***СУСИ***

***1. Увод***

*1.1 Описание и идея на проекта:*

**- Проектът „СУСИ“ реализира информационна система, която записва и съхранява данни за студенти. Програмата изпълнява определен брой действия спрямо наличните студенти. Информацията за студентите може да се извлича от файлове.**

*1.2. Цели и задачи:*

**- За пълноценно развитие на информационната система следните цели и задачи трябва да бъдат изпълнени:**

***1.2.1. Представяне на информация за даден студент по подходящ начин.***

***1.2.2. Изграждане на подходяща структура за програмата, която да поддържа наличните студенти и промените, които могат да се извършват спрямо тях.***

***1.2.3. Прочитане на данни от файлове и записване на направените промени във файл съответно за студентите.***

*1.3. Структура на документацията:*

**1.3.1. Увод**

**1.3.2. Проектиране**

**1.3.3 Реализация**

***2. Проектиране***

*- За реализацията на проекта са използвани следните класове:*

* class Vector – Шаблонен клас
* Има следните член-данни:
* T\* vector – Динамичен масив от тип подадения шаблон
* size\_t size – Размер на масива
* size\_t capacity – Капацитет на масива
* Има следните private методи:
* void erase() – Изтрива на масива
* void copy(const Vector<T>&) – Копира
* void resize() – Удвоява капацитета
* Има следните методи:
* Vector() – Дефолтен конструктор
* Vector(const Vector<T>&) – Копи конструктор
* Vector<T>& operator=(const Vector<T>&) – Оператор =
* ~Vector() - Деструктор
* size\_t getSize() const – Взима размера на масива
* void insert(const T&, const size\_t&) – Вмъква елемент на указана позиция
* void add(const &) – Добавя елемент в края на масива
* void remove(const size\_t&) – Премахва елемент на указана позиция
* void clear() – Изчиства масива
* void print() const – Извежда всички елементи
* T& operator[](const size\_t&) const – Връща елемент от указана позиция
* bool operator!=(const char\*) const – Оператор !=
* friend std::ostream& operator<<(std::ostream& out, const Vector<T>& vector) – Приятелска функция, предефинираща оператор <<
* friend std::istream& operator>>(std::istream& in, const Vector<T>& vector) – Приятелска функция, предефинираща оператор >>
* class String
* Има следните член данни:
* char\* name – Чар масив
* Има следните private методи:
* void copy(const String&) - Копира
* void erase() – Изтрива масива
* Има следните методи:
* String() – Дефолтен конструктор
* String(const char\*) - Конструктор
* String(const String&) – Копи конструктор
* String& operator=(const String&) – Оператор =
* ~String() - Деструктор
* void clear() – Изчиства масива
* const char\* getName() const – Връща масива
* size\_t getLength() const – Връща размера на масива
* size\_t convertFromString() – Конвертира от Стринг
* void remove(const size\_t&) – Премахва елемент на указана позиция
* String& operator=(const char\*)
* bool operator==(const String&) const
* bool operator==(const char\*) const
* bool operator!=(const String&) const
* bool operator!=(const char\*) const
* char& operator[](const size\_t&)
* String& operator+=(const char&)
* friend std::ostream& operator<<(std::ostream& out, const String& str)
* friend std::istream& operator>>(std::istream& in, String& str)
* class Discipline
* Има следните член данни:
* String name – Име на дисциплина
* String type – Тип на дисциплине (Избираема/Задължителна)
* size\_t course – Курс, в който може да бъде записана дисциплина
* Double mark – Оценка за дисциплина
* Има следните методи:
* Discipline()
* Discipline(const String, const String, size\_t)
* void setName(const String&) – Задава име на дисциплина
* void setMark(const double) – Задава оценка за дисциплина
* const String getName() const – Връща името на дисциплината
* size\_t getCourse() const – Връща курса на дисциплината
* double getMark() const – Връща оценката за дисциплината
* Discipline& operator=(const Discipline&)
* bool operator==(const Discipline&) const
* bool operator!=(const Discipline&) const
* friend std::ostream& operator<<(std::ostream& out, const Discipline& disc)
* friend std::istream& operator>>(std::istream& in, Discipline& disc)
* class Specialty
* Има следните член данни:
* String name – Име на специалност
* Vector<Discipline> list – Списък от дисциплини за специалността
* Има следните методи:
* Specialty()
* Specialty(const String \_name)
* const String getName() const
* size\_t getListSize() const
* bool isDisciplineInList(const Discipline&) const
* void addDiscipline(const Discipline&)
* void print() const
* Specialty& operator=(const Specialty&)
* bool operator==(const Specialty&) const
* bool operator==(const char\*) const
* bool operator!=(const Specialty&) const
* friend std::ostream& operator<<(std::ostream& out, const Specialty& spec)
* friend std::istream& operator>>(std::istream& in, Specialty& spec)
* class Student
* Има следните член данни:
* String name – Име на студент
* size\_t facultyNumber – Факултетен номер
* size\_t course - Курс
* Specialty specialty - Специалност
* size\_t group - Група
* String status – Статус (Записан/Прекъснал/Завършил)
* Vector<Discipline> list – Списък от дисциплини, в които е регистриран
* Има следните методи:
* Student()
* double avarage() – Връща средно аритметичния успех на студента
* bool passedAllExams() const – Проверява дали е взел всички изпити
* bool isDiscInList(const Discipline&) – Проверява дали се съдържа указана дисциплина в списъка с дисциплини
* void removeDisc(const size\_t) – Премахва дисциплина от списъка с дисциплини
* void clearList() – Изчиства целия списък с дисциплини
* void setSpecialty(const Specialty&)
* void setGroup(const size\_t&)
* void setCourse(const size\_t&)
* const String getName() const
* size\_t getFacultyNumber() const
* size\_t getCourse() const
* const String getSpecialty() const
* size\_t getListSize() const
* void save(std::ofstream&) – Запазва данните за студент във файл
* void load(std::ifstream&) – Чете данните за студент от файл
* void enroll(const size\_t&, const Specialty&, const size\_t&, const String&) – Записва студент с посочените данни
* void advance() – Прехвърля студент в следващ курс
* void graduate() – Маркира студент като завършил
* void interrupt() – Маркира студент като прекъснал
* void resume() – Маркира студент като записан
* void print() – Извежда информация за студент
* void takeUpDiscipline(const Discipline&) – Записва студент в указана дисциплина
* void addMark(const Discipline&, const double&) – Добавя оценка на студент за указана дисциплина
* void examsInfo() – Извежда информация относно всички взети и невзети изпити
* void protocol(const Discipline&) – Ако студент е записан в указана дисциплина, се извежда информация за студента
* friend std::ostream& operator<<(std::ostream& out, Student& st)
* friend std::istream& operator>>(std::istream& in, Student& st)
* class Command
* Има следните член данни:
* String command – Команда
* String filename – Име на файл
* size\_t fileSize – Брой на студенти, записани в указан файл
* bool isOpen – Информация дали указан файл е отворен
* Vector<Student> students – Списък от студенти
* Vector<Specialty> specialties – Списък от специалности
* Vector<Discipline> disciplines – Списък от дисциплини
* Има следните методи:
* Command()
* void commandList() const – Извежда всички команди
* void chooseCommand() – Избор и изпълнение на команда
* bool checkFn(const size\_t&) const – Проверява дали съществува указан факултетен номер
* bool checkSpec(Specialty&) – Проверява дали съществува указана специалност
* bool checkDisc(Discipline&) – Проверява дали съществува указана дисциплина
* String takeCommand(String) – Взима ключовата дума за команда от въведените команда и данни
* size\_t takeFacultyNumber(String&) – Взима факултетен номер от команда
* String takeSpecialty(String) – Взима специалност от команда
* size\_t takeNumber(String) – Взима група/курс от команда
* String takeName(String) – Взима име от команда
* const char\* takeFileName(String&) – Взима име на файл от команда
* void printDisciplines() const – Извежда всички дисциплини
* void printSpecialties() const – Извежда всички специалности
* void printDiscInfo(Discipline&) – Извежда информация за указана дисциплина
* void printSpecInfo(Specialty&) – Извежда информация за указана специалност
* void save() – Запазва информацията в текущо отворен файл
* void saveAs() – Запазва информацията в указано местоположение
* void open() – Отваря файл от указано местоположение
* void close() – Затваря текущо отворен файл
* void enroll(const size\_t&, Specialty&, const size\_t&, const String&) – Записва студент с указани данни
* void advance(const size\_t&) – Прехвърля студент в следващ курс
* void change(const size\_t&) – Сменя указана данна за студент
* void graduate(const size\_t&) – Маркира студент като завършил
* void interrupt(const size\_t&) – Маркира студент като прекъснал
* void resume(const size\_t&) – Маркира студент като записан
* void print(const size\_t&) const – Извежда информация за указан студент
* void printAll(Specialty&, const size\_t&) – Извежда информация за всички студенти в указана специалност и курс
* void enrollIn(const size\_t&, Discipline&) – Записва студент в указана дисциплина
* void addGrade(const size\_t&, Discipline&, const double&) – Задава оценка на указан студент за указана дисциплина
* void protocol(Discipline&) – Извежда информация за всички студенти, записани в указана дисциплина
* void report(const size\_t&) const – Извежда информация за всички взети/невзети изпити на указан студент

***3. Реализация***

* **Въвеждане и четене от файл на студенти: данните за всеки студент са на отделен ред**.
* **Записване на задължителни/избираеми дисциплини само за регистрирани студенти.**
* **Добавяне на оценки/изпити (за указана дисциплина, оценка и регистриран студент).**
* **Отпечатване на протоколи относно студенти, дисциплини или специалности.**
* **Академична справка за оценките на посочен студент.**
* **Изпълняване на команди от командния ред.**

***Дата: 23.05.2021г. Автор: Стилян Тодоров***