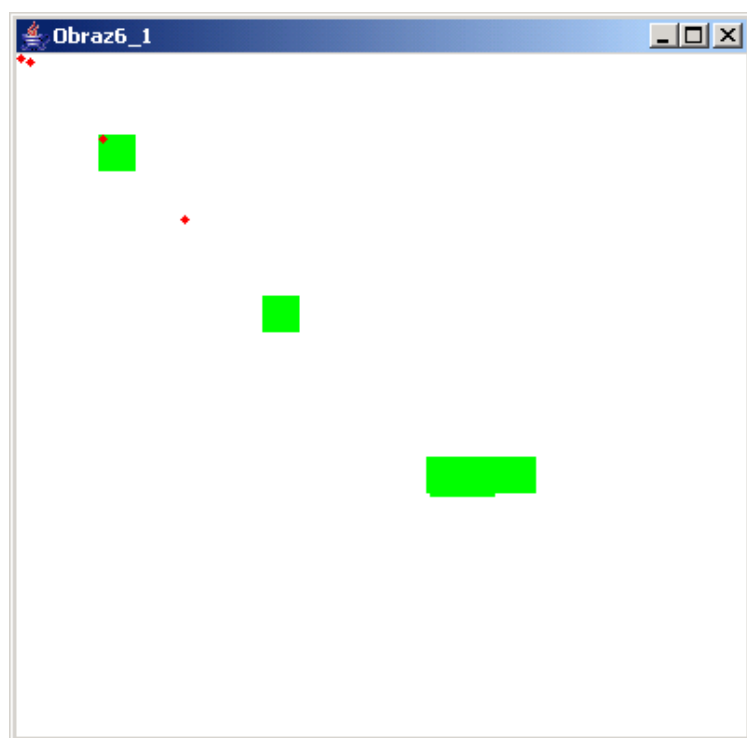


## LABORATORIUM 4

**Cel:** Budowa klas z wykorzystaniem dziedziczenia oraz implementowania interfejsów. Wprowadzenie do trybu graficznego. Animacje obiektów w trybie graficznym – obsługa zdarzeń od klawiatury (*KeyListener*), myszy (*MouseListener*) oraz uaktywniania widoku (*ActionListener*). Należy wykonać zadanie 2 oraz 3 lub 4 – ocena 4. Na ocenę 5 lub 5.5 należy dodatkowo wykonać zadanie 5.

1. Wykonaj program (klasa *Obraz\_6\_1*), który wyświetla na ekranie dane figury, typu *Punkt*, *Kwadrat* tak, jak program *Obraz6\_1* (projekt *Rys6\_1*) oraz dodatkowo dane prostokąty. Klasa *Prostokat* powinna dziedziczyć od klasy *Kwadrat* (definicja konstruktora w klasie *Prostokat*) i dodatkowo powinna posiadać metody obsługujące drugi bok prostokąta (*szerokosc*). W programie *Obraz\_6\_1*, który ma teraz dodatkowo definicję klasy *Prostokat* w porównaniu z programem *Obraz6\_1*, należy w klasie *Figury* w metodzie *wypelnij()* dodać wstawianie obiektów typu *Prostokat* do tablicy *figury*. W programie można wybrać figurę, podając z klawiatury jej indeks w tablicy *figury* i następnie klawiszami strzałek przesuwać ją: w lewo, w prawo, w górę oraz w dół. Figury przesuują się, pozostawiając ślad. Po usunięciu komentarza w metodzie *paintComponent*, czyli uruchamianiu dziedziczonej metody *super.paintComponent* ślad po przesuwanej figurze jest usuwany (odświeżanie zawartości okna). W przypadku naciśnięcia klawisza *shift* można zmienić zakres losowanych wartości przesunięcia figury. Obsługa klawiatury jest możliwa po implementacji metod słuchacza klawiatury – *KeyListener*. Słuchaczami tego typu zdarzeń jest klasa *Figury*, dziedzicząca od *JPanel*.



```
C:\PROGRA~1\XINOX5~1\JCREAT~4\GE2001.exe
false, ze jestem kwadratem, bo jestem Punkt@126e85f ,X=0 , Y=0 , odleglosc=62.22539674441618
true, ze jestem kwadratem, bo jestem Kwadrat@8916a2 ,X=44 , Y=44 , odleglosc=0.0
false, ze jestem kwadratem, bo jestem Punkt@2ce908 ,X=44 , Y=44 , odleglosc=124.45079348883236
true, ze jestem kwadratem, bo jestem Kwadrat@77158a ,X=132 , Y=132 , odleglosc=62.22539674441618
false, ze jestem kwadratem, bo jestem Punkt@27391d ,X=88 , Y=88 , odleglosc=186.67619023324855
5
```

2. Wykonaj program (klasa *Obraz\_6\_2*), który wyświetla na ekranie dane figury typu *Punkt*, *Kwadrat* tak, jak program *Obraz6\_2* (projekt *Rys6\_2*) oraz dodatkowo dane prostokąty. Klasa *Prostokat* powinna dziedziczyć od klasy *Kwadrat* (definicja konstruktora w klasie *Prostokat*) i dodatkowo powinna posiadać metody obsługujące drugi bok prostokąta (*szerokosc*). W programie *Obraz\_6\_2*, który ma teraz dodatkowo definicję klasy *Prostokat* w porównaniu z programem *Obraz6\_2*, należy w klasie *Figury* w metodzie *wypelnij()* dodać wstawianie obiektów typu *Prostokat* do tablicy *figury*. W programie można wybrać figurę, podając z klawiatury jej indeks w tablicy *figury* i następnie klawiszami strzałek przesuwać ją: w lewo, w prawo, w górę oraz w dół. Figury przesuują się, pozostawiając ślad. Po usunięciu komentarza w metodzie *paintComponent*, czyli uruchamianiu dziedziczonej metody *super.paintComponent* ślad po przesuwanej figurze jest usuwany (odświeżanie zawartości okna). W przypadku naciśnięcia klawisza *shift* można zmienić zakres losowanych wartości przesunięcia figury. Obsługa klawiatury jest możliwa po implementacji metod słuchacza klawiatury – *KeyListener*. Można dodatkowo wybierać figurę do animacji za pomocą klikania myszą w wybraną figurę. Obsługa myszy jest możliwa po implementacji metod słuchacza klawiszy myszy – *MouseListener*. Słuchaczami obu typów zdarzeń jest klasa *Figury*, dziedzicząca od *JPanel*.

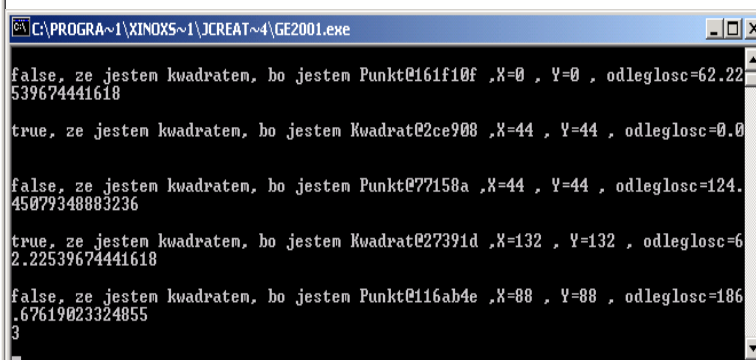
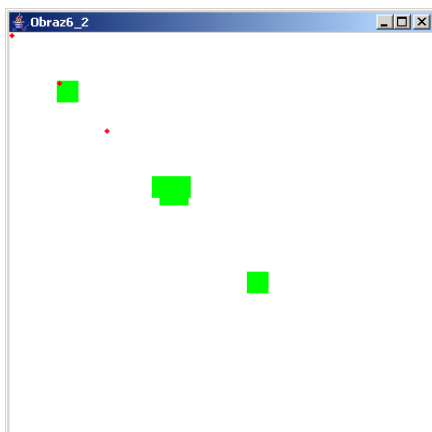
Uwaga:

Popraw definicje klas z rodziny *Punkt*, dodając metodę o takim samym nagłówku w każdej z tych klas, która sprawdza, czy punkt o podanych współrzędnych *x1*, *y1* podanych jako parametry metody, leży na powierzchni danej figury. Wtedy metoda *mouseClicked* może wywoływać w pętli tę metodę każdej figury – wynik zwrócony przez metodę dla jednej z figur równy *true* oznacza, że przycisk myszy został kliknięty na powierzchni tej figury.

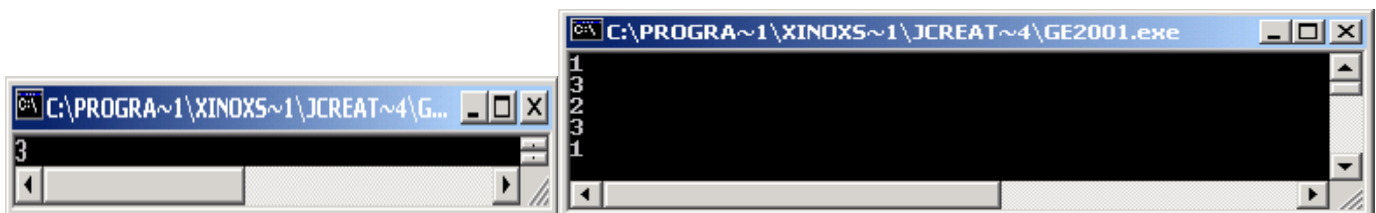
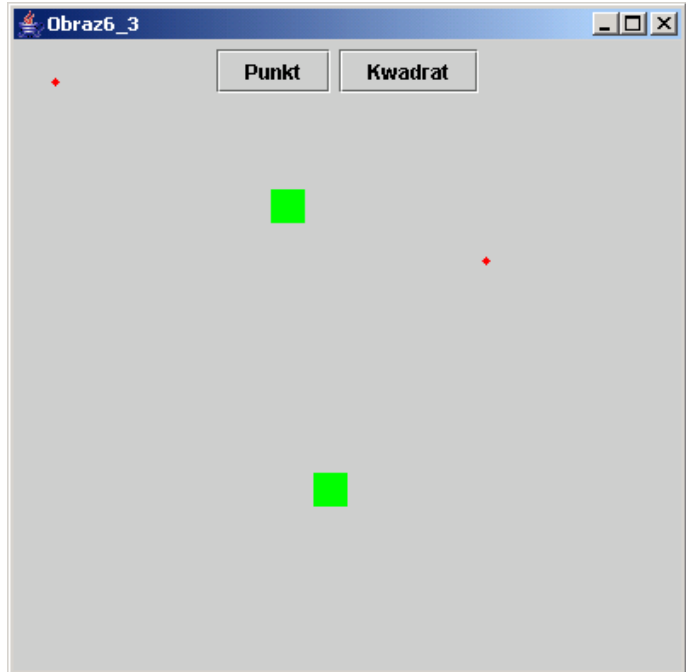
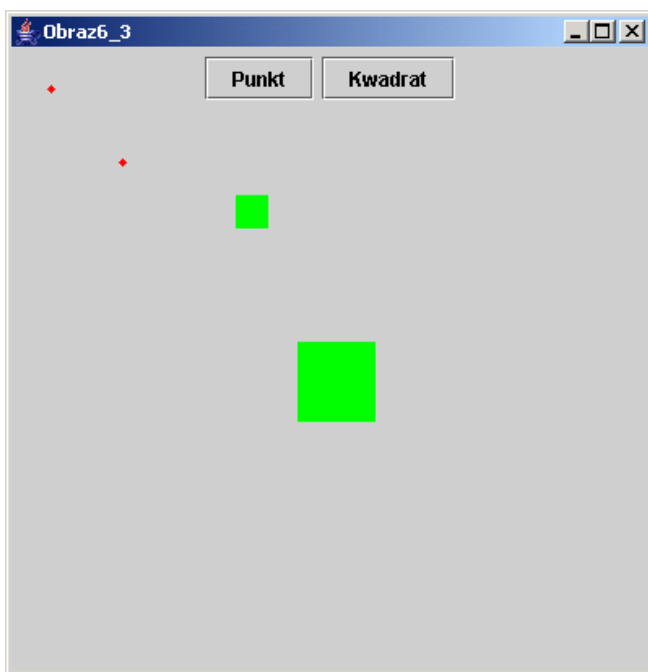
```
public void mouseClicked(MouseEvent e) // obecna metoda
// Invoked when the mouse button has been clicked (pressed and released) on a component.
{int x_=e.getX();
int y_=e.getY();
for (int i=0; i<figury.length;i++)
if (figury[i].podajX()+figury[i].podajDl(>=x_ && figury[i].podajX()<=x_ &&
figury[i].podajY()+figury[i].podajDl(>=y_ && figury[i].podajY()<=y_)
{ biezacy=i;
System.out.println(biezacy);
break;}
requestFocus(); //przywrocenie zdolnosc do obslugi zdarzen od klawiatury
repaint();
}
public void mouseClicked(MouseEvent e) // metoda po zmianie
// Invoked when the mouse button has been clicked (pressed and released) on a component.
{int x_=e.getX();
int y_=e.getY();
for (int i=0; i<figury.length;i++)
if (figury[i].lezy_na(x_,y_)) //po dodaniu metody lezy_na(int x_, int y_) w klasie Punkt
```

/\*Należy napisać jedną metodę *lezy\_na* w klasie *Punkt*, wykorzystującą metody wirtualne: *podajDI*, zwracającą aktualny rozmiar figury np w kierunku poziomym oraz *podajSzer*, zwracającą rozmiar w kierunku pionowym - obie metody w klasie *Punkt* zwracają rozmiar średnicy tej figury; w klasie *Kwadrat* obie metody zwracają długość boku, a w klasie *Prostokat* metoda *podajDI* jest dziedziczona, a metoda *podajSzer* zwraca teraz rozmiar drugiego boku prostokąta - *szerokosc*\*/

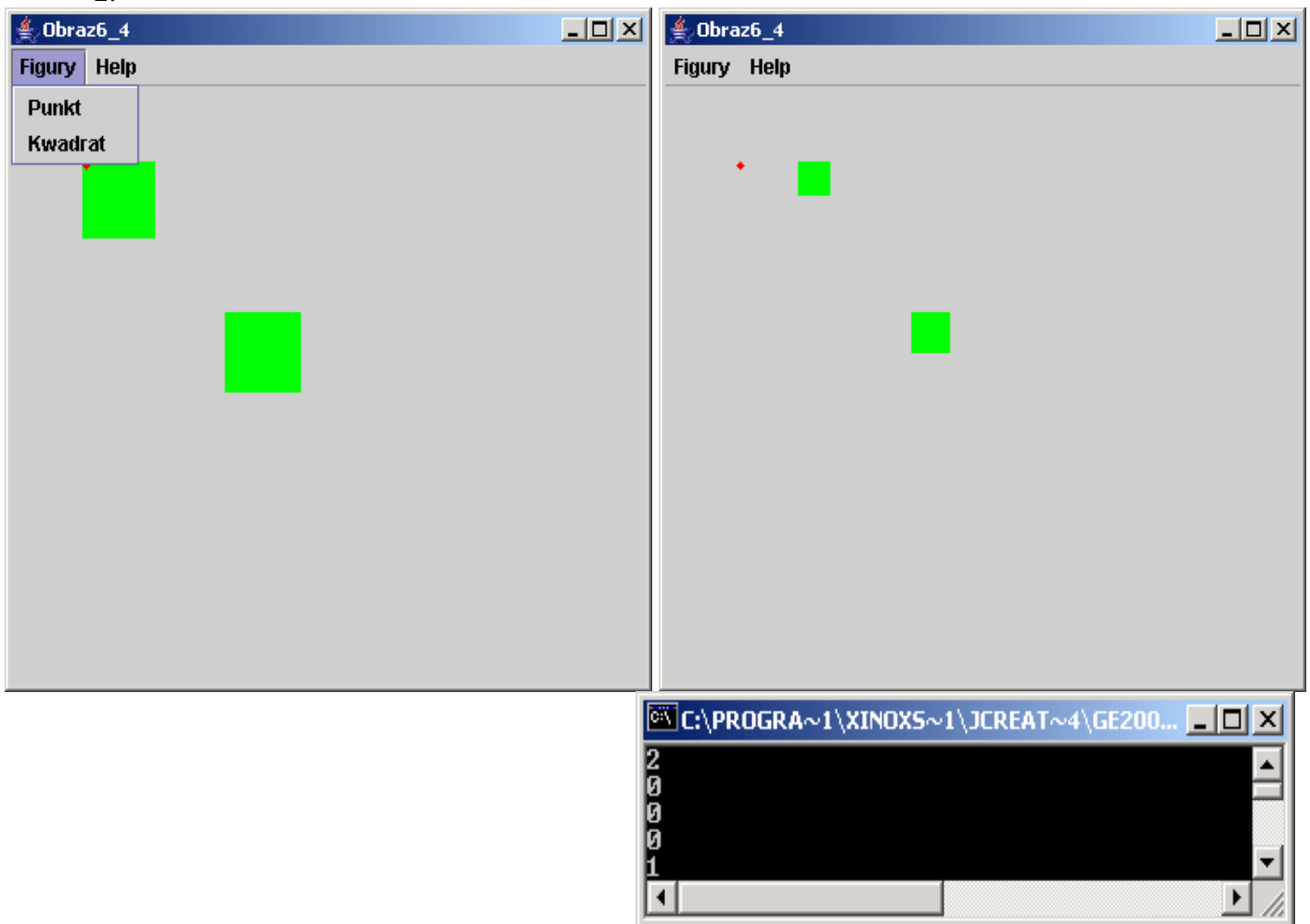
```
{ biezacy=i;
System.out.println(biezacy);
break;}
requestFocus(); //przywrocenie zdolnosc do obslugi zdarzen od klawiatury
repaint();
}
```



3. Wykonaj program (klasa *Obraz\_6\_3*), który wyświetla na ekranie dane figury typu *Punkt*, *Kwadrat* tak, jak program *Obraz6\_3* (projekt *Rys6\_3*) oraz dodatkowo dane prostokąty. Klasa *Prostokat* powinna dziedziczyć od klasy *Kwadrat* (definicja konstruktora w klasie *Prostokat*) i dodatkowo powinna posiadać metody obsługujące drugi bok prostokąta (*szerokosc*). W programie *Obraz\_6\_3*, który ma teraz dodatkowo definicję klasy *Prostokat* w porównaniu z programem *Obraz6\_3*, należy w klasie *Figury* dodać wstawianie obiektów typu *Prostokat* do tablicy *figury* za pomocą metody *Dodaj\_prostokat()*. W programie *Obraz6\_3* wstawianie figur do animacji dokonuje się za pomocą przycisków typu *JButton*. Należy w programie *Obraz\_6\_3* dodać przycisk *prostokat* do wstawiania figur typu *Prostokat*. Obsługa przycisków jest możliwa po implementacji metod słuchacza aktywności przycisków typu *JButton* – *ActionListener*. Jest nim główne okno aplikacji – pochodne od *JFrame*. W programie do animacji można wybrać figurę, podając z klawiatury jej indeks w tablicy *figury* i następnie klawiszami strzałek przesuwać ją: w lewo, w prawo, w górę oraz w dół. Figury przesuwają się, nie pozostawiając śladu. Teraz wywołanie metody *super.paintComponent* w metodzie *paintComponent()* jest obowiązkowe, gdyż umożliwi prawidłowe narysowanie przycisków i powoduje przesuwanie figur bez śladu (odświeżanie zawartości okna). W przypadku naciśnięcia klawisza *shift* można zmienić zakres losowanych wartości przesunięcia figury. Obsługa klawiatury jest możliwa po implementacji metod słuchacza klawiatury – *KeyListener*. Można dodatkowo wybierać figurę do animacji za pomocą klikania myszą w wybraną figurę. Obsługa myszy jest możliwa po implementacji metod słuchacza klawiszy myszy – *MouseListener*. Słuchaczami obu typów zdarzeń jest klasa *Figury*, dziedzicząca od *JPanel*. Wykonaj poleceni podane w *Uwaga* zadania 2.



4. Wykonaj program (klasa *Obraz\_6\_4*), który wyświetla na ekranie dane figury typu *Punkt*, *Kwadrat* tak, jak program *Obraz6\_4* (projekt *Rys6\_4*) oraz dodatkowo dane prostokąty. Klasa *Prostokat* powinna dziedziczyć od klasy *Kwadrat* (definicja konstruktora w klasie *Prostokat*) i dodatkowo powinna posiadać metody obsługujące drugi bok prostokąta (*szerokosc*). W programie *Obraz\_6\_4*, który ma teraz dodatkowo definicję klasy *Prostokat* w porównaniu z programem *Obraz6\_4*, należy w klasie *Figury* dodać wstawianie obiektów typu *Prostokat* do tablicy *figury* za pomocą metody *Dodaj\_prostokat()*. W programie *Obraz6\_4* wstawianie figur do animacji dokonuje się za pomocą rozwijanego menu: obiektów typu *JMenu*, *JMenuItem*, *JMenuBar*. Należy w programie *Obraz\_6\_4* dodać pozycję rozwijanego menu typu *JMenuItem* *prostokat* do wstawiania figur typu *Prostokat*. Należy także obsłużyć pomoc programu, czyli rozwijane menu *Help* w celu podania informacji o obsłudze programu: jedna pozycja powinna dotyczyć wstawiania figur, druga wyboru figury do animacji. Do wyświetlania informacji można wykorzystać standardowe okienko *JOptionPane.showMessageDialog*. Obsługa menu jest możliwa po implementacji metod słuchacza aktywności pozycji rozwijanego menu – *ActionListener*. Jest nim główne okno aplikacji – pochodne od *JFrame*. W programie do animacji można wybrać figurę, podając z klawiatury jej indeks w tablicy *figury* i następnie klawiszami strzałek przesuwać ją: w lewo, w prawo, w górę oraz w dół. Figury przesuwiają się, nie pozostawiając śladu. Teraz wywołanie metody *super.paintComponent* w metodzie *paintComponent()* jest obowiązkowe, gdyż umożliwia prawidłowe narysowanie przycisków i jednocześnie powoduje przesuwanie figur bez śladu (odświeżanie zawartości okna). W przypadku naciśnięcia klawisza *shift* można zmienić zakres losowanych wartości przesunięcia figury. Obsługa klawiatury jest możliwa po implementacji metod słuchacza klawiatury – *KeyListener*. Można dodatkowo wybierać figurę do animacji za pomocą klikania myszą w wybraną figurę. Obsługa myszy jest możliwa po implementacji metod słuchacza klawiszy myszy – *MouseListener*. Słuchaczami obu typów zdarzeń jest klasa *Figury*, dziedzicząca od *JPanel*. Wykonaj polecenie podane w *Uwaga* zadania 2.



5\*. W zadaniu tym należy do animacji z zadania 3 lub 4 wprowadzić rozpoznanie zderzenia się figur za pomocą metody *odleglosc* i np. po wykryciu zderzenia zmieniać ich kolor lub kształt.