`
	`}`

	`void`
	`setup(){size(200,200);background(250,250,50);}`
	
	`void draw()`
	`{`
	`int a=25;`
	`int c=20;`
	`fill(200,30,60);`
	`stroke(70,80,250);`
	`strokeWeight(8);`
	`for(int y=c;y<200;y=y+a+c){`
	`  for(int x=c;x<200;x=x+a+c){`
	`  rect(x,y,a,a);}}`
	`saveFrame("fere.bmp");`
	`}`

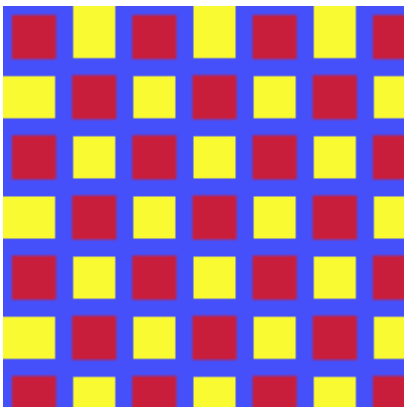
Креативні програми – це нові (новостворені) програми, або модифіковані програми (програми у яких є якісні зміни) створюються здобувачами освіти самостійно під час проектної діяльності, практикуму тощо. Учитель виступає у якості коуча і може надавати лише певні консультації (за потреби або за проханням учня). В даному випадку надається можливість проявити свою фантазію, творчість, креатив. При виникненні труднощів з творчим компонентом учитель може надати усний опис програми яку потрібно створити або надати кінцевий результат виконання програми (малюнок) для створення відповідного коду.

Наприклад.

	`void setup(){size(200,200);background(220);}`
	
	`void draw()`
	`{`
	`int a=40;`
	`fill(100,40,40);`
	`stroke(0);`

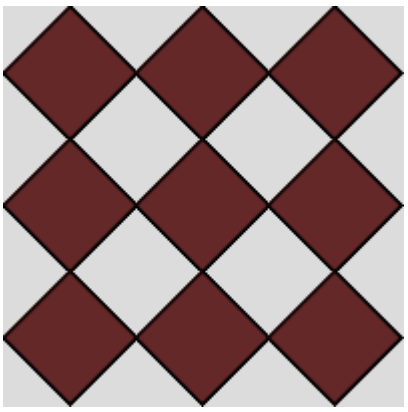


```
strokeWeight(1);
for(int y=0;y<200;y=y+2*a){
  for(int x=0;x<200;x=x+2*a){
        rect(x,y,a,a);
          rect(x+a,y+a,a,a);}}
saveFrame("fere.bmp");
}
```



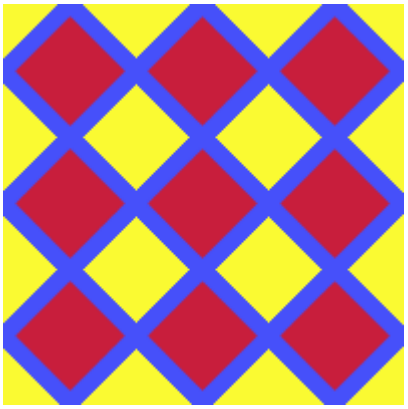
```
void
setup(){size(200,200);background(250,250,50);}

void draw()
{
int a=30;
fill(200,30,60);
stroke(70,80,250);
strokeWeight(8);
for(int y=0;y<200;y=y+2*a){
  for(int x=0;x<200;x=x+2*a){
        rect(x,y,a,a);
          rect(x+a,y+a,a,a);}}
saveFrame("fere.bmp");
}
```



```
void setup(){size(200,200);background(220);}

void draw()
{
int a=33;
fill(100,40,40);
stroke(0);
strokeWeight(1);
for(int y=a;y<200;y=y+2*a){
  for(int x=a;x<200;x=x+2*a){
       quad(x-a,y,x,y-a,x+a,y,x,y+a);}}
saveFrame("fere.bmp");
}
```



```
void
setup(){size(200,200);background(250,250,50);}

void draw()
{
int a=33;
fill(200,30,60);
stroke(70,80,250);
strokeWeight(8);
for(int y=a;y<200;y=y+2*a){
  for(int x=a;x<200;x=x+2*a){
       quad(x-a,y,x,y-a,x+a,y,x,y+a);}}
saveFrame("fere.bmp");
}
```

Додатковими видами діяльності учнів котрі виконали поставленне завдання швидше за інших і дозволяють масштабувати урок можуть бути оптимізація коду та дописування коментарів. Автор намагався писати програмний код таким чином, щоб зробити код максимально читабельним та зрозумілим. Програмний код можна і потрібно оптимізовувати. Учитель може ознайомити учнів зі способами і методами оптимізації і надавати відповідну допомогу.

// Програма із звичайним кодом
void setup()
{size(200,200);
background(250,250,50);}
 	
void draw()
{int a=25;
int c=20;
fill(200,30,60);
stroke(70,80,250);
strokeWeight(8);
for(int y=c;y<200;y=y+a+c){
  for(int x=c;x<200;x=x+a+c){
  rect(x,y,a,a);}}
}
	// Програма із оптимізованим кодом
	void setup()
	{size(200,200);
	background**(#FAFA32);**}
	
	void draw()
	{**int a=25,c=20,k=8;**
	fill**(#C81E3C);**
	stroke**(#4650FA);**
	strokeWeight**(k);**
	for(int y=c;y<**width**;y=y+a+c){
	  for(int x=c;x<**height**;x=x+a+c){
	  rect(x,y,a,a);}}
	noLoop();
	}

Рекомендується доповнення програми коментарями, оскільки це є важливим компонентом перевірки та повторення знаннєвої складової навчальної програми.

```
void setup()
{size(200,200);//задаємо розміри вікна
background(250,250,50);}//задаємо колір тла(фону)
void draw()
{int a=25;//задаємо ширину квадрату
int c=20;//задаємо відстань між квадратами
int k=8;//задаємо ширину лінії квадрату
fill(200,30,60);//задаємо колір заливка квадрату
stroke(70,80,250);//задаємо колір лінії квадрату
strokeWeight(8);//задаємо ширину лінії квадрату
for(int y=c;y<200;y=y+a+c){// команда повторення-малюємо квадрати по
вертикалі
  for(int x=c;x<200;x=x+a+c){// команда повторення-малюємо квадрати по
горизонталі
rect(x,y,a,a);}}//малюємо квадрат
}
```

Додаток_2

Програма із звичайним кодом

```
void setup()
{size(200,200);
background(250,250,50);}

void draw()
{int a=25;
int c=20;
fill(200,30,60);
stroke(70,80,250);
strokeWeight(8);
for(int y=c;y<200;y=y+a+c){
  for(int x=c;x<200;x=x+a+c){
        rect(x,y,a,a);}}
}
```

Програма із оптимізованим кодом

```
void setup()
{size(200,200);
background(#FAFA32);}

void draw()
{int a=25,c=20,k=8;
fill(#C81E3C);
stroke(#4650FA);
strokeWeight(k);
for(int y=c;y<width;y=y+a+c){
  for(int x=c;x<height;x=x+a+c){
        rect(x,y,a,a);}}
}
```

Програма з коментарями

```
void setup()
{size(200,200);//задаємо розміри вікна
background(250,250,50);}//задаємо колір тла(фону)

void draw()
{int a=25;//задаємо ширину квадрату
int c=20;//задаємо відстань між квадратами
int k=8;//задаємо ширину лінії квадрату
fill(200,30,60);//задаємо колір заливка квадрату
stroke(70,80,250);//задаємо колір лінії квадрату
```

```
strokeWeight(8);//задаємо ширину лінії квадрату
for(int y=c;y<200;y=y+a+c){// команда повторення-малюємо квадрати по
вертикалі
  for(int x=c;x<200;x=x+a+c){// команда повторення-малюємо квадрати по
горизонталі
rect(x,y,a,a);}}//малюємо квадрат
}
```

Додаток_3