

Домашно No. 2

Забележка: Домашното няма да се оценява. Дадените точки са за информация на студента каква оценка би получил в зависимост от представеното решение на домашното.

Приложете следните принципи на Обектно ориентираното програмиране:

- *hiding of information*
- *software reuse*
- *inheritance*
- *polymorphism*

за решаване на следната задача.

Даден е `class Vector` със следната декларация:

```
public class Vector
{
    // reference to an array of int
    int[] p;
    // value of the first index of a Vector element
    final int first;
    // the number of elements in Vector,
    // i.e the length of Vector
    final int length;
    Vector () {
        // default constructor
    }
    Vector (int first, int length) {
        // general purpose constructor
    }
    Vector (Vector v) {
        //copy constructor
    }
}
```

`class Vector` трябва да представя масив (`p[]`) с произволна дължина (`length`), чиито елементи са от тип `int`. За разлика от обикновени масиви, **индексването не винаги е задължително да започва от нула**. Вместо това, може да се избира стойност (`first`) за индекса на първия елемент. Индексването трябва да е безопасно, тъй че да не е възможно да се посочва елемент извън масива.

Предайте архивирани в един WinRaR файл NetBeans Java приложението- да е в `vectorpack package`

Problem tasks

1. Задайте **подходящи модификатори за достъп** до данните на класа, за да приложите **information hiding**. Дефинирайте по-горе декларираните **class constructors**, а също и един **toString()** метод, прилагайки **software reuse**. Изисква се **first** да е неотрицателно (0 по подразбиране) и **length** да е положително число (1 по подразбиране).

Точки:10

2. Дефинирайте **get** метод с подходящи модификатори за достъп (**access modifiers**):

- a) `getFirst`- връща първия индекс от `Vector`
- b) `getLast`- връща последният индекс от `Vector`
- c) `getLength`- връща дължината на `Vector`

Точки:6

3. Дефинирайте

```
void setAt(int myIndex, int myValue)
```

и
`int getFrom(int myIndex)`

методи, позволяващи да се манипулират елементите на масива `p[]` на `Vector` спрямо зададеният от потребителя първи индекс (`first`). При погрешно зададен индекс се издава съобщение за грешка с диалогов прозорец `JOptionPane` и операцията (`set` или `get`) не се изпълнява.

Точки:10

4. Добавете целочислена данна `count` към `class Vector` и необходим `program code`, така че `count` да показва във всеки момент колко обекта от `class Vector` са активни (създадени и се използват)

Точки:10

5. Дефинирайте `interface Polynomial` и го приложете към `class Vector`. `interface Polynomial` да съдържа само следния метод:

`double Evaluate(double x);`

които пресмята полином на `x`, където степените на `x` и съответните им коефициенти на такъв полином са зададени от индексите и компонентите на обект `v` от `class Vector` т.е. полиномът пресмятан с по- горните методи е

$$\sum_{i=i1}^{i1+num-1} (p[i-i1] x^i)$$

където `i1= first`, `num= length` и `p[]` са данните на `v`

Точки:14

6. Напишете графично приложение `class VectorTest`, което:

- а) Създава обект `v1` от `class Vector`, съответстващо на полинома $3x^3 - 2x$
б) Използва графичен потребителски интерфейс за въвеждане на аргумента `x` на полинома `v1` и след пресмятането му за въведената стойност на `x`, изведете получената стойност, форматирана с два знака след десетичната запетая
в) Изведете в графичния интерфейс и стойността на `count` данната в `class Vector`

Точки:40

7. Компилирайте и изпълнете Java приложението.

Точки:10

Използвайте правилата за Добро програмиране, дадени на лекции, компилирайте и тествайте програмата си с различни входни данни.

Оценки:

2	от 0 до 54 точки
3	от 55 до 64 точки
4	от 65 до 74 точки
5	от 75 до 84 точки
6	от 85 до 100 точки

Качете Java сорс файла от свое име на web site- а за курса в рамките на седмица 2