Friedrich-Schiller-Universität Jena
Fakultät für Mathematik und Informatik
Institut für Informatik
Lehrstuhl Digitale Bildverarbeitung

Dr. Sven Sickert

Programmierübung zur Vorlesung
Objektorientierte Programmierung mit C++
Sommersemester 2021

Aufgabenblatt 7: Klassen

Ausgabe: 21.06.2021 Abgabe: 28.06.2021

Aufgabe 1 Konstruktoren (4 Punkte)

Implementieren Sie Grundfunktionen einer Klasse Time, die zur Speicherung der Uhrzeit bzw. eines Zeitraumes **eine** Variable `minuten` enthält!

```
class Time {
// Daten
private:
  int minuten;

// Methoden
public:
  virtual void set(int h,int m);
  virtual void get(int &h,int &m) const;
  virtual void inc();
};
```

Ergänzen Sie folgendes:

- Einen Standard-Konstruktor, der den Minuten-Wert mit 0 initialisiert.
- Einen Konstruktor, der einen Integerwert (Minuten) als Parameter hat.
- Einen Konstruktor, der Stunde und Minute (int) als Parameter hat.
- Einen Kopier-Konstruktor.
- Einen (virtuellen) Destruktor.

Fügen Sie den Konstruktoren und dem Destruktor Ausgaben hinzu, die den Aufruf dokumentieren, z.B. mit `cout << "Standard-Konstruktor" << endl;` .
Folgendes Testprogramm verwendet die Klasse Time:

```
void print(Time t)
```

```
{
  int h, m;
  t.get(h, m);
  cout << h << ":" << m << endl;
}

Time input()
{
  Time result;
  int h, m;
  cout << "Uhrzeit eingeben - Stunde: " ;
  cin >> h ;
  cout << "                   Minute: " ;
  cin >> m ;
  result.set(h, m);
  return result;
}

int main(int argc, char* argv[])
{
  Time t1(10, 10);
  Time t2;
  t2 = input();
  t2.Inc();
  print(t2);
  print(1234);
  return 0;
}
```

Bilden Sie dieses Programm! Erklären Sie jeden Konstruktor- und Destruktor-Aufruf, der sich bei der Ausführung meldet!

Aufgabe 2 Integer-Menge (6 Punkte)

Schreiben Sie eine Klasse in C++, die den Datentyp *IntMenge* implementiert. *IntMenge* repräsentiert eine Menge von Zahlen des Types `int` aus einem beim Anlegen der Menge vorzugebenden Wertebereich. Dies soll durch einen `vector<bool>` realisiert werden, der für jeden Wert des gegebenen Bereiches festhält, ob er zur Menge gehört. *IntMenge* wird durch einen Konstruktor `IntMenge(int min,int max)` konstruiert und erlaubt folgende Operationen:

- `bool isValid(int i)` testet, ob `i` im Wertebereich der Menge von `min` bis `max` liegt.
- `void add(int i)` fügt die übergebene Zahl der Menge hinzu, wenn sie noch nicht darin vorkommt
- `void remove(int i)` entfernt die übergebene Zahl aus der Menge, wenn sie darin enthalten ist
- `bool contains(int i)` gibt `true` zurück, wenn die übergebene Zahl in der Menge ist, ansonsten `false`
- `bool isEmpty()` gibt `true` zurück, wenn die Menge leer ist
- `int getSize()` liefert die Anzahl der Elemente der Menge zurück
- `vector<int> getElements()` liefert einen `int`-Vektor zurück, welcher alle Elemente der Menge enthält

- `void print()` gibt die Elemente der Menge auf dem Bildschirm aus

Verwenden Sie ein Testprogramm der folgenden Art, um Ihre Implementierung zu überprüfen:

```
int main()
{
    IntMenge m(10,100);
    int input;
    std::cin >> input ;
    while ( input != 0 )
    {
       if ( m.isValid(input) )
           m.add(input);
       std::cout << "Menge enthält " << m.getSize()
                 << " Elemente:" << std::endl;
       m.print();
       std::cin >> input;
    }
    std::vector<int> intv = m.getElements();
    for ( int i=0; i<intv.size(); i++ )
        m.remove(intv[i]);
    std::cout << (m.isEmpty() ? "Menge ist leer" : "Menge ist nicht leer")
              << std::endl;
}
```