Programiranje 2

Klase

dr Đorđe Obradović

Singidunum - Centar Novi Sad

Klase

Primer tačke u 2D prostoru

- naziv klase: Point
- atributi:
  - x : double
  - y : double
- metode:
  - rastojanje
- operator +

Organizacija projekta

- main.cpp pocetak izvrsavanja programa
- point.h opis klase Point
- point.cpp implementacija klase Point
- makefile pravila za kompajliranje

point.h

```
0
1  class Point {
2    public:
3      double x;
4      double y;
5
6      Point(double, double);
7
8      double rastojanje(Point);
9      Point operator+(Point);
10 };
```

point.cpp

```
0  #include "point.h"
1  #include<math.h>
2
3  Point::Point(double aX, double aY) {
4    x = aX;
5    y = aY;
6  }
7
8  double Point::rastojanje(Point p){
9    double ret = 0;
10   ret = sqrt((x-p.x)*(x-p.x)+(y-p.y)*(y-p.y));
11   return ret;
12 }
13
14 Point Point::operator+(Point p){
15   return Point(x+p.x, y+p.y);
16 }
```

main.cpp

```
0
1  #include "point.h"
2  #include <iostream>
3  using namespace std;
4
5
6  int main() {
7     Point p1(0, 2);
8     Point p2(1, 1);
9     Point p3 = p1+p2;
10
11    cout << "P3: x:" << p3.x << " y:" << p3.y<<"\n";
12    cout << "Rastojanje p1 i p2: " << p1.rastojanje(p2)<<"\n";
13    return 0;
14 }
```

Standardna biblioteka

vektori

```
0  #include "point.h"
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  void primer2(){
6    int n = 5;
7    vector<Point> lista;
8
9    for(int i=0; i<n; i++){
10     lista.push_back(Point(i, 2));
11   }
12 }
```

map

```
0  #include "point.h"
1  #include <iostream>
2  #include <map>
3
4  using namespace std;
5
6  void primer3(){
7    map<string, Point*> mapa;
8    mapa["prvi"] = new Point(1,2);
9    mapa["drugi"] = new Point(3,2);
10   cout<<"mapa[\"prvi\"]:"<<mapa["prvi"]->x<<"\n";
11 }
12
13 int main() {
14    primer3();
15    return 0;
16 }
```

Vektori

Zbir elemenata vektora

```
0  #include "point.h"
1  #include <iostream>
2  #include <string>
3  #include <vector>
4  using namespace std;
5
6  vector<double> primer4(){
7    vector<double> lista;
8    double el = -1;
9    while(el != 0){
10     cout<<"Unesite element:";
11     cin>>el;
12     lista.push_back(el);
13   }
14   return lista;
15 }
```

Zbir elemenata vektora

```
0
1  double suma(vector<double> lista){
2    double ret = 0;
3    int n = lista.size();
4    for(int i=0; i<n; i++){
5      ret += lista[i];
6    }
7    return ret;
8  }
```

Zbir elemenata vektora

```
0  int main() {
1
2      cout << "Zbir elemenata: "<<suma(primer4())<<"\n";
3      return 0;
4  }
```

Matrice

Matrice

```
0  float a[2][2], b[2][2], rez[2][2];
1
2  a[0][0] = 1;
3  a[0][1] = 1;
4  a[1][0] = 1;
5  a[1][1] = 1;
6
7  b[0][0] = 1;
8  b[0][1] = 1;
9  b[1][0] = 1;
10 b[1][1] = 2;
```

Matrice

```
10
11  for(int i=0; i<2; i++){
12    for(int j=0; j<2; j++){
13      rez[i][j] = a[i][j]+b[i][j];
14    }
15  }
```

Datoteke

Rad sa datotekama

```
0  #include <iostream>
1  #include <iomanip>
2  #include <fstream>
3  using namespace std;
4
5  int main() {
```

Rad sa datotekama

```
5       int sum = 0;
6       int x;
7       ifstream inFile;
8
9       inFile.open("test.txt");
10      if (!inFile) {
11          cout << "Unable to open file";
12          return 1; // terminate with error
13      }
```

Rad sa datotekama

```
14      while (inFile >> x) {
15          sum = sum + x;
16      }
17
18      inFile.close();
19      cout << "Sum = " << sum << endl;
20
21      ofstream myfile;
22      myfile.open ("out.txt");
23      myfile << "Sum = " << sum << endl;
24      myfile.close();
```

Stringovi

Tokenizacija

```
0  string s = "ako;b;c;d";
1  string delimiter = ";";
2
3  size_t pos = 0;
4  string token;
5  while ((pos = s.find(delimiter)) != string::npos) {
6      token = s.substr(0, pos);
7      cout << token << endl;
8      s.erase(0, pos + delimiter.length());
9  }
10 cout << s << endl;
```

Spajanje stringova

```
0 string sa = "Nesto ";
1 string sb = "drugo";
2 s = sa + sb+" kraj";
3 cout << s << endl;
```

Podstring

```
0  int i = 0;
1  int app = 0;
2  string input = "danas nam je divan dan";
3  for(i = input.find("dan", 0); i != string::npos;
4      i = input.find("dan", i))
5  {
6      app++;
7      i++;
8  }
9  cout<<app;
```