

```

package de.preclipse.bo;

import java.io.Serializable;
import java.util.List;
import java.util.ArrayList;
import java.util.Iterator;

public class PrevalentSystem implements Serializable {

    private static final long serialVersionUID = 1L;
    private List<Classe> classe = new ArrayList<Classe>();
    private long nextClasseID = 1;
    private List<Docente> docente = new ArrayList<Docente>();
    private long nextDocenteID = 1;
    private List<Sala> sala = new ArrayList<Sala>();
    private long nextSalaID = 1;
    private List<Curso> curso = new ArrayList<Curso>();
    private long nextCursoID = 1;
    private List<EncontroPredefinido> encontroPredefinido = new
ArrayList<EncontroPredefinido>();
    private long nextEncontroPredefinidoID = 1;

    public PrevalentSystem() {
        super();
    }

    public List<Classe> getClasse() {
        return classe;
    }

    public void setClasse(List<Classe> classe) {
        this.classe = classe;
    }

    public List<Docente> getDocente() {
        return docente;
    }

    public void setDocente(List<Docente> docente) {
        this.docente = docente;
    }

    public List<Sala> getSala() {
        return sala;
    }

    public void setSala(List<Sala> sala) {
        this.sala = sala;
    }

    public List<Curso> getCurso() {
        return curso;
    }

    public void setCurso(List<Curso> curso) {
        this.curso = curso;
    }
}

```

```

    public List<EncontroPredefinido> getEncontroPredefinido() {
        return encontroPredefinido;
    }

    public void setEncontroPredefinido(List<EncontroPredefinido>
encontroPredefinido) {
        this.encontroPredefinido = encontroPredefinido;
    }

    public long getNextClasseID() {
        return nextClasseID;
    }

    public void setNextClasseID(long nextClasseID) {
        this.nextClasseID = nextClasseID;
    }

    public long getNextDocenteID() {
        return nextDocenteID;
    }

    public void setNextDocenteID(long nextDocenteID) {
        this.nextDocenteID = nextDocenteID;
    }

    public long getNextSalaID() {
        return nextSalaID;
    }

    public void setNextSalaID(long nextSalaID) {
        this.nextSalaID = nextSalaID;
    }

    public long getNextCursoID() {
        return nextCursoID;
    }

    public void setNextCursoID(long nextCursoID) {
        this.nextCursoID = nextCursoID;
    }

    public long getNextEncontroPredefinidoID() {
        return nextEncontroPredefinidoID;
    }

    public void setNextEncontroPredefinidoID(long
nextEncontroPredefinidoID) {
        this.nextEncontroPredefinidoID = nextEncontroPredefinidoID;
    }

    public Classe addClasse(int codigo, String nome, int curso, int
cargaHoraria, int minAulasGeminadas, int limiteDiarioAulas, int
numAlunos, String siapeDocente, ArrayList<Restricao> restricoes) {
        Classe newClasse = new Classe(nextClasseID++, codigo, nome,
curso, cargaHoraria, minAulasGeminadas, limiteDiarioAulas, numAlunos,
siapeDocente, restricoes);
        this.classe.add(newClasse);
        return newClasse;
    }

```

```

    }

    public Classe getClasse(int codigo) {
        for (Iterator<Classe> i = classe.iterator(); i.hasNext(); ) {
            Classe currentClasse = (Classe) i.next();
            if (currentClasse.getCodigo() == codigo) return
currentClasse;
        }
        return null;
    }

    public boolean removeClasse(int codigo) {
        for (Iterator<Classe> i = classe.iterator(); i.hasNext(); ) {
            Classe currentClasse = (Classe) i.next();
            if (currentClasse.getCodigo() == codigo) { i.remove();
return true; }
        }
        return false;
    }

    public int sizeClasse() {
        return classe.size();
    }

    public Docente addDocente(int siape, String nome,
ArrayList<Restricao> restricoes) {
        Docente newDocente = new Docente(nextDocenteID++, siape, nome,
restricoes);
        this.docente.add(newDocente);
        return newDocente;
    }

    public Docente getDocente(int siape) {
        for (Iterator<Docente> i = docente.iterator(); i.hasNext(); )
{
            Docente currentDocente = (Docente) i.next();
            if (currentDocente.getSiape() == siape) return
currentDocente;
        }
        return null;
    }

    public boolean removeDocente(int siape) {
        for (Iterator<Docente> i = docente.iterator(); i.hasNext(); )
{
            Docente currentDocente = (Docente) i.next();
            if (currentDocente.getSiape() == siape) { i.remove();
return true; }
        }
        return false;
    }

    public int sizeDocente() {
        return docente.size();
    }

    public Sala addSala(int siape, String nome, int capacidade) {

```

```

        Sala newSala = new Sala(nextSalaID++, siape, nome,
capacidade);
        this.sala.add(newSala);
        return newSala;
    }

    public Sala getSala(int codigo) {
        for (Iterator<Sala> i = sala.iterator(); i.hasNext(); ) {
            Sala currentSala = (Sala) i.next();
            if (currentSala.getCodigo() == codigo) return
currentSala;
        }
        return null;
    }

    public boolean removeSala(int codigo) {
        for (Iterator<Sala> i = sala.iterator(); i.hasNext(); ) {
            Sala currentSala = (Sala) i.next();
            if (currentSala.getCodigo() == codigo) { i.remove();
return true; }
        }
        return false;
    }

    public int sizeSala() {
        return sala.size();
    }

    public Curso addCurso(int codigo, int turno, String nome) {
        Curso newCurso = new Curso(nextCursoID++, codigo, turno,
nome);
        this.curso.add(newCurso);
        return newCurso;
    }

    public Curso getCurso(int codigo) {
        for (Iterator<Curso> i = curso.iterator(); i.hasNext(); ) {
            Curso currentCurso = (Curso) i.next();
            if (currentCurso.getCodigo() == codigo) return
currentCurso;
        }
        return null;
    }

    public boolean removeCurso(int codigo) {
        for (Iterator<Curso> i = curso.iterator(); i.hasNext(); ) {
            Curso currentCurso = (Curso) i.next();
            if (currentCurso.getCodigo() == codigo) { i.remove();
return true; }
        }
        return false;
    }

    public int sizeCurso() {
        return curso.size();
    }

```

```
        public EncontroPredefinido addEncontroPredefinido(Restricao
restricao, Docente docente, Classe classe) {
            EncontroPredefinido newEncontroPredefinido = new
EncontroPredefinido(nextEncontroPredefinidoID++, restricao, docente,
classe);
            this.encontroPredefinido.add(newEncontroPredefinido);
            return newEncontroPredefinido;
        }

        public int sizeEncontroPredefinido() {
            return encontroPredefinido.size();
        }
    }
}
```